https://www.overleaf.com/project/627337754fa5db1037f4f028

The min-timespan parallel technician-and-drone scheduling in door-to-door sampling service system

1st Pham Phu Manh
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
phumanh1998@gmail.com

2nd Tran Thi Hue
Faculty of Management Information Systems
Banking Academy of Vietnam
Hanoi, Vietnam
huett@hvnh.edu.vn

3rd Huynh Thi Thanh Binh
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
binhht@soict.hust.edu.vn

4th Nguyen Khanh Phuong
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
phuongnk@soict.hust.edu.vn

***Abstract*—This paper considers a variant of the Vehicle Routing Problem with Drones applied in the door-to-door sampling service system. Given a set of technicians and a fleet of drones departing from a medical center and working independently, the objective consists in designing feasible trips for technicians and drones such that the time from collection to arrival at the medical center of each customer's test kits is not exceeded a limited duration and minimal makespan is achieved. A mixed-integer linear program (MILP) model is presented and solved to optimality with small instances due to the computational complexity of this problem. Thus, a tabu search approach is proposed. Experiments are then carried out to demonstrate the performance of the proposed algorithm and the advantage of integrating drones into the sampling service system.**

***Index Terms*—tabu search, drone, healthcare, routing**

I. Introduction

In the past decade, technological advances in robotics have significantly promoted the application of drones in many fields of life, such as logistics, healthcare, disaster management, agriculture, and surveillance. In particular, the use of drones in last-mile delivery has witnessed a remarkable growth of interest in research due to the tremendous popularity of e-commerce [1]. Large companies like Amazon, DHL, Walmart, UPS, and Alibaba also began developing their own drones and launched tests, as they see the potential of deploying drones with conventional land-based transportation methods could reduce the operating costs and waiting time for last-mile delivery [1].

Moreover, drones also have been used to deliver health-related items such as first aid kits, medical aids, human body parts, personal protective equipment, etc., to provide more feasible and faster deliveries in times of critical need [2]. As of December 2021, the Zipline, an American medical product delivery company, announced that they have made 225,000 drones' deliveries containing more than 5 million units of vaccines and other medical products, reducing delivery emissions by 98% compared to traditional methods [3]. To the best of our knowledge, most of the existing academic literature focuses on the use of a drone or a fleet of drones solely for medical transportation rather than how to integrate them into traditional methods. The goal of this paper is to contribute to the investigation of possible integration, using drones and technicians in parallel to shorten the test kits collection time of the door-to-door sampling service system.

The main contributions of the paper are the following:

- We formally define and present the first formulation for the Parallel technician and-drone scheduling in the door-to-door sampling service system (PTDS-DDSS) addressing the integration of drones into existing sampling service system.
- We propose a tabu search meta-heuristic to address the problem.
- We analyze the performance of the proposed method through the comparisons with results obtained by GUROBI on small-sized instances and study the advantage of integrating drones into the Door-to-door sampling service system.

The remainder of the paper is organized as follows. Section II introduces the problem description. Section IV provides the mathematical model. Literature review is presented in Section III. The details of the proposed methodologies are described in Section V. Computational results are then reported and analyzed in Section VI. Finally, we conclude our work in Section VII.

II. Problem description

The sampling service system in the PTDS-DDSS is composed of a medical center where a set of technicians $k \in \mathcal{K}$ and a set of homogeneous drones $d \in \mathcal{D}$ are based. A number of locations where customers required to get samples are available. Customers are classified into two categories: (1) the former could be only serviced by the technician due to either the complexity of sampling technique or customer requirements,

(2) the latter could be serviced by any technician or drone. The route schedule consists of two parts. Each technician performs only one trip that departs from the medical center, gets samples from a subset of customers, then brings them back to the medical center. Since the drones and technicians do not cooperate, each drone could do multiple trips, and on each trip, it flies directly between the medical center and one or several customers. The duration of each drone's trip must not be longer than the maximum flight time L_d due to its limited battery endurance. In order to enhance the quality of the sampling service system, we assume that there is a limited allowable waiting time of each taken sample, defined as L_w. That is, the time from when the sample of each customer is taken until it is arrived at the medical center must not be higher than L_w. The following assumptions are relevant:

- Once arriving at a customer's location, the technician and drone must take sample from the customer immediately (without any delay).
- The times required by the technician and drone to collect sample from a customer, and the time to swap the drone's battery at the medical center is negligible.
- Neither technician nor drone may revisit any customers.
- The drone is assumed always to be fully recharged when leaving the medical center, and remain in constant flight while on a sortie.

All the technicians and drones must leave the medical center from time 0. The aim of this problem is to minimize the makespan of the sampling service system, i.e., the maximum working time among all technicians and drones.

Figure 1 illustrates a PTDS-DDSS solution consisting of one technician and two drones. The solid lines stand for the technician trip, while the dashed lines indicate the back-and-forth trips of the drones. The circle nodes represent customers where the filled circle nodes correspond to the customers requesting to be serviced by the technician only, while the remaining nodes correspond to the customer serviced by the technician or drones.

III. Literature review

Due to the characteristics of the PTDS-DDSS problem, this section reviews related literature in terms of research on: (1) parallel truck-and-drone scheduling problem and (2) drone applications in health-related items delivery.

A. Parallel truck-and-drone scheduling problem

The literature on the use of drones in scheduling problems could be classified into three categories: (1) a drone or several drones performing the delivery, (2) two fleets of drones and trucks operating separately in a parallel mode, (3) hybrid fleet of drones and trucks working together for delivery in synchronization scenario. The PTDS-DDSS shares the parallel setting with the second category. The idea of using drones and trucks in parallel to achieve faster deliveries was first studied in the work of Murray and Chu [4]. In their work, a single truck and a homogeneous fleet of drones service a set of customers to minimize the makespan, while there is no cooperation between drones and truck. The drones could perform back-and-forth trips between the depot and customers, but serve only one customer per trip. The authors proposed a heuristic based on constructing an initial solution in which the drones serve all the eligible customers (e.g., those with delivery demand not exceed the drone capacity) while the remaining customers are served by the delivery truck, and then applying local search to improve this solution. Our proposed PTDS-DDSS extends the work of Murray and Chu [4] by enabling multiple technicians (i.e., multiple trucks in the postal service context of [4]), and allowing each trip of drone could serve more than one customer but within the limited flight time duration. In addition, to tackle the service quality of the sampling service system, our problem considers the constraint of the allowable waiting time limit to be arrived at the medical center from the collection time for each sample. To the best of our knowledge, this is a characteristic constraint of the sampling service system, which has not been considered in the literature.

Table I summarizes the main features of the parallel truck-and-drone scheduling problem using multiple trucks and drones studied in the scientific literature, from which the comparison between the main features of our PTDS-DDSS problem with those of other problems is displayed. One could see that for studies with the objective function of minimizing the makespan, the working time limit obviously does not need to be considered. The PTDS-DDSS problem is applied in the context of a sampling service system where technicians and drones visit customers to pick up small-sized test kits such as hair, blood, and urine samples. Therefore, it is reasonable not to consider the capacity of drones and technicians in the PTDS-DDSS problem. The other problems in Table I studied postal services, the limitation on the capacity of vehicles is indeed a real-world issue that need to be considered.

B. Drone applications in health-related items delivery

The use of drones in the sector of healthcare, health-related services and disaster relief for last-mile distribution also has attracted significant attention of researchers (see the surveys of [12], [13]). However, the majority of works focus on the use of a drone or a fleet of drones solely for transportation, rather than how to integrate them into traditional methods. Only the work of Scott et al. [14] considers the hybrid mode between a truck and a drone for delivery. They study the problem where emergency medical supplies need to be delivered to an outlying area that is not completely serviced by good roads but is too far for drone delivery alone. In this scenario, a drone is used for delivery, while a truck serves as a mobile depot for the drone to reload, and the drone performs only one delivery in each trip. This problem eventually resembles the synchronization mode as mentioned earlier.

IV. Model formulation

Let $\mathcal{C} = \{1, 2, .., c\}$ represents the set of all customers and let $\mathcal{C}_1 \subset \mathcal{C}$ denotes the subset of customers that require to be serviced by technicians only. We also denote $\mathcal{C}_2 = \mathcal{C} \setminus \mathcal{C}_1$ the

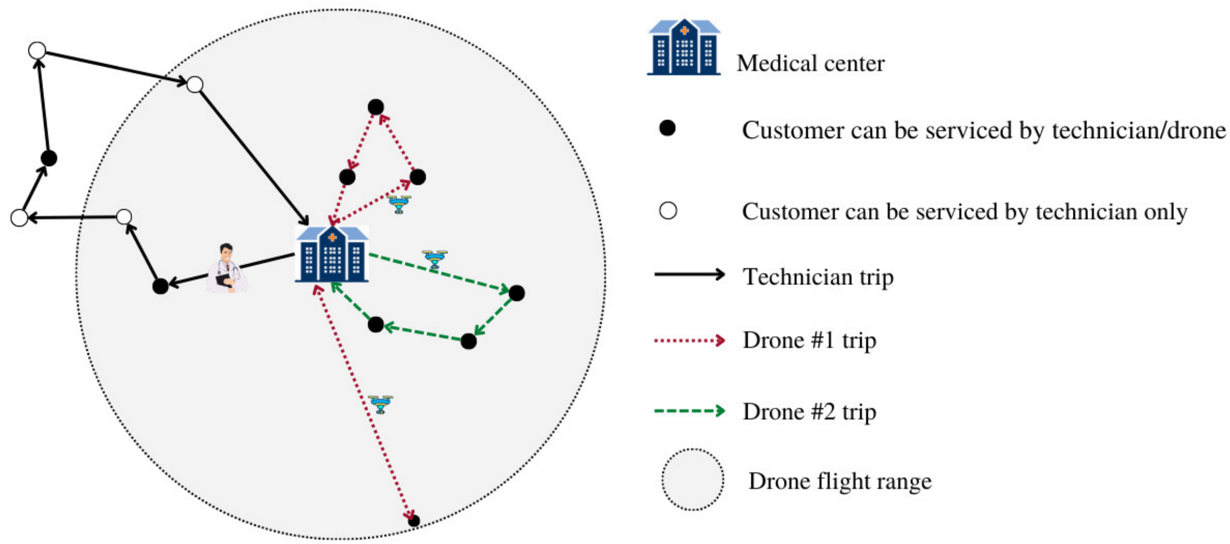

Fig. 1. An example of a PTDS-DDSS solution

TABLE I
SUMMARY OF THE RELATED WORKS

Reference	Drone duration	Drone capacity	Truck capacity	Working time limit	Multi-visit per drone trip	Multi-trip drone	Limited waiting time	Objective function	Solution method
Ulmer & Thomas [5]	yes	no	no	yes	no	yes	no	#customers visited	heuristic
Chen et al. [6]	yes	no	no	yes	no	yes	no	#customers visited	deep Q-learning
Nguyen et al. [7]	yes	yes	yes	yes	no	yes	no	operational cost	MILP, metaheuristic
Raj et al. [8]	yes	no	no	no	no	yes	no	makespan	branch-and-price
Ham [9]	yes	no	no	no	no	yes	no	makespan	constraint programming
Wang et al. [10]	yes	yes	no	no	yes	yes	no	makespan	three step heuristic
Saleu et al. [11]	yes	no	no	no	no	yes	no	makespan	branch-and-cut, local search
This work	yes	no	no	no	yes	yes	yes	makespan	MILP, tabu search

subset of customers that can be serviced by either a technician or a drone. Although a single physical medical center location exists, notationally, we assign it to two unique node numbers, such that technicians/drones depart from the medical center at node 0 and return to the medical center at node $c+1$. Thus, $N = \{0, 1, ..., c+1\}$ represents the set of all nodes in the network. The time required for the technician to travel from node $i \in N \setminus \{c+1\}$ to node $j \in N \setminus \{0\}$ is given by τ_{ij}. Parameter τ'_{ij} represents the analogous travel time for the drone. As it is assumed that each technician and drone must not revisit any node, τ_{ii} and τ'_{ii} are undefined for all $i \in N$. Note that, for the sake of completeness, $\tau_{0,c+1} \equiv 0$ and $\tau'_{0,c+1} \equiv 0$.

We use the notation summarized in Table II to formulate the problem as a MILP model mathematically. From a network perspective, the term *node* refers to a customer's location or the medical center.

$$\text{Min Max}\{A_k, B_d\} \quad \forall k \in \mathcal{K}, d \in \mathcal{D} \tag{1}$$

$$x_{ij}^k \in \{0,1\} \quad \forall i \in C \cup \{0\}, \quad \forall j \in C \cup \{c+1\}, \quad k \in \mathcal{K} \tag{2}$$

$$y_{ij}^{dr} \in \{0,1\} \quad \forall i \in C_2 \cup \{0\}, \quad \forall j \in C_2 \cup \{c+1\}, \quad d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{3}$$

$$A_k \geq 0 \quad \forall k \in \mathcal{K} \tag{4}$$

$$B_d \geq 0 \quad \forall d \in \mathcal{D} \tag{5}$$

$$T_d^r \geq 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{6}$$

$$\sum_{j \in C \cup \{c+1\}} x_{0j}^k = \sum_{i \in C \cup \{0\}} x_{i,c+1}^k \quad \forall k \in \mathcal{K} \tag{7}$$

$$\sum_{i \in C \cup \{0\}} \sum_{j \in C \cup \{c+1\}} x_{ij}^k = 0 \text{ if } \sum_{j \in C \cup \{c+1\}} x_{0j}^k = 0 \quad \forall k \in \mathcal{K} \tag{8}$$

$$\sum_{j \in C \cup \{c+1\}} x_{0j}^k \leq 1 \quad \forall k \in \mathcal{K} \tag{9}$$

TABLE II
NOTATION USED IN THE MATHEMATICAL MODEL

Parameters	
τ_{ij}	Travel time from node i to node j by technician
$\tau_{ij}^{'}$	Travel time from node i to node j by drone
L_d	Flight range limit of drone
L_w	Maximum waiting time of a sample
Sets	
C_1	Set of all customer nodes can only be serviced by technician
C_2	Set of all customer nodes an be serviced by either an technician or a drone
C	Set of all customer nodes, $C = C_1 \cup C_2$
N	Set of all nodes in the network.
$\mathcal{R}$	Set of drone trips
$\mathcal{D}$	Set of drones
$\mathcal{K}$	Set of technicians
Indexes	
i, j, z	customers and medical center (0, c+1 indicate starting and ending medical center, respectively)
r	drone trip's indexes
d	drone's indexes
k	index of technician
Variables	
x_{ij}^k	binary variable; 1 if k^{th} technician uses arc (i, j) to move from node i to node j; otherwise, 0;
y_{ij}^{dr}	binary variable; 1 if r^{th} trip of d^{th} drone uses arc (i, j) to move from node i to node j; otherwise, 0
A_k	the task completion time of the k^{th} technician
B_d	the task completion time of the d^{th} drone;
T_d^r	the task completion time of the r^{th} trip of d^{th} drone;
l_i^k	the level of node i serviced by k^{th} technician
l_i^{dr}	the level of node i serviced by d^{th} drone at r^{th} trip

$$\sum_{j \in C_2} y_{0j}^{dr} = \sum_{i \in C_2} y_{i,c+1}^{dr} \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{10}$$

$$\sum_{i \in C_2 \cup \{0\}} \sum_{j \in C_2 \cup \{c+1\}} y_{ij}^{dr} = 0 \ \text{ if } \sum_{j \in C_2} y_{0j}^{dr} = 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{11}$$

$$\sum_{j \in C_2} y_{0j}^{dr} \leq 1 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{12}$$

$$l_0^k = 0 \quad \forall k \in \mathcal{K} \tag{13}$$

$$l_j^k = l_i^k + 1 \ \text{ if } x_{ij}^k = 1 \quad \forall k \in \mathcal{K}, i \in C \cup \{0\}, \quad j \in C \cup \{c+1\} \tag{14}$$

$$l_0^{dr} = 0 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{15}$$

$$l_j^{dr} = l_i^{dr} + 1 \ \text{ if } y_{ij}^{dr} = 1 \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R}, i \in C_2 \cup \{0\}, \quad j \in C_2 \cup \{c+1\} \tag{16}$$

$$\sum_{i \in C \cup \{0\}} \sum_{k \in \mathcal{K}} x_{ij}^k + \sum_{i \in C_2 \cup \{0\}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}} y_{ij}^{dr} = 1 \quad \forall j \in C_2 \tag{17}$$

$$\sum_{i \in C \cup \{0\}} \sum_{k \in \mathcal{K}} x_{ij}^k = 1 \quad \forall j \in C_1 \tag{18}$$

$$\sum_{j \in C \cup \{c+1\}} \sum_{k \in \mathcal{K}} x_{ij}^k + \sum_{j \in C_2 \cup \{c+1\}} \sum_{d \in \mathcal{D}} \sum_{r \in \mathcal{R}} y_{ij}^{dr} = 1 \quad \forall i \in C_2 \tag{19}$$

$$\sum_{j \in C \cup \{c+1\}} \sum_{k \in \mathcal{K}} x_{ij}^k = 1 \quad \forall i \in C_1 \tag{20}$$

$$\sum_{i \in C \cup \{0\}} x_{ij}^k = \sum_{i \in C \cup \{c+1\}} x_{ji}^k \quad \forall j \in C, \quad k \in \mathcal{K} \tag{21}$$

$$\sum_{i \in C_2 \cup \{0\}} y_{ij}^{dr} = \sum_{i \in C_2 \cup \{c+1\}} y_{ji}^{dr} \quad \forall j \in C_2, \quad d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{22}$$

$$\sum_{j \in C_2} y_{0j}^{dr} \geq \sum_{j \in C_2} y_{0j}^{d,r+1} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{23}$$

$$A_k = \sum_{i \in C \cup \{0\}} \sum_{j \in C \cup \{c+1\}} x_{ij}^k t_{ij} \quad \forall k \in \mathcal{K} \tag{24}$$

$$T_d^r = \sum_{i \in C2 \cup \{0\}} \sum_{j \in C2 \cup \{c+1\}} y_{ij}^{dr} t_{ij}^{'} \quad \forall d \in \mathcal{D}, r \in \mathcal{R} \tag{25}$$

$$B_d = \sum_{r \in \mathcal{R}} T_d^r \quad \forall d \in \mathcal{D} \tag{26}$$

$$A_k - \sum_{j \in C \cup \{c+1\}} x_{0j}^k t_{0j} \leq L_w \quad \forall k \in \mathcal{K} \tag{27}$$

$$T_d^r - \sum_{j \in C_2 \cup \{c+1\}} y_{0j}^{dr} t_{0j}^{'} \leq L_w \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{28}$$

$$T_d^r \leq L_d \quad \forall d \in \mathcal{D}, \quad r \in \mathcal{R} \tag{29}$$

The objective function (1) minimizes the makespan of the sampling service system. Constraints (2) - (6) specify the domains and allowable ranges of the variables. Constraints

(7) - (12) guarantee respectively each technician/drone that leaves the medical center must return to it when completing their trips. Each trip can only leave the medical center at most once. Constraints (13) - (16) ensure that each technician and drone trip has no cycles created by visited customers. Constraints (17) - (20) ensure each customer is serviced exactly once by either technicians or drones. Constraints (21) - (22) respectively state that if a technician/drone arrives at a customer then it must leave this customer. Constraints (23) ensures the correct sequence of assignment of trips to each drone. Constraints (24) - (26) calculate the task completion time of technicians and drones. Constraints (27) - (28) ensure the limited waiting time of samples. Constraints (29) ensure limited flying time of drones.

V. Tabu search meta-heuristic

A. General structure

The overall structure of the proposed tabu search algorithm is given in Algorithm 1. The best solution z generated by three greedy methods presented in Section V-C is used as the initial solution. At each iteration of the tabu search, one neighborhood is selected randomly, then the selected neighborhood is explored, and the best move is chosen (lines 4-5). This move must not be tabu, unless it improves the current best solution z_{best}. The search is terminated when the maximum number of iterations IT_{max} is reached or after IT_{imp} iterations without improvement on the best solution (line 10). Finally, a post-optimization procedure is performed to potentially improve the current best solution z_{best} (line 11).

Algorithm 1 Tabu search

1: Generate an initial solution z
2: $z_{best} \leftarrow z$
3: **repeat**
4:     Select a neighborhood randomly
5:     Find the best solution z' (the one with minimum fitness) in the selected neighborhood of z and is not tabu or satisfies its aspiration criteria
6:     **if** z' is better than z_{best} **then**
7:         $z_{best} \leftarrow z'$
8:     **end if**
9:     $z \leftarrow z'$
10: **until** the stopping condition is met
11: $z_{best} \leftarrow$ *Post-optimization*(z_{best})
12: return z_{best}

B. Search space

A solution to our problem is a set of trips made by technicians and drones. The search space is thus made up of these feasible and infeasible trips. We do not accept the solutions in which there is any customer of set C_1 in drones' trips.

For a given solution z, let $c(z)$ denote the completion time of its trips, and let $d(z)$, $w(z)$ denote the total violation of limited drone's flight endurance L_d and maximum waiting time L_w, respectively. The total drone's flight endurance violation is computed on a drone's trip with respect to the value L_d, thus is equal to $\sum_{d \in D} \sum_{r \in \mathcal{R}^d} max\{(A_r^d - D_r^d - L_d), 0\}$ where $\mathcal{R}^d$ is the trip set of the d^{th} drone in the set D, D_r^d and A_r^d are respectively the departure time and arrival time at the medical center of each r^{th} trip in the drone's trip set $\mathcal{R}^d$. The total waiting time violation is equal to $\sum_{c \in C} max\{(A_c - D_c - L_w), 0\}$ where D_c and A_c are respectively the time at which the sample is taken and the arrival time at the medical center of the customer $c \in C$.

The quality of a given solution z is then evaluated by an weighted fitness function $f(z)$ which is a linear combination of the objective function $c(z)$ considering the completion time of the system with a penalty function considering the degree of solution infeasibility: $f(z) = c(z) + \alpha_1 d(z) + \alpha_2 w(z)$ where α_1, α_2 are penalty parameters adjusted dynamically during the search. The updating scheme is based on the idea of Cordeau et al. [15]. At each iteration, the value of α_1, α_2 are modified by a factor $1 + \beta > 1$. If the current solution is feasible with respect to drone's flight endurance constraint, the value of α_1 is divided by 1 + β; otherwise it is multiplied by 1 + β. The same rule applies to α_2 with respect to maximum waiting time constraint. In our algorithm, we set $\alpha_1 = \alpha_2 = 1$ and $\beta = 0.5$.

C. Initial solution

In order to generate a good initial solution for the tabu search, we use three greedy heuristics to generate three solutions, and the best generated one will be selected.

- Method 1 attempts to construct a solution with minimum total travel distance and keep the technicians and drones working as much as possible. In the first phase, we generate a trip consisting only one customer for each of $|\mathcal{D}|$ drones and $|\mathcal{K}|$ technicians. Thus, a number of $|\mathcal{D}|$ customers in the range of drone flight and closest to the medical center are assigned as starting points in the first trip of each drone, and similarly, each of $|\mathcal{K}|$ unassigned customers closest to the medical center is added to the trip of a technician. In the second phase, each of $|\mathcal{D}| + |\mathcal{K}|$ generated trips is then continue to be built in turn. At each iteration, an unassigned-customer is selected to insert into the current considered trip if it is closet to the last customer and the insertion does not make the violation of any problem constraints. The customers not yet assigned left at the end of second phase, if any, are then shuffled and added sequentially to the end of existing trips of $|\mathcal{D}|$ drones in case maximum flight time L_d is not violated, otherwise a new trip of the drone with smallest completion time is created and the customer is added to this new trip.
- Method 2 aims to build the solution to minimize the time from collection to arrive at the medical center of all samples as much as possible. It thus follows the same process as method 1, but the initial customers in the first phase are selected from those farthest to the medical center.
- Method 3 follows the cluster first-route second scheme. During clustering, customers are first sorted in increasing order of the angle they make with the medical center.

Next, a customer j is chosen randomly and all $|\mathcal{C}|$ customers in the order of j, $j+1$,..., $|\mathcal{C}|$, 1,..., $j-1$ are added sequentially to $|\mathcal{K}|$ technician trips and $|\mathcal{D}|$ drone trips in turn until there is a violation on problem constraints. If there exists customers not yet assigned once the $|\mathcal{K}|+|\mathcal{D}|$ trips are created, we follow the same process as in method 1 to add them to the trips of drones.

One could see that method 1 could provide good solution when customers are evenly distributed around the medical center (see Figure 2a). However, if the customers distributed around main roads (see Figure 2b), trips in solutions generated by method 1 will consisting sequence of customers in the ascending order of the distance to the medical center, which eventually reduces the number of customers serviced during the trip, otherwise the waiting time of first customer's sample might be exceeded the limitation L_w. Method 2 thus builds trips servicing customers in reversed order to prevent the violation on L_w of samples. For example, if applying method 2, all customers 1, 2, 3, 4, 5 in Figure 2b could be serviced in one trip {medical center $\rightarrow$ 4 $\rightarrow$ 2 $\rightarrow$ 5 $\rightarrow$ 3 $\rightarrow$ 1 $\rightarrow$ medical center}. However, when applying method 1, it might need two trips {medical center $\rightarrow$ 1 $\rightarrow$ 3$\rightarrow$ 5 $\rightarrow$ medical center} and {medical center $\rightarrow$ 2 $\rightarrow$ 4 $\rightarrow$ medical center}, as the trip {medical center $\rightarrow$ 1 $\rightarrow$ 3 $\rightarrow$ 5 $\rightarrow$ 2 $\rightarrow$ 4 $\rightarrow$ medical center} makes the violation on the limited waiting time L_w of customer 1 and 3. Similarly, it could generate two trips {medical center $\rightarrow$ 1 $\rightarrow$ 2$\rightarrow$ 3 $\rightarrow$ medical center} and {medical center $\rightarrow$ 4 $\rightarrow$ 5 $\rightarrow$ medical center} when applying method 3, as the trip {medical center $\rightarrow$ 1 $\rightarrow$ 2 $\rightarrow$ 3 $\rightarrow$ 4 $\rightarrow$ 5 $\rightarrow$ medical center} makes the violation on the limited waiting time L_w of customer 1 and 2. In the case locations of customers follow the ring distribution (see Figure 2c), method 3 should be applied.

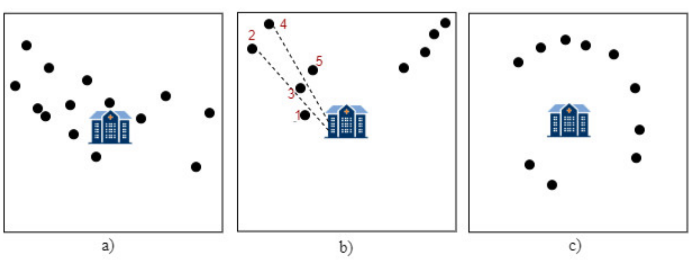

Fig. 2. Illustration of customer distribution

D. Neighborhoods

Using one neighborhood only is not sufficient for the problem setting at hand. We have created five efficient inter- and intra-route neighborhoods. It not only enables each neighborhood to transfer information to the others, but also increases substantially the capability of the algorithm to uncover new search space. The five neighborhoods are described as following:

- *(1,0) move*: For every two customers x and y, the customer x is taken from its current position and inserted after y.
- *(1,1) move*: two customers x and y are swapped their positions.
- *(2,0) move*: two adjacent customers x, x' in the same trip are taken from their current positions and inserted after y.
- *(2,1) move*: swap two adjacent customers x, x' in a same trip with another customer y.
- *2-opt move*:
  - For every two customers x and y in a same trip of each technician/drone, the edges (x, x'), (y, y') emanating from them are removed, two edges (x, y), (y, y') are added: one of which connects these two customers, and the other connects their successor customers. Figure 3 illustrates an example of 2-opt on the same trip. It means the order of customers in segment $x'..y$ is reversed.
  - For every two customers x, y in different trips of technicians/drones, the route segments following them are swapped preserving the order of customers succeeding them in each route (see Figure 4). In this case, we evaluate all following possibilities: x and y serviced by two technicians, x and y serviced by different trips of the same drone or two drones, x and y serviced by a technician and a drone.

In all neighborhoods, we will not evaluate the case creating any solution in which customers of set C_1 belongs to drones' trips.

E. Tabu list and tabu duration

We keep a separate tabu list for each type of move. Elements of a solution generated by a move are given a tabu as follows:

- *(1,0) move*: the position of customer x just inserted after customer y cannot be changed by the same type of move while it is tabu.
- *(1,1) move*: two customers x and y just swapped cannot be swapped again while they are tabu.
- *(2,0) move*: the positions of customers x, x' just inserted after customer y, cannot be changed by the same type of move while they are tabu.
- *(2,1) move*: customers x, x', and y just swapped cannot be swapped again while they are tabu.
- *2-opt move*: a 2-opt move applied to customers x and y cannot be applied again to the same customers while they are tabu.

Each tabu status is stored in the corresponding tabu list for θ iterations, where θ is a predefined parameter. In our tabu search, a move declared tabu is accepted only if it improves the current best solution z_{best}.

F. Post-optimization

The best solution obtained through the tabu search is enhanced by applying a number of well-known local search route

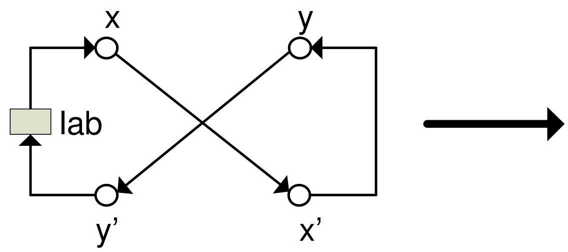

Fig. 3. 2-opt on the same trip of a technician/drone

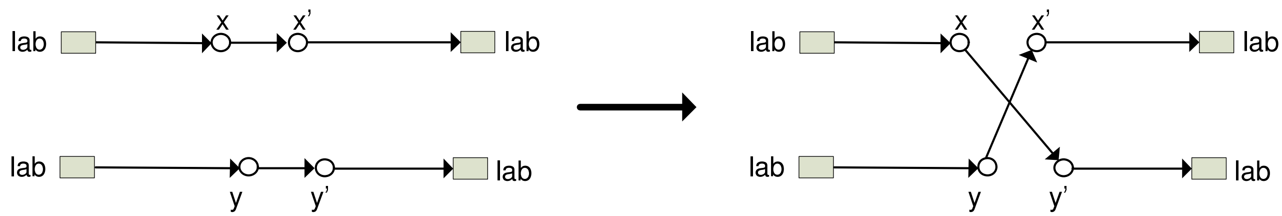

Fig. 4. 2-opt on the different trips of technicians/drones

improvement techniques on trips of technicians and drones. Four are intra-route operators: the relocate, the exchange, the 2-opt of [16] and the Or-opt of [17] (see Figure 5). The others are inter-route operators: the relocate, the exchange, the Or-opt of [17], the 2-opt* of [18], the CROSS-exchange of [19] and the Ejection chain of [20] (see Figure 6). In order to save the computation time, we only consider the case where the length of the shift sequence is limited to two (i.e., only change positions of 2 customers) in the ejection chain operator.

The post-optimization procedure starts by applying in random order six inter-route operators. Each neighborhood is searched on all possible pairs of technician/drone trips (in random order). The solution is then modified and the process is repeated until no further improvement can be found. The search is then continued by locally improving each trip of a technician/drone in turn. The intra-route neighborhoods are sequentially and repeatedly applied until no more improvement is found.

VI. Computational results

Our tabu search meta-heuristic was implemented in Python. Experiments were run on an Intel Core i9-12900K 3.20 GHz. The 80 instances generated by Sacramento et al. [21], were used throughout the experiments. The number of customers is ranged from 6 to 100. The medical center is located at [0,0] in all instances. Customers in each instance are generated following the uniform distribution $U(-d/2, d/2)$ in a grid size $d \times d$, ranging between 5×5 and 40×40. The value of limited waiting time L_w of each sample is set to 60 minutes. The instance sets and detailed results can be accessed via https://github.com/huetran1611/PTDS-DDSS-problem. The user-defined parameters (IT_{max}, IT_{imp}, θ) were set to (500, 50, 5) for all instances. The objective of the computational experimentation is threefold. First, to study the impact of neighborhood selection strategies on the performance of the proposed tabu search in order to identify the most efficient one. The second objective consists in evaluating the performance of tabu search through comparison with GUROBI results. We finally analyze the benefits of integrating drones on solution quality.

A. Calibration of the neighborhood selection strategies

The neighborhood selection strategy specifies which neighborhood is chosen at each iteration. We studied two strategies, cycle and random:

- The former creates a list of neighborhoods ordered in ascending order of move size which are (1, 0) move, (1, 1) move, (2, 0) move, (2, 1) move, 2-opt move. The neighborhood in this list is selected sequentially, and only one neighborhood is explored at each iteration. Once the last neighborhood in the list is explored, the search is resumed with the first one.
- The latter selects a neighborhood randomly at each iteration to explore.

Table III reports comparison results between these two strategies on small-sized instances. For each strategy, the Table shows the average cost of best solutions (Column Best), standard deviation (Column Std), and average computation time in second (Column Time(s)) over 10 runs for each instance set of same size. Columns $GAP_1(\%)$ and $GAP_2(\%)$ display respectively the average gaps of the best solutions obtained by cyclic strategy and GUROBI solutions with respect to the best solutions obtained by random strategy. The values in the Column $GAP_1(\%)$ indicate the better performance of random strategy compared to the cyclic one over all small-sized instance, with an average GAP of -1.44%. This result

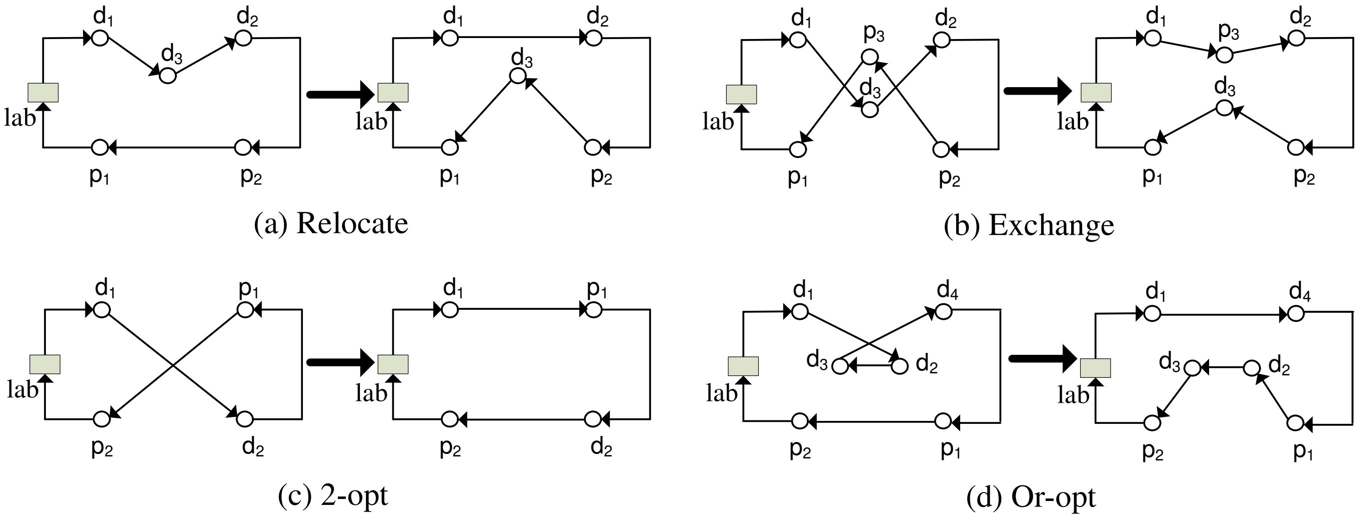

Fig. 5. Intra-route operators

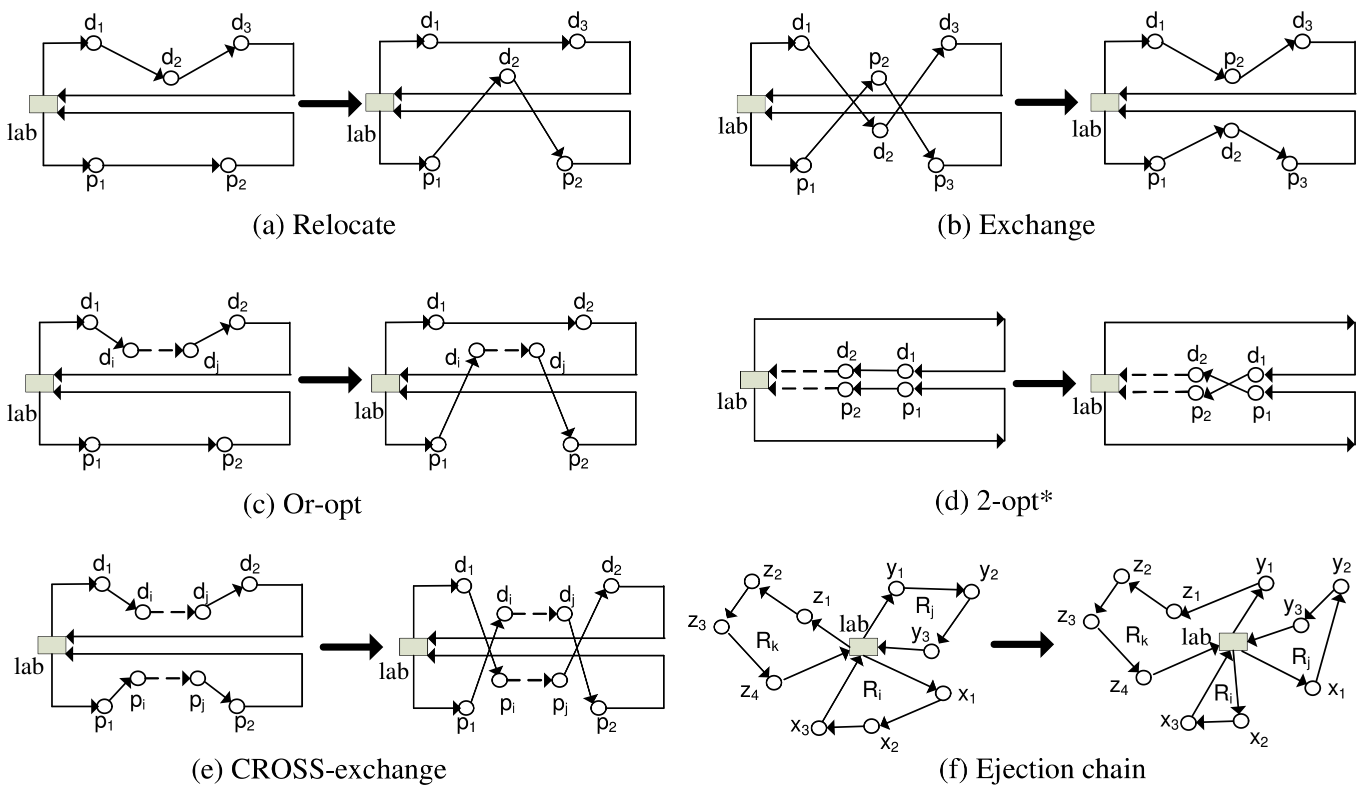

Fig. 6. Inter-route operators

eventually supports our selection, i.e., a neighborhood is selected randomly at each iteration in our tabu search meta-heuristic.

B. The performance of the proposed MILP and tabu search meta-heuristic

The mathematical model presented in Section IV has been solved using GUROBI for 5 hours. From the results, it could be seen that when the number of customers increases, the mathematical model becomes more difficult to solve to

TABLE III
COMPARATIVE PERFORMANCES BETWEEN NEIGHBORHOOD SELECTION STRATEGIES

	Cyclic strategy			Random strategy				
Instance set	Best	Std	Time(s)	Best	Std	Time(s)	$GAP_1(\%)$	$GAP_2(\%)$
6-customer	24.12	0.04	1.32	24.12	0.84	1.40	0	0
10-customer	29.66	0.02	5.07	29.05	0.76	6.20	-2.06	0
12-customer	32.95	0.00	8.11	32.72	0.71	10.44	-0.70	0
20-customer	22.45	0.00	51.01	21.78	0.77	39.96	-2.98	-8.72
Average	27.30	0.015	16.38	26.92	0.77	14.50	-1.44	-2.18

optimality in a reasonable amount of time. Actually, it is able to obtain the optimal solutions within only 2 minutes on instances with 6, 10, 12 customers. However, GUROBI spends 5 hours but could not solve to optimality instances with 20 customers.

From the result displayed in the column $GAP_2(\%)$ of Table III, it could be seen that the optimal solution can be reached in all cases by the tabu search for instances with 6, 10, 12 customers. Moreover, for 20-customer instances, the tabu search could obtain even better results than GUROBI, with an average gap of 8.72%. This proves the effectiveness of our proposed tabu search. Noteworthy, the post-optimization process helped to improve solution quality by 0.17% on average, but requiring 85.81% of running time.

C. The benefits of integrating drones

To evaluate the value of integration of drone into the sampling system, we considered two scenarios: 1) $\mathcal{K}$ technicians + $\mathcal{D}$ drones, using $||\mathcal{K}||$ technicians and $||\mathcal{D}||$ drones; 2) $\mathcal{K}$ + $\mathcal{D}$ technicians + 0 drones, using $||\mathcal{K}||$ + $||\mathcal{D}||$ technicians and no drones. However, the second scenario could not provide feasible solution in several instances. In this case, we run another experiment with additional technicians to get feasible solutions. We therefore provide results in two Tables. The Table IV shows results for instances where feasible solutions could be obtained for the second scenario, while the Table V for instances where additional technicians are needed to get feasible solutions.

Table IV compares the average best results for instances provided feasible solutions over 10 runs of these scenarios. The average best objective value of the integrated scenario $\mathcal{K}$ technicians + $\mathcal{D}$ drone, i.e., the PTDS-DDSS problem, is given in column *Best*. The *GAP(%)* column displays the gaps for the average best objective values obtained by the second scenario with respect to that of the first scenario $\mathcal{K}$ technicians + $\mathcal{D}$ drone. The Table IV indicates that the integration of drone provides better solutions compared to the case using technicians only. More specifically, replacing all $||\mathcal{D}||$ drones by $||\mathcal{D}||$ technicians makes an increase in the objective value of 25.14%.

Table V compares the average best results for instances provided infeasible solutions as mentioned earlier over 10 runs between two scenarios using and without using drones. The first column shows the name of instance in the format of $n.m.t$, where n is the number of customers, m is the dimension of the grid and t is the generic name of the scenario in which Sacramento et al. [21] used to generate instances.

TABLE IV
COMPARISON OF WITH AND WITHOUT DRONE ON FEASIBLE INSTANCES

Instance set	$\mathcal{K}$ technicians + $\mathcal{D}$ drones	$\mathcal{K}$+$\mathcal{D}$ technicians + 0 drones
	Best	GAP(%)
6-customer	24.12	32.98
10-customer	29.05	31.13
12-customer	32.72	22.58
20-customer	21.78	26.12
50-customer	50.93	18.12
100-customer	52.63	27.22
Average	35.21	25.14

The second and third columns display three values: number of technicians, number of drones used in the instance and the cost of best solution. The second column corresponds to the PTDS-DDSS problem where the drones are used, while the third column shows the minimum number of technicians need to be used to get feasible solution if drones are not used and the corresponding best solution obtained. The last column displays the gaps of the best solutions obtained by the scenario not using drone with respect to those of scenario using drone. The results show that even using additional technicians, we could not obtain better results compared to the scenario using drones in some instances. The reason could be when the number of customers in the range of drone flight increases, the advantage of the drone's flight speed in transportation is brought into effect.

TABLE V
COMPARISON OF WITH AND WITHOUT DRONE ON INFEASIBLE INSTANCES

Instance	(#technicians & #drones), Best	(#technicians & 0 drones), Best	GAP (%)
12.20.1	(2 & 2), 39.06	(5 & 0), 43.27	10.78
12.20.2	(2 & 2), 44.54	(5 & 0), 50.54	13.47
20.20.2	(2 & 2), 36.90	(5 & 0), 48.59	31.68
20.20.3	(2 & 2), 42.38	(5 & 0), 47.66	12.46
50.40.1	(3 & 3), 77.73	(9 & 0) 97.09	24.91
100.30.1	(4 & 4), 60.97	(10 & 0) 84.21	38.12
100.40.1	(4 & 4), 76.55	(13 & 0) 101.26	32.28
Average	54.02	67.52	23.39

D. The detailed results

Table VI summarizes the results of our proposed tabu search for all 80 instances grouped by number of customers. The algorithm was run 10 times per instance. The average best objective values, standard deviations, and computation times in seconds are reported.

TABLE VI
DETAILED RESULTS

Instance set	Best	Std	Time(s)
6-customers	24.12	0.84	1.4
10-customers	29.05	0.76	6.20
12-customers	32.72	0.71	10.44
20-customers	21.78	0.77	39.96
50-customers	50.93	1.74	79.61
100-customers	52.63	1.58	1875.91
Average	35.20	1.07	335.59

VII. Conclusion

We introduce the PTDS-DDSS problem addressing the integration of drone into existing sampling service system. A MILP is proposed to formulate mathematically the problem, from which small instances can be solved to optimality. A tabu search with multiple neighborhoods is also introduced to solve the problem. Experimental results illustrated clearly the superior performance of the proposed drone integration compared to the conventional technician alone sampling service system.

Acknowledgment

This work was funded by Vingroup and supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2022.DA00183.

References

[1] A. Otto, N. Agatz, J. Campbell, B. Golden, and E. Pesch, "Optimization approaches for civil applications of unmanned aerial vehicles (uavs) or aerial drones: A survey," *Networks*, vol. 72, no. 4, pp. 411–458, 2018.

[2] "Intricacies of medical drones in healthcare delivery: Implications for africa," *Technology in Society*, vol. 66, p. 101624, 2021.

[3] "Zipline receives the 2021 u.s. secretary of state's award for corporate excellence in health security," *SUAS News*, December 9, 2021. [Online]. Available: https://www.suasnews.com/2021/12/zipline-receives-the-2021-u-s-secretary-of-states-award-for-corporate-excellence-in-health-security/

[4] C. C. Murray and A. G. Chu, "The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 86–109, 2015.

[5] M. W. Ulmer and B. W. Thomas, "Same-day delivery with heterogeneous fleets of drones and vehicles," *Networks*, vol. 72, no. 4, pp. 475–505, 2018.

[6] X. Chen, M. W. Ulmer, and B. W. Thomas, "Deep q-learning for same-day delivery with vehicles and drones," *European Journal of Operational Research*, vol. 298, no. 3, pp. 939–952, 2022.

[7] M. A. Nguyen, G. T.-H. Dang, M. H. Hà, and M.-T. Pham, "The min-cost parallel drone scheduling vehicle routing problem," *European Journal of Operational Research*, vol. 299, no. 3, pp. 910–930, 2022.

[8] R. Raj, D. Lee, S. Lee, J. Walteros, and C. Murray, "A branch - and - price approach for the parallel drone scheduling vehicle routing problem," July 2021. [Online]. Available: https :// ssrn . com / abstract =3879710

[9] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.

[10] D. Wang, P. Hu, J. Du, P. Zhou, T. Deng, and M. Hu, "Routing and scheduling for hybrid truck-drone collaborative parcel delivery with independent and truck-carried drones," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 483–10 495, 2019.

[11] R. G. M. Saleu, L. Deroussi, D. Feillet, N. Grangeon, and A. Quilliot, "The parallel drone scheduling problem with multiple drones and vehicles," *European Journal of Operational Research*, vol. 300, no. 2, pp. 571–589, 2022.

[12] A. Rejeb, K. Rejeb, S. Simske, and H. Treiblmaier, "Humanitarian drones: A review and research agenda," *Internet of Things*, vol. 16, p. 100434, 2021.

[13] B. Hiebert, E. Nouvet, V. Jeyabalan, and L. Donelle, "The application of drones in healthcare and health-related services in north america: A scoping review," *Drones*, vol. 4, no. 3, 2020.

[14] J. Scott and C. Scott, "Drone delivery models for healthcare," in *Proceedings of the 50th Hawaii international conference on system sciences*, 2017.

[15] J.-F. Cordeau, G. Laporte, and A. Mercier, "A unified Tabu Search heuristic for vehicle routing problems with time windows," *Journal of the Operational Research Society*, vol. 52, pp. 928–936, 2001.

[16] S. Lin and B. W. Kernighan, "An effective heuristic for the traveling salesman problem," *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.

[17] I. Or, "Traveling Salesman-type Combinatorial Problems and their relation to the Logistics of Blood Banking," Ph.D. dissertation, Department of Industrial Engineering and Management Science, Northwestern University, Evanston, IL, 1976.

[18] J.-Y. Potvin and J.-M. Rousseau, "An exchange heuristic for routing problems with time windows," *Journal of the Operational Reasearch Society*, vol. 46, pp. 1433–1446, 1995.

[19] E. D. Taillard, P. Badeau, M. Gendreau, F. Guertin, and J.-Y. Potvin, "A tabu search heuristic for the Vehicle Routing Problem with soft time windows," *Transportation Science*, vol. 31, pp. 170–186, 1997.

[20] F. Glover, "New ejection chain and alternating path methods for traveling salesman problems," in *Proceedings of the ORSA Meetings in computer Science in Operations Research, Williamsburg, VA*, 1992.

[21] D. Sacramento, D. Pisinger, and S. Ropke, "An adaptive large neighborhood search metaheuristic for the vehicle routing problem with drones," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 289–315, 2019.