A bi-objective Medical Sampling Service System

Anonymous Author(s)

Abstract

Drones are increasingly adopted in logistics due to their flexibility and efficiency, presenting novel solutions for complex service systems. This study develops a bi-objective optimization model for medical sampling service systems, where technicians and drones independently collect samples from patient locations. The model addresses critical challenges by simultaneously minimizing system completion time and patient test kit waiting duration, while incorporates realistic constraints, such as technicians' dynamic velocity variations reflecting traffic conditions and drone energy consumption dependent on load. A hybrid algorithm combining Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Tabu Search is proposed, ensuring elite solution preservation and genetic diversity. Experimental validation demonstrates better performance compared to NSGA-II, Tabu Search, and Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) across hypervolume and non-dominated solution metrics, highlighting the approach's effectiveness in optimizing medical sampling logistics with disease outbreaks serving as a compelling use case.

ACM Reference Format:

Anonymous Author(s). 2025. A bi-objective Medical Sampling Service System. In *Proceedings of The Genetic and Evolutionary Computation Conference 2025 (GECCO '25).* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

Remarkable advances in robotics over the past decade have driven the widespread adoption of drones in various sectors including logistics, healthcare, disaster management, agricultural operations, and security surveillance systems [1]. In particular, the e-commerce boom has especially accelerated research into drone-based last-mile delivery, with industry leaders in the logistics sector like Amazon, DHL, Alibaba pioneering drone delivery trials to enhance their logistics networks. Drones present distinct operational advantages over traditional delivery methods, mainly due to their ability to navigate through direct trajectories independent of existing road infrastructure and their immunity to terrestrial traffic congestion. The implementation of drone-based delivery systems demonstrates the potential for a paradigm shift in logistics, offering threefold benefits: reduced operational expenditure for service providers, enhanced delivery speed for consumers, and minimized environmental impact through reduced carbon emissions.

GECCO '25, July 14–18, 2025, Málaga, Spain

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM

https://doi.org/10.1145/nnnnnnn.nnnnnnn

In addition, drones have emerged as a critical tool in healthcare logistics, enabling rapid delivery of essential medical supplies including first aid equipment, medical devices, biological materials, and protective gear during emergencies [2]. A notable example is Zipline, an American medical delivery company that achieved a significant milestone in December 2021 by completing 225,000 drone deliveries, transporting over 5 million units of vaccines and medical supplies while reducing delivery-related emissions by 98% compared to conventional methods [3]. In academia, researchers have begun developing frameworks for drone-based medical services. For instance, in 2022, Manh et. al [4] expanded on the parallel drone scheduling Traveling Salesman Problem (TSP) model proposed by Murray and Chu [5] to develop an innovative system where drones and technicians work collaboratively to collect patient test kits. Their model allows for multiple patient visits per drone flight while upholding service quality through waiting time constraints on test kits. Their experiments demonstrated that the proposed drone integration significantly reduced the test kits' collection time compared to the traditional technician-only sampling system.

This research extends the framework of Manh et. al [4] by introducing the Medical Sampling Service System with Variable Technician Speed and Drone Energy Consumption (MSSVTDE). The proposed framework incorporates three extensions: the integration of time-varying speed constraint for technicians, a comprehensive drone energy consumption model, and a bi-objective optimization approach that simultaneously minimizes system completion time and patient test kits waiting duration, striving to balance operational efficiency with service quality. Our contributions are both in adapting NSGA-II and tabu search operators to the particular requirements of the MSSVTDE and in designing the overall organization of the hybrid algorithm to answer the challenges of this problem setting. Experimental results validate the approach's efficacy, demonstrating better performance compared to NSGA-II, Tabu Search, and MOEA/D across hypervolume and non-dominated solution metrics.

In the following, section 2 introduces problem description and section 3 presents the related works. Section 4 describes the proposed hybrid NSGAII-Tabu algorithm. Computational results are reported and analyzed in section 5, while the conclusions of this study are considered in section 6.

2 Problem description

The medical sampling service system in the MSSVTDE 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 C where patients required to get samples are available. Patients are classified into two categories: (1) the former could be only serviced by the technician due to either the complexity of sampling technique or patients 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 patients, then brings them back to the medical center. Each drone could do multiple trips, and on each trip, it flies directly between the medical center and one or several patients. The time a technician or a drone service patient $i \in C$ is σ_i and σ'_i, respectively. The weight of the sample taken from patient $i \in C$ is given by w_i. All the technicians and drones must leave the medical center from time 0. The aim of this problem is to simultaneously minimize the system completion time and the total of patient test kits waiting duration. The latter is quantified as the temporal differential between specimen collection and arrival at the medical center. The following assumptions are made:

- Once arriving at a patient's location, the technician and drone must take sample from the patient immediately (without any delay).
- Neither technician nor drone may revisit any patients.
- Technicians are capable of transporting samples without concern for their weight. Their velocities vary according to the time frame, reflecting real-life traffic congestion. A detailed discussion of the velocity variants is provided in section 2.1.
- The drone is assumed to be fully recharged to energy level E upon departure from the medical center. Endurance calculations are provided in section 2.2.

2.1 Technician time-dependent speed

In practical applications, technician travel times are substantially affected by dynamic traffic conditions within the road network. To accurately represent these temporal variations in travel speeds, we implement the framework developed by Vuong et al.[6], which discretizes the operational day into L distinct temporal intervals $[T_l, T_{l+1}]$, where $l = 0, \ldots, L-1$. This framework enables the modeling of technician velocities as a function of time-variant traffic conditions. A significant characteristic of this model is that technician speeds may fluctuate along identical road segments when traversing temporal interval boundaries. The time-dependent travel speed between locations i and j during interval l is mathematically expressed as: $v_{ijl} = \theta V$, where V represents the baseline average speed of the technician, and θ denotes a congestion factor bounded by $[F_L, F_U]$, with F_L and F_U corresponding to the lower and upper bounds of the congestion coefficient, respectively. The magnitude of θ inversely correlates with congestion levels. During peak traffic periods, θ approaches F_L, indicating maximum congestion, while during off-peak hours, θ tends toward F_U, representing optimal traffic flow conditions.

2.2 Drone energy consumption model

Each drone trip originates from the medical center, where drones collect samples from one or multiple patients before returning to the center. To demonstrate the energy consumption model, we analyze a case where the drone visits patients i and j sequentially. The complete trip consists of 11 distinct operational phases, each characterized by specific payload and velocity as shown in the Figure 1. The drone operates at three predetermined velocities: v_t for takeoff operations, v_c for horizontal cruise flight, and v_l for landing maneuvers. The phases are as follows: (1) Initial Ascent Phase: The drone executes a vertical takeoff to altitude h with zero payload ($w = 0$) at velocity v_t; (2) First Transit Phase: Horizontal flight from medical center to patient i at cruise velocity v_c, maintaining zero payload; (3) First Landing Phase: Controlled descent to patient i at velocity v_l with no payload; (4) First Collection Phase: Sample collection at patient location i, transitioning to payload ($w = w_i$); (5) Second Ascent Phase: Vertical climb to cruise altitude h at velocity v_t with payload w_i; (6) Second Transit Phase: Horizontal flight to patient j at velocity v_c carrying payload w_i; (7) Second Landing Phase: Descent to patient j at velocity v_l while maintaining payload w_i; (8) Second Collection Phase: Additional sample collection at location j, increasing payload to $w = w_i + w_j$; (9) Final Ascent Phase: Takeoff from patient j with combined payload $(w_i + w_j)$ at velocity v_t; (10) Return Transit Phase: Horizontal return flight to medical center at cruise velocity v_c with full payload; (11) Final Landing Phase: Terminal descent at the medical center with payload $(w_i + w_j)$ at velocity v_l. The drone employs a linear energy consumption

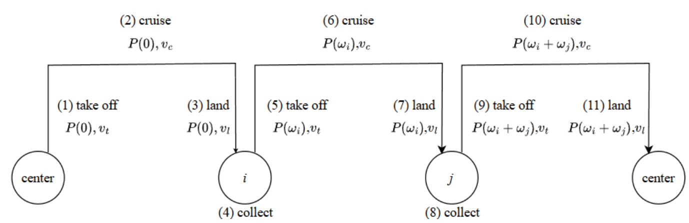

Figure 1: Illustration about drone energy consumption.

model that correlates with payload mass but demonstrates independence from velocity parameters, building upon the theoretical framework established by Dorling et al. [7]. The model assumes uniform power consumption characteristics across all flight phases, expressed mathematically as: $P(\omega) = \beta\omega + \gamma$ where: β represents the power consumption coefficient per unit payload mass, γ denotes the baseline power requirement for an unloaded drone, ω represents the current payload mass.

3 Literature review

With the rise of the low-altitude economy, driven by technological advancements and strategic policy support from the government, the potential for drone–vehicle cooperative delivery systems has significantly increased. According to survey of Macrina et al. [8], the use of drones in routing and scheduling problems can be classified into two main categories: (1) using only drones for deliveries, (2) drones and trucks completing their missions together. The idea of creating an integrated model using drones and ground vehicles to achieve faster deliveries was first introduced by Murray and Chu [5] in 2015. From an operations research perspective, the authors presented two new variants extending the traditional Traveling Salesman Problem (TSP): the Flying Sidekick TSP (FSTSP) and the Parallel Drone Scheduling TSP (PDSTSP). In the PDSTSP, a single truck and a homogeneous fleet of drones operate independently to minimize the completion time for servicing customers. The drones perform back-and-forth trips between the depot and customers but can serve only one customer per trip. The integration of drones into the Medical sampling service system in the MSSVTDE is a variant of PDSTSP, where drones and technicians operate in parallel.

Many attempts have been made to solve larger PDSTSP instances efficiently. Saleu et al. [9] introduced an iterative two-step heuristic.

The first step, known as the coding step, converts a solution into a customer sequence, while the second step, referred to as the decoding step, breaks down this sequence into a truck tour and drone trips. Dell'Amico et al. [10] proposed a matheuristic approach in which, in the first step, a customer sequence (also called a giant or TSP tour) is constructed to visit all customers. In the second step, a MILP approach is applied to divide the giant tour into a truck route and a set of drone trips, resulting in a complete PDSTSP solution. A nature-inspired algorithm called Hybrid Ant Colony Optimization (HACO) was proposed in [11]. Nguyen et al. [12] developed an exact branch-and-cut algorithm. Some studies extend the PDSTSP by incorporating multiple trucks and drones working together [13] and by modifying the objective function [14]. In these studies, a single truck is employed and performs only one trip, while drones conduct multiple trips but are restricted to serving one customer per trip. In contrast, the MSSVTDE allows drones to visit multiple customers within a single trip and incorporates limited waiting times to enhance the overall service quality of the system. The deployment of multi-point drone collection for medical specimens could be suited to structured healthcare environments such as drive-through testing facilities, community health centers, and pharmacies. In these environments, qualified healthcare professionals oversee all aspects of specimen handling and drone operations, eliminating direct patient-drone interactions. This model facilitates secure and efficient multi-point collection, preserving both operational integrity and sample quality—a framework particularly valuable for large-scale health screening initiatives and distributed diagnostic services across healthcare networks.

Ulmer and Thomas [15] introduced a problem in the context of same-day delivery, focusing on the dynamic service challenge with the objective is maximizing the number of customers served. In their model, both trucks and drones are allowed to make multiple trips. However, while trucks can visit multiple points in a single trip, drones are restricted to serving only one customer per trip. In the MSSVTDE, patient orders are static, drones could serve multiple patients in a single trip. Ham [16] extended the PDSTSP by introducing two types of drone tasks: drop-off and pickup. In this model, drones can make multiple visits, but the maximum number of customers served per trip is limited to two, with each drone having a single-unit capacity. After completing a drop-off, the drone can either fly directly to the next customer to pick up a parcel or return to the depot to begin a new trip. In the MSSVTDE, the number of patients a drone can serve per trip is not predetermined. Instead, it is contingent upon the total weight of the test kits collected from patients during a single trip, which must not exceed the drone's capacity, as well as the drone's energy levels.

The transportation systems in the studies mentioned above primarily focus on delivery tasks. Only Ham's research explores a system where vehicles handle both pick-up and delivery operations. Our transportation system, however, focuses solely on pick-up tasks. Table 1 summarizes the main features of the parallel truck-and-drone scheduling problem in the scientific literature. From which the comparison between the main features of our MSSVTDE problem with those of other problems is displayed.

Previous studies have primarily focused on single-objective optimization models for the vehicle routing problem with drone delivery, targeting economic benefits such as minimizing delivery costs or travel time. Das et al. [20] were among the first to propose a bi-objective model that minimizes travel costs while maximizing customer service levels through timely delivery, emphasizing the balance between economic efficiency and service quality. Similarly, Zhang et al. [21] expanded this approach by considering three objectives: minimizing total delivery costs, total delivery time, and total energy consumption, addressing environmental concerns alongside economic and temporal factors. In the domain of humanitarian logistics, Lu et al. [22] introduced a pickup and delivery problem with two objectives: minimizing the maximum cooperative routing time and maximizing the minimum demand fulfillment rate. Kuo et al. [23] developed a bi-objective model that minimizes the makespan of the delivery and total carbon emission. Further advancements were made by Lu et al. [24], who explored the synchronization of vehicles and drones under time-varying weather conditions. Their model sought to minimize delivery completion time across all synchronized routes while reducing disparities in demand fulfillment rates. Despite these significant contributions, current multi-objective VRP models with drone integration do not explicitly address minimizing total waiting time as a primary objective. The problem proposed in this study distinguishes itself by incorporating both the minimization of completion time and total waiting time, thereby addressing both operational efficiency and patient satisfaction. Moreover, while the above models focus on synchronized operations between trucks and drones, the MSSVTDE represents the first variant of the PDSTP framework to incorporate multi-objective optimization.

4 The proposed hybrid NSGA-II and tabu search

This section is dedicated to introducing the Hybrid NSGA-II and tabu search algorithm (HNSGAII-TS) we propose. We start by describing the general structure of the HNSGAII-TS. The individual representation (section 4.2) is introduced next. The algorithm uses one population only, which contains only feasible individuals. The initial population is created using a greedy heuristic (section 4.3). A new population is generated from the current one through the non-Dominated sorting, parent selection, offspring generation, replacement, and education phases of the algorithm (section 4.4 to 4.8, respectively).

4.1 HNSGAII-TS general structure

The proposed HNSGAII-TS algorithm employs a hybrid evolutionary approach, as depicted in Algorithm 1. The method commences with the initialization of N feasible solutions comprising the initial population. During each generational iteration, genetic operators are applied to produce an offspring pool $\mathcal{P}'$ of equivalent size N, followed by the implementation of repair mechanisms when required to maintain solution feasibility (lines 4-10). The subsequent generation is determined through a selection procedure that operates on the unified population $\mathcal{P} \cup \mathcal{P}'$, utilizing non-dominated sorting in conjunction with crowding distance metrics (lines 12). A notable enhancement to the traditional NSGA-II framework is introduced through the integration of tabu search: when the Pareto front exhibits stagnation for *nonImp* consecutive generations, the algorithm activates a tabu search procedure to intensify the exploration of solutions within the current Pareto set. These solutions are

Table 1: Summary of the related works on the parallel truck- and-drone scheduling problem

Literature	#T	#D	Truck capacity	Drone capacity	Multi trip truck	Multi visit drone	Limited waiting time	Objective	Algorithm
Murray and Chu [5]	1	m	no	no	no	no	no	makespan	MILP, heuristic
Saleu et al. [9]	1	m	no	no	no	no	no	makespan	MILP, heuristic
Ulmer & Thomas[15]	n	m	no	no	yes	no	no	#customers	heuristic
Ham [16]	n	m	no	no	no	yes	no	makespan	constraint programming
Dell'Amico et al.[10]	1	m	no	no	no	no	no	makespan	MILP, matheuristic
Dinh et al. [11]	1	m	no	no	no	no	no	makespan	hybrid ant colony
Salue et al.[17]	n	m	no	no	no	no	no	makespan	MILP, meta-heuristic
Nguyen et al. [14]	n	m	no	yes	no	no	no	costs	MILP, meta-heuristic
Manh et al. [4]	n	m	no	no	no	yes	yes	makespan	MILP, meta-heuristic
Nguyen et al. [12]	1	m	no	no	no	no	no	makespan	MILP, branch and cut
Nguyen et al. [18]	1	m	no	no	no	no	no	makespan	meta-heuristic
Montemanni et al. [19]	n	m	yes	yes	no	no	no	costs	constraint programming
Montemanni et al [13]	n	m	no	yes	no	no	no	makespan	constraint programming
Our proposal	n	m	yes	yes	no	yes	no	makespan, waiting time	NSGA-II+tabu search

subsequently integrated into the current population $\mathcal{P}$ (lines 14-17). The selection process is then reapplied to maintain the prescribed population size N (line 18). This algorithmic cycle persists until predetermined termination criteria are satisfied.

Algorithm 1 Hybrid NSGAII-TS metaheuristic

1: Generate an initial population $\mathcal{P}$ of N individuals
2: **repeat**
3: New population $\mathcal{P}' \leftarrow \emptyset$
4: **repeat**
5: Select two parents from $\mathcal{P}$
6: A crossover operator is selected randomly and applied to procedure offspring O with probability p_c
7: Apply mutation to O with probability p_m
8: Repair O if it is infeasible
9: Add O to $\mathcal{P}'$
10: **until** the number of individuals in $\mathcal{P}'$ equals N
11: Update Pareto set of the problem based on $\mathcal{P} \cup \mathcal{P}$'
12: $\mathcal{P} = Select(\mathcal{P} \cup \mathcal{P}', N)$ ▷ Generation replacement
13: **if** Pareto set is not improved for *nonImp* generations **then**
14: **for** each individual x in the Pareto set **do**
15: $T(x) \leftarrow$ Tabu search on x
16: $\mathcal{P} \leftarrow \mathcal{P} \cup T(x)$
17: **end for**
18: $\mathcal{P} = Select(\mathcal{P}, N)$
19: **end if**
20: **until** termination criteria are satisfied

4.2 Individual representation

An individual for HNSGAII-TS corresponds to a feasible solution to the MSSVTDE specifying the number of trips and the collect test kits order within each trip. The relevant characteristics of each individual are encoded into a chromosome which is a sequence of trips servicing patients, starting with a trip for each technician and then a single trip or a sequence of trips for each drone. A number larger than the number of patients is added at the end of each trip for the decoding. This number is different for each trip.

4.3 Initial population

The algorithm initializes with a population of N feasible chromosomes and implements a simultaneous routing strategy for both technicians and drones through an assignment procedure. Beginning with random selection of initial patients for each technician and drone's first trip, the procedure employs an iterative nearest-ranking-neighbor approach based on temporal proximity. Given S as the set of all unserved patients and a ranking function $R(i, j, S)$ that returns the ranking of point j relative to i based on distance, considering all points in S. The construction process proceeds by identifying the vehicle (technician or drone) with the minimum current travel time for route expansion at each iteration. For a trip currently terminating at patient i, the subsequent patient j is selected by minimizing $R(i, j, S) + R(j, i, S \cup \{i\})$ among unserved patients while maintaining all operational constraints; when drone trips violate energy or load constraints during insertion, the drone returns to the medical center and initiates a new trip following identical selection criteria. This sequential assignment process continues iteratively until all patients have been successfully assigned to either technician or drone trips, thereby completing the routing solution.

4.4 Non-dominated sorting

For each chromosome in the population, the crowding distance metric is computed to quantify the solution's proximity to its nearest neighbors in the objective space, with this value being appended to the chromosome representation. The population is then partitioned into fronts based on the principle of Pareto dominance, where the first front consists of solutions that dominate all others in the population, while the second front contains solutions dominated exclusively by chromosomes of the first front. This hierarchical sorting continues until all chromosomes are assigned to their respective fronts. Within each front, solutions are further ordered based on their crowding distance values, thereby promoting diversity in the Pareto-optimal set. This dual-criterion sorting mechanism—combining dominance rank with crowding distance—ensures both convergence toward the Pareto-optimal front and maintenance of solution diversity, facilitating a well-distributed approximation of the entire Pareto frontier.

4.5 Parent selection

We employ a hybrid approach combining tournament selection with truncation selection to select two parents. The selection process begins by implementing a truncation-based filtering, where the top 50% of solutions from the current population are identified and grouped. This filtering ensures that only the most promising individuals are considered as potential parents. From this elite group, parent selection proceeds using a tournament-based approach governed by a normalized objective function $f(x)$, defined as:

$$f(x) = \frac{x.obj1 - bestobj1}{worstobj1 - bestobj1} + \frac{x.obj2 - bestobj2}{worstobj2 - bestobj2}$$

where $x.obj1$ and $x.obj2$ represent the values of the first and second objectives for individual x, respectively; $bestobj1$ ($bestobj2$) and $worstobj1$ ($worstbj2$) denote the best and worst values of the first (second) objective found in the current population. The individual exhibiting the minimum $f(x)$ value within the selected group is designated as the first parent. To obtain the second parent, this selection procedure is repeated, with the additional constraint that the selected individual must differ from the first parent, thereby ensuring genetic diversity in the offspring generation process.

4.6 Offspring generation

Crossover and mutation operators. The algorithm implements a probabilistic multi-operator crossover strategy with a crossover rate of p_c. For each offspring generation, one of four classical operators—Partially Mapped Crossover (PMX), Order Crossover (OX), Cycle Crossover (CX), or Position-based Crossover (POS) is randomly selected and applied to the parent chromosomes to produce an offspring. The mutation operator is applied to each offspring yielded by the crossover operators. Mutation consists in swapping positions of two patients with a probability p_m.

Chromosome repair mechanism. The application of genetic operators may produce infeasible offspring individuals, necessitating a repair mechanism to restore feasibility. Specifically, for infeasible drone trips, patients causing constraint violations are systematically transferred to the end of subsequent trips, propagating through the sequence of drone trips. This forward-pushing mechanism continues until reaching the terminal drone trip. Any remaining patients that still render the individual infeasible are then optimally inserted into technician trips, with positions selected to minimize impact on trip efficiency while maintaining feasibility constraints. This repair methodology ensures that all offspring individuals conform to the problem constraints while preserving the beneficial genetic information inherited from parent chromosomes.

4.7 Generation replacement

The generation replacement phase implements an elitist strategy designed to maintain population diversity while preserving best characteristics of the individuals encountered so far. The process begins by merging N individuals from the current population with N newly created offspring into a combined pool of $2N$ solutions. From this pool, N distinct solutions are selected to constitute the subsequent generation's population. The selection process first sorted individuals in this pool into non-dominated fronts. Within each front, solutions exhibiting identical objective values undergo a uniqueness filtering process, whereby only individuals with distinct serviced patients sequences are retained. Subsequently, the remaining solutions within each front are arranged according to their crowding distance values in descending order, promoting genetic diversity in the population. The new population is constructed through an iterative process, transferring individuals from the sorted fronts sequentially. Beginning with the first front, solutions are transferred one at a time until either all individuals in the current front have been moved or the new population reaches its target size N. If the population remains incomplete after exhausting a front, the process continues with the subsequent front until exactly N individuals have been selected for the new generation.

4.8 Education

Upon detecting Pareto front stagnation persisting for *nonImp* consecutive generations, the algorithm activates a tabu search to "educate" all individuals in the current population, and thus to enhance the current population $\mathcal{P}$. This mechanism serves to augment both the solution diversity inherent to NSGA-II and the distributional uniformity of nondominated solutions. The tabu search methodology, detailed in Algorithm 2, executes for a maximum of IT_{TS} iterations for each solution x selected from the current population. Each iteration begins with the random selection of a neighborhood type (line 4). Let $N(x)$ denote the set of feasible neighboring solutions accessible from x through the application of the selected neighborhood. Non-dominated sorting is subsequently applied to $N(x)$ to identify the first Pareto front F (line 5). The selection of the solution x' for the subsequent tabu search iteration follows a structured process (lines 7-14): a candidate solution x' is randomly selected from the first front F and evaluated against two criteria: it must either dominate x or be absent from the tabu list to be accepted. If these conditions are not met, the process iteratively selects and evaluates alternative solutions from F until either a suitable x' is identified or F is exhausted. Throughout the tabu search process applied to a specific solution x, all solutions from the first front F at each iteration are aggregated into a set $T(x)$ (line 6). The algorithm then proceeds to integrate the accumulated solutions from $T(x)$ for all solutions x in the current population $\mathcal{P}$ into $\mathcal{P}$ itself. Subsequently, the selection process described in section 4.7 is reapplied to maintain the population size of $\mathcal{P}$ at N individuals.

In the following, we present four neighborhood structures and their corresponding tabu list implementations used in our proposed tabu search:

Algorithm 2 Tabu search metaheuristic

$x \leftarrow$ a solution of current population $\mathcal{P}$
$T(x) \leftarrow \emptyset$
for $IT \leftarrow 0$ to IT_{TS} **do**
  Select a neighborhood randomly
  $F \leftarrow$ solutions in the first front of $N(x)$
  $T(x) \leftarrow T(x) \cup F$
  repeat
    Select a solution x' from F randomly
    if x' dominates x or x' is not tabu **then**
      $x \leftarrow x'$
      break
    end if
    Remove x' from F
  until F is empty
  if F is empty **then**
    break
  end if
end for
return $T(x)$

- The neighborhood types comprise: (1,0) move, involving the relocation of patient x after patient y; (1,1) move, executing a position swap between patients x and y; (2,0) move, relocating adjacent patients x and x' after patient y; and (2,1) move, exchanging adjacent patients x and x' with patient y.
- Each move type maintains its dedicated tabu list with the following restrictions: for (1,0) moves, the newly assigned position of x is protected; for (1,1) moves, the swapped positions of x and y are preserved; for (2,0) moves, the new positions of adjacent patients x, x' following y are secured; and for (2,1) moves, the swapped positions of x, x' and y are maintained.

5 Computational results

The algorithm was implemented in C++. Experiments were performed on GitHub Actions using GitHub runners with workflow Ubuntu-latest. The virtual machines setting are: 4 CPU, 16GB RAM, 14GB SSD, architecture x86_64, OS Version: 24.04.1 LTS, Kernel Version: 6.8.0-1017-azure, Image Version: 20250105.1.0 and Systemd version: 255.4-1ubuntu8.4. The experimental dataset, sourced from Sacramento et al. [25], comprises 60 instances with number of patients ranging from 20 to 200, consistently positioned around a medical center at coordinates [0,0]. Patient distribution follows a uniform distribution $U(-d/2, d/2)$ across grid sizes spanning 5×5 to 40×40 miles. Experimental parameters include a technician base speed of $V = 35$ mph with variability range $[F_L, F_U] = [0.4, 0.9]$, test kit weights randomly assigned between 10-100 grams, and drone configurations adapted from Murray and Raj Chu [26], featuring 563.0 KJ energy, variable speed profiles (takeoff: 17.5 mph, cruising: 35 mph, landing: 8.75 mph), with $\beta = 210.8$ (w/kg), $\gamma = 181.2$ w, and a 50-mile cruise altitude.

5.1 Analysis of design decisions

A hybrid meta-heuristic like HNSGAII-TS involves critical design decisions regarding algorithm structure, components, and parameters. Due to page limitations, this study focuses on three primary design components: crossover operators, population size and number of tabu search iterations. Other parameters were fixed as follows: crossover rate $p_c = 0.9$, mutation rate $p_m = 0.05$, and tabu search activation set at 30 unimproved generations.

5.1.1 Variants of crossover operators. The initial experiment assessed crossover operators' impact on solution quality by investigating several combinations of four crossover operator types (PMX, CX, OX, POS) using NSGA-II for 1500 generations with a population size of 200. All crossover combinations were evaluated against the reference combination PMX+POS+OX, with performance metrics displayed in Table 2 including computation time (seconds), hypervolume value (HV), relative time difference (GapT, positive values indicating longer runtime), hypervolume difference (GapHV, positive values indicating superior solutions), and distribution of performance outcomes (+/=/-, representing instances where combinations performed better, equally, or worse than the reference). The results showed comparable computation times across most combinations, with PMX+POS+CX being a notable exception in demonstrating significantly reduced computational time at the cost of markedly inferior solution quality. The reference combination PMX+POS+OX achieved the best performance with a mean hypervolume value of 0.82 and demonstrated superior results across the majority of test instances, leading to its selection for subsequent experimental phases based on its consistent performance advantages across the evaluated dataset.

5.1.2 Calibration of population size. Table 3 examines population size impact from 200 to 350 using HNSGAII-TS across 1500 generations with 10 tabu search iterations, using 200 as the reference population size. For instances with 20 and 50 patients, population size 200 outperformed larger sizes across three criteria: lower average time, better average hypervolume, and more instances with improved hypervolume. Conversely, for instances with larger patient numbers, higher population sizes enhanced solution quality, though each 50-generation increase required approximately 30% more computational time. Consequently, the population size of 200 was selected, providing optimal results for small customer sets while maintaining relatively good performance for larger instances with less time consumed.

5.1.3 Calibration of tabu search iterations. This experiment was conducted by varying the number of tabu search iterations from 10 to 50 per execution, with computational time constraints standardized across configurations based on problem size for the fairness: 20 seconds for 20-customer instances, 200 seconds for 50-customer instances, 1000 seconds for 100-customer instances, and 2000 seconds for 200-customer instances. The experimental results, presented in Table 4, utilized the 10-iteration configuration as a baseline, which achieved an average hypervolume of 0.79 and ranked second highest among all tested variants, demonstrating superior performance compared to configurations with 30 or more iterations in the majority of the 60 test instances. However, the 20-iteration configuration

Table 2: Performance comparison between crossover combinations

Dataset		Crossover Combinations										
		PMX+POS+OX		POS+OX+CX			PMX+POS+CX			PMX+OX+CX		
Patients	Grid size	Time (s)	HV	GapT (%)	GapHV (%)	+/=/-	GapT (%)	GapHV (%)	+/=/-	GapT (%)	GapHV (%)	+/=/-
20	5	1.98	0.69	2.81	-0.94	1/1/2	-20.22	-1.16	0/1/3	-15.41	-0.34	1/1/2
	10	1.55	0.79	10.90	0.76	1/0/3	-18.25	1.96	1/0/3	11.66	-1.10	1/0/3
	20	2.03	0.75	21.95	1.20	3/0/1	-16.92	-2.33	2/0/2	1.98	1.25	4/0/0
	Average	**1.85**	**0.74**	**11.88**	**0.34**	**5/1/6**	**-18.46**	**-0.51**	**3/1/8**	**-0.59**	**-0.06**	**6/1/5**
50	10	10.70	0.86	-1.88	-10.02	0/0/4	-39.07	-26.37	0/0/4	17.46	-3.61	0/0/4
	20	12.90	0.86	-0.46	-5.84	0/0/4	-48.57	-9.98	0/0/4	5.69	-0.16	2/0/2
	30	10.73	0.76	8.96	4.08	2/0/2	-60.74	-1.23	1/0/3	6.56	-0.20	1/0/3
	40	17.93	0.84	-17.47	-1.95	1/0/3	-76.28	-9.89	0/0/4	-6.60	-4.89	1/0/3
	Average	**13.06**	**0.83**	**-2.71**	**-3.43**	**3/0/13**	**-56.16**	**-11.87**	**1/0/15**	**5.78**	**-2.21**	**4/0/12**
100	10	52.65	0.88	-12.63	-9.78	0/0/4	-54.63	-24.89	1/0/3	-1.29	-3.89	1/0/3
	20	54.10	0.88	-10.97	-4.27	0/0/4	-60.45	-7.42	1/0/3	-4.11	-6.00	0/0/4
	30	102.75	0.85	-17.16	-4.94	1/0/3	-74.72	-22.02	0/0/4	-4.55	-8.03	1/0/3
	40	118.23	0.72	11.79	0.26	2/0/2	-79.79	-19.81	1/0/3	11.96	-0.59	2/0/2
	Average	**81.93**	**0.83**	**-7.24**	**-4.68**	**3/0/13**	**-67.40**	**-18.53**	**3/0/13**	**0.50**	**-4.63**	**4/0/12**
200	10	260.43	0.92	-3.23	-9.60	0/0/4	-65.54	-30.56	0/0/4	11.62	-4.52	1/0/3
	20	328.55	0.88	3.70	1.39	2/0/2	-66.73	-20.94	0/0/4	1.08	5.48	3/0/1
	30	332.28	0.85	5.66	5.22	3/0/1	-69.07	-10.69	0/0/4	-4.49	6.88	3/0/1
	40	705.43	0.80	-3.60	0.70	2/0/2	-80.58	-4.77	2/0/2	6.77	2.91	3/0/1
	Average	**406.67**	**0.86**	**0.63**	**-0.57**	**7/0/9**	**-70.48**	**-16.74**	**2/0/14**	**3.75**	**2.69**	**10/0/6**
Summary		**134.15**	**0.82**	**-0.11**	**-2.25**	**18/1/41**	**-55.44**	**-12.67**	**9/1/50**	**2.56**	**-1.12**	**24/1/35**

Table 3: Performance comparison between population sizes

Dataset		Population Size										
		200		250			300			350		
Patients	Grid size	Time (s)	HV	GapT (%)	GapHV (%)	+/=/-	GapT (%)	GapHV (%)	+/=/-	GapT (%)	GapHV (%)	+/=/-
20	5	5.70	0.61	10.02	-1.26	1/1/2	30.25	-2.24	1/1/2	61.84	0.24	2/1/1
	10	3.35	0.54	37.08	0.29	1/1/2	88.66	0.50	2/1/1	86.45	0.34	2/1/1
	20	3.35	0.81	27.14	0.07	2/0/2	47.99	-0.59	0/0/4	70.21	-0.74	0/0/4
	Average	**4.13**	**0.65**	**24.75**	**-0.30**	**4/2/6**	**55.63**	**-0.77**	**3/2/7**	**72.83**	**-0.06**	**4/2/6**
50	10	29.95	0.91	28.83	-1.59	1/0/3	39.07	0.38	2/0/2	41.85	-2.48	1/0/3
	20	62.98	0.87	21.64	-1.19	1/0/3	26.97	-0.59	1/0/3	44.85	-0.25	2/0/23
	30	62.60	0.80	1.31	-1.97	1/0/3	-0.71	-1.44	1/0/3	1.18	-2.72	0/0/4
	40	84.85	0.78	18.10	0.07	2/0/2	39.18	0.07	3/0/1	58.02	0.17	2/0/2
	Average	**60.09**	**0.84**	**17.47**	**-1.17**	**5/0/11**	**26.13**	**-0.39**	**7/0/9**	**36.48**	**-1.32**	**5/0/11**
100	10	94.35	0.78	62.52	2.45	4/0/0	123.65	4.90	4/0/0	186.42	12.33	4/0/0
	20	104.25	0.84	65.06	2.88	3/0/1	118.14	3.67	3/0/1	153.96	5.56	4/0/0
	30	184.20	0.75	51.77	7.89	3/0/1	106.64	12.42	4/0/0	152.86	11.71	4/0/0
	40	302.93	0.69	50.62	3.23	3/0/1	72.53	6.46	3/0/1	122.85	9.35	4/0/0
	Average	**171.43**	**0.77**	**57.49**	**4.11**	**13/0/3**	**105.24**	**6.86**	**14/0/2**	**154.02**	**9.74**	**16/0/0**
200	10	275.13	0.82	24.25	0.95	2/0/2	56.18	3.10	3/0/1	101.18	4.15	2/0/20
	20	387.35	0.92	13.00	-8.49	0/0/4	51.30	-6.41	0/0/4	97.12	-5.74	2/0/2
	30	458.23	0.82	36.25	-0.48	3/0/1	68.16	4.75	4/0/0	106.12	2.95	3/0/1
	40	1042.90	0.63	1.85	14.99	4/0/0	24.54	13.11	4/0/0	56.95	25.15	4/0/0
	Average	**540.90**	**0.80**	**18.84**	**1.74**	**9/0/7**	**50.04**	**3.64**	**11/0/5**	**90.34**	**6.63**	**11/0/5**
Summary		**206.81**	**0.77**	**29.96**	**1.19**	**31/2/27**	**59.50**	**2.54**	**35/2/23**	**89.46**	**4.00**	**36/2/22**

emerged as the optimal choice, achieving the highest average hypervolume and outperforming the reference configuration in 30 out of 60 instances, leading to its selection for subsequent HNSGAII-TS experiments.

5.2 Performance of HNSGAII-TS

The performance of HNSGAII-TS was evaluated through comparisons with established approaches including NSGA-II, Tabu Search (TS), and MOEA/D (Multi-Objective Evolutionary Algorithm based on Decomposition). All algorithms were executed under standardized computational time limits based on problem size to ensure fair comparison as indicated in section 5.1.3. MOEA/D, introduced by Zhang and Li [27] which decomposes multi-objective optimization problems into collaborative scalar optimization subproblems, was configured with population sizes matching HNSGAII-TS and NSGA-II to ensure equitable resource allocation. The experimental results demonstrated the proposed HNSGAII-TS's superior performance across multiple metrics. For hypervolume comparison, Table 5 shows that HNSGAII-TS achieved an average value of 0.87 for 50-patient instances, outperforming NSGA-II, TS, and MOEA/D by

Table 4: Performance comparison between number of tabu search iterations

Dataset		Number of tabu search iterations								
		10	20		30		40		50	
Patients	Grid size	HV	GapHV (%)	+/=/-	GapHV (%)	+/=/-	GapHV (%)	+/=/-	GapHV (%)	+/=/-
20	5	0.67	0.36	2/1/1	-1.28	1/1/2	-1.02	0/1/3	-0.38	1/1/2
	10	0.48	38.46	1/1/2	38.72	2/1/1	-0.39	0/2/2	-24.99	1/1/2
	20	0.84	0.48	3/0/1	0.01	3/0/1	0.07	4/0/0	0.59	3/1/0
	Average	**0.66**	**13.10**	**6/2/4**	**12.48**	**6/2/4**	**-0.45**	**4/3/5**	**-8.26**	**5/3/4**
50	10	0.84	6.04	4/0/0	5.37	4/0/0	5.24	3/0/1	1.67	3/0/1
	20	0.83	-1.24	1/0/3	-1.88	1/0/3	-1.43	1/0/3	-2.73	0/0/4
	30	0.84	-0.26	1/0/3	-0.35	1/0/3	0.24	1/0/3	0.00	2/0/2
	40	0.78	-0.32	2/0/2	-0.14	1/0/3	-0.40	1/0/3	-0.09	1/0/3
	Average	**0.82**	**1.05**	**8/0/8**	**0.75**	**7/0/9**	**0.91**	**6/0/10**	**-0.29**	**6/0/10**
100	10	0.81	7.96	3/0/1	-1.15	1/0/3	1.26	3/0/1	-8.54	0/0/4
	20	0.86	-2.94	2/0/2	-2.38	2/0/2	-5.40	0/0/4	-3.46	1/0/3
	30	0.82	-0.97	2/0/2	-2.07	2/0/2	-5.04	1/0/3	-4.61	0/0/4
	40	0.73	1.40	2/0/2	-0.32	1/0/3	-0.83	2/0/2	2.07	2/0/2
	Average	**0.81**	**1.36**	**9/0/7**	**-1.48**	**6/0/10**	**-2.50**	**6/0/10**	**-3.64**	**3/0/13**
200	10	0.85	6.20	3/0/1	-8.07	1/0/3	-6.83	1/0/3	-14.60	1/0/3
	20	0.92	-8.88	0/0/4	-15.56	0/0/4	-14.46	0/0/4	-23.15	0/0/4
	30	0.88	3.23	1/0/3	-16.40	0/0/4	-8.62	0/0/4	-19.67	0/0/4
	40	0.76	5.88	3/0/1	-3.11	1/0/3	-1.07	3/0/1	-0.65	2/0/2
	Average	**0.85**	**1.61**	**7/0/9**	**-10.78**	**2/0/14**	**-7.75**	**4/0/12**	**-14.52**	**3/0/13**
Summary		**0.79**	**3.69**	**30/2/28**	**-0.57**	**21/2/37**	**-2.58**	**20/3/37**	**-6.57**	**17/3/40**

Table 5: Performance comparison among algorithms

Dataset		HNSGAII-TS	NSGA-II		TS		MOEA/D	
Patients	Grid size	HV	GapHV (%)	+/=/-	GapHV (%)	+/=/-	GapHV (%)	+/=/-
20	5	0.73	-4.45	0/1/3	-4.68	1/1/2	9.03	3/1/0
	10	0.89	-3.70	0/0/4	-2.72	1/0/3	-7.47	3/0/1
	20	0.85	-1.69	2/0/2	-18.38	0/0/4	1.66	2/0/2
	Average	**0.82**	**-3.28**	**2/1/9**	**-8.59**	**2/1/9**	**1.07**	**8/1/3**
50	10	0.93	-1.23	1/0/3	-39.58	0/0/4	-41.72	2/0/2
	20	0.83	-2.86	0/0/4	-46.35	0/0/4	-10.21	1/0/3
	30	0.82	-7.84	0/0/4	-40.28	0/0/4	-51.92	0/0/4
	40	0.88	-16.29	0/0/4	-12.44	0/0/4	-67.58	0/0/4
	Average	**0.87**	**-7.05**	**1/0/15**	**-34.66**	**0/0/16**	**-42.86**	**3/0/13**
100	10	0.98	-9.61	0/0/4	-51.93	0/0/4	-71.99	0/0/4
	20	0.94	-18.21	0/0/4	-49.42	0/0/4	-26.92	0/0/4
	30	0.92	-24.82	0/0/4	-34.83	0/0/4	-60.82	0/0/4
	40	0.85	-39.15	0/0/4	-26.92	0/0/4	-90.15	0/0/4
	Average	**0.92**	**-22.95**	**0/0/16**	**-40.78**	**0/0/16**	**-62.47**	**0/0/16**
200	10	0.97	-11.40	0/0/4	-19.79	0/0/4	-11.04	0/0/4
	20	0.99	-8.14	0/0/4	-22.68	0/0/4	-8.34	0/0/4
	30	0.99	-19.95	0/0/4	-40.26	0/0/4	-41.88	0/0/4
	40	0.96	-32.20	0/0/4	-14.75	0/0/4	-94.55	0/0/4
	Average	**0.98**	**-17.92**	**0/0/16**	**-24.37**	**0/0/16**	**-38.95**	**0/0/16**
Summary		**0.90**	**-13.43**	**3/1/56**	**-28.33**	**2/1/57**	**-38.26**	**11/1/48**

7.05%, 34.66%, and 42.86% respectively. This dominance extended to larger instances (100 and 200 patients), although for 20-patient instances, MOEA/D showed better performance while HNSGAII-TS still surpassed NSGA-II and TS. In terms of solution quality and quantity, Table 6 indicates that HNSGAII-TS generated 12.70, 29.90, 54.35, and 41.44 solutions on average for instances with 20, 50, 100, and 200 patients respectively, with a high proportion remaining non-dominated when compared against other algorithms. While MOEA/D produced more solutions across all instance sizes (21.27, 50.16, 132.94, and 214.81), most were dominated by HNSGAII-TS solutions, as evidenced by their low non-dominated solution counts (15.38, 18.73, 5.19, and 16.29). Similar patterns emerged with NSGA-II (5.32, 10.25, 7.83, 4.58 non-dominated solutions) and TS (2.32, 0.39, 0.36, 2.31 non-dominated solutions). Visualization of results for a 100-patient instance 100.10.1 from Figures 2 further confirmed HNSGAII-TS's effectiveness, showing superior Pareto front coverage and consistently lower values in both objectives - completion time and total waiting time - compared to other algorithms' more scattered and less optimal solution distributions.

Table 6: Summary (average) of non-dominated solutions for each algorithm

	Size of instances			
Average	20	50	100	200
HNSGAII-TS solutions	12.70	29.90	54.35	41.44
HNSGAII-TS solutions not dominated by NSGA-II	8.68	24.23	50.11	37.93
HNSGAII-TS solutions not dominated by TS	10.70	29.53	54.16	40.18
HNSGAII-TS solutions not dominated by MOEA/D	5.87	27.94	53.99	40.78
NSGA-II solutions	11.80	23.78	32.28	27.65
NSGA-II solutions not dominated by HNSGAII-TS	5.32	10.25	7.83	4.58
TS solutions	10.82	23.38	26.78	45.33
TS solutions not dominated by HNSGAII-TS	2.32	0.39	0.36	2.31
MOEA/D solutions	21.27	50.16	132.94	214.81
MOEA/D solutions not dominated by HNSGAII-TS	15.38	18.73	5.19	16.29

5.3 Impact of drone configuration

While our preceding experimental analyses exclusively employed low-speed drone configurations, we conducted additional investigations to evaluate the influence of drone velocity on solution quality.

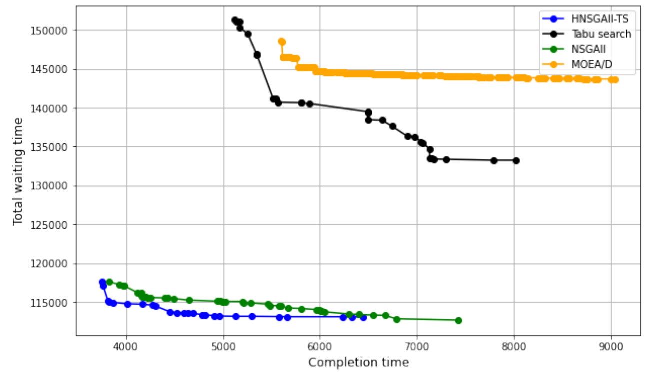

Figure 2: Performance comparison on instance 100.10.1.

Table 7: Specifications of drone

Speed type	Energy (KJ)	β (w/kg)	γ (w)	v_t (mph)	v_c (mph)	v_l (mph)
Low	563	210.8	181.2	17.5	35.0	8.75
High	904	24.2	1392.0	35.0	70.0	17.5

This supplementary study incorporated high-speed drone configurations to provide a comprehensive comparative analysis. The operational specifications of both drone variants, as documented by Murray and Raj Chu [26], are summarized in Table 7. Low-speed drones could achieve maximum flight distances of approximately 27 miles and 14 miles for payload masses of 100 grams and 1000 grams, respectively. In contrast, high-speed drones manifest substantially reduced ranges of 12.6 miles and 12.4 miles under identical payload conditions. Despite their reduced velocity and energy consumption, low-speed drones demonstrate enhanced coverage capabilities. Further analysis of average hypervolume values, presented in Table 8, reveals a nuanced performance distribution across varying operational contexts: high-speed drones exhibit better performance in scenarios with small grid sizes (5 or 10 units), while low-speed configurations achieve superior hypervolume metrics in cases of widely dispersed patient distribution, attributable to their enhanced coverage characteristics.

Table 8: Impact of drone configuration on solution quality

	Grid size				
Average Hypervolume of	5	10	20	30	40
Low speed	0.38	0.79	**0.86**	**0.90**	**0.91**
High speed	**0.94**	**0.89**	0.38	0.37	0.43

6 Conclusions

The MSSVTDE framework demonstrates a new variant in medical sampling service systems by integrating time-varying technician constraints, drone energy consumption modeling, and bi-objective optimization. Our hybrid algorithm, combining NSGA-II and tabu search, effectively addresses complex operational challenges. Experimental validation confirms better performance across multiple computational metrics, highlighting the approach's potential to enhance efficiency and service quality in medical sampling logistics.

References

[1] Macrina, G.; Di Puglia Pugliese, L.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies* **2020**, *120*, 102762.

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

[3] Zipline Receives the 2021 U.S. Secretary of State's Award for corporate excellence in health security. *SUAS News* **December 9, 2021**,

[4] Manh, P. P.; Hue, T. T.; Binh, H. T. T.; Phuong, N. K. The min-timespan parallel technician-and-drone scheduling in door-to-door sampling service system. 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA). pp 1–10.

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

[6] Vuong, Q. H.; Dang, G. T.-H.; Do Quang, T.; Pham, M.-T. Traveling Salesman Problem with Truck and Drones: A Case Study of Parcel Delivery in Hanoi. International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences. 2021; pp 75–86.

[7] Dorling, K.; Heinrichs, J.; Messier, G. G.; Magierowski, S. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2017**, *47*, 70–85.

[8] Macrina, G.; Pugliese, L. D. P.; Guerriero, F.; Laporte, G. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies* **2020**, *120*, 102762.

[9] Mbiadou Saleu, R. G.; Deroussi, L.; Feillet, D.; Grangeon, N.; Quilliot, A. An iterative two-step heuristic for the parallel drone scheduling traveling salesman problem. *Networks* **2018**, *72*, 459–474.

[10] Dell'Amico, M.; Montemanni, R.; Novellani, S. Matheuristic algorithms for the parallel drone scheduling traveling salesman problem. *Annals of Operations Research* **2020**, *289*, 211–226.

[11] Dinh, Q. T.; Do, D. D.; Hà, M. H. Ants can solve the parallel drone scheduling traveling salesman problem. Proceedings of the Genetic and Evolutionary Computation Conference. 2021; pp 14–21.

[12] Nguyen, M. A.; Luong, H. L.; Hà, M. H.; Ban, H.-B. An efficient branch-and-cut algorithm for the parallel drone scheduling traveling salesman problem. *4OR* **2023**, *21*, 609–637.

[13] Montemanni, R.; Dell'Amico, M.; Corsini, A. Parallel drone scheduling vehicle routing problems with collective drones. *Computers & Operations Research* **2024**, *163*, 106514.

[14] Nguyen, M. A.; Dang, G. T.-H.; Ha, M. H.; Pham, M.-T. The min-cost parallel drone scheduling vehicle routing problem. *European Journal of Operational Research* **2022**, *299*, 910–930.

[15] Ulmer, M.; Thomas, B. Same-day delivery with heterogeneous fleets of drones and vehicles. *Networks* **2018**, *72*, 475–505.

[16] Ham, A. M. 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* **2018**, *91*, 1–14.

[17] Saleu, R. G. M.; Deroussi, L.; Feillet, D.; Grangeon, N.; Quilliot, A. The parallel drone scheduling problem with multiple drones and vehicles. *European Journal of Operational Research* **2022**, *300*, 571–589.

[18] Nguyen, M. A.; Ha, M. H. The Parallel Drone Scheduling Traveling Salesman Problem with Collective Drones. *Transportation Science* **2023**,

[19] Montemanni, R.; Dell'Amico, M. Constraint programming models for the parallel drone scheduling vehicle routing problem. *EURO Journal on Computational Optimization* **2023**, *11*, 100078.

[20] Das, D. N.; Sewani, R.; Wang, J.; Tiwari, M. K. Synchronized truck and drone routing in package delivery logistics. *IEEE Transactions on Intelligent Transportation Systems* **2020**, *22*, 5772–5782.

[21] Zhang, S.; Liu, S.; Xu, W.; Wang, W. A novel multi-objective optimization model for the vehicle routing problem with drone delivery and dynamic flight endurance. *Computers & Industrial Engineering* **2022**, *173*, 108679.

[22] Lu, Y.; Yang, C.; Yang, J. A multi-objective humanitarian pickup and delivery vehicle routing problem with drones. *Annals of Operations Research* **2022**, *319*, 291–353.

[23] Kuo, R.; Edbert, E.; Zulvia, F. E.; Lu, S.-H. Applying NSGA-II to vehicle routing problem with drones considering makespan and carbon emission. *Expert Systems with Applications* **2023**, *221*, 119777.

[24] Lu, Y.; Yang, J.; Yang, C. A humanitarian vehicle routing problem synchronized with drones in time-varying weather conditions. *Computers & Industrial Engineering* **2023**, *184*, 109563.

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

[26] Murray, C. C.; Raj, R. The multiple flying sidekicks traveling salesman problem: Parcel delivery with multiple drones. *Transportation Research Part C: Emerging Technologies* **2020**, *110*, 368–398.

[27] Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation* **2007**, *11*, 712–731.