UV-CDS: An Energy-Efficient Scheduling of UAVs for Premises Sterilization

Tahrat Tazrin, Mostafa M. Fouda, *Senior Member, IEEE,* Zubair Md Fadlullah, *Senior Member, IEEE*, and Nidal Nasser, *Senior Member, IEEE.*

***Abstract*—Despite the severity of the second wave of the novel coronavirus disease (COVID-19) and the recent hope for vaccine roll-outs, many public and private institutions are forced to resume their activities subject to ensuring an adequate sterilization of their premises. The existing off-the-shelf drones for such environment sanitization have limited flight-time and payload-carrying capacity. In this paper, we address this challenge by formulating an optimization problem to minimize the energy consumed by drones equipped with ultraviolet-C band (UV-C) panels. To solve this computationally hard problem, we propose several heuristics, such as a randomized path selection algorithm whose solution is further improved with a genetic algorithm-based UV-C drone-based sterilization (UV-CDS) scheduling technique. We consider educational institutions, confronting increasing infections, as an important use-case for the problem. Due to the energy constraint of the drones, the number of drones required for sterilization of the campus is smartly altered for various campus scenarios. The respective energy-efficient paths in the proposed heuristics and our envisioned UV-CDS are estimated for the drones. The performance is evaluated through extensive computer-based simulations which clearly demonstrates the effectiveness of UV-CDS in terms of sub-optimal performance and much faster execution time in contrast with the other methods.**

***Index Terms*—COVID-19, drones, genetic algorithm, sterilization, energy-efficiency, ultra-violet panel, optimization.**

I. Introduction

With the continued impact of novel coronavirus (COVID-19) all around the world, various technologies are emerging to strive against the virus [1], [2]. Among them, disinfectant spraying robots appeared as a popular choice in many countries to keep various premises sterilized. In particular, Unmanned Aerial Vehicles (UAVs) or drones carrying Ultraviolet-C (UV-C) rods have gained popularity as a more viable mode of sterilization due to their performance and efficiency. This can be successfully employed in various public and private facilities (e.g., educational institutions, offices, shopping malls, and so forth) if efficiently planned, as shown in Fig. 1. Due to the reopening of institutions in various parts of the world (such as the USA, Canada, and so forth) despite the successive pandemic waves, it is important to ensure that the institution premises are properly sterilized to limit further spread of the virus among students, instructors, and personnel. In such scenarios, the combination of drones and UV-C technology offers several advantages. First, instead of carrying much heavier disinfection agents and pumping/spraying mechanisms, a relatively lighter payload (i.e., UV-C panel) is conducive to the drone-based deployment. Second, UV-C has been widely used for disinfecting various domestic and industrial environments due to its decent sterilization accuracy [3] in contrast with spray-based sanitization techniques. Third, drones can hover over places having limited accessibility and permit UV-C emission for enough time to eradicate microorganisms. Hence, coupling the sterilization capability with the drone's agile coverage [4] is considered in this paper. This leads to the research problem of effectively scheduling the UV-C drones to carry out disinfection missions to prevent them from harming people in the vicinity. We address this problem in this paper by presenting the use-case of a school/university campus, which is regarded as a highly potential site for extensive COVID-19 spread due to high mobility of students and staff members at on-premises locations including classrooms, library, cafeteria, common room, and so forth.

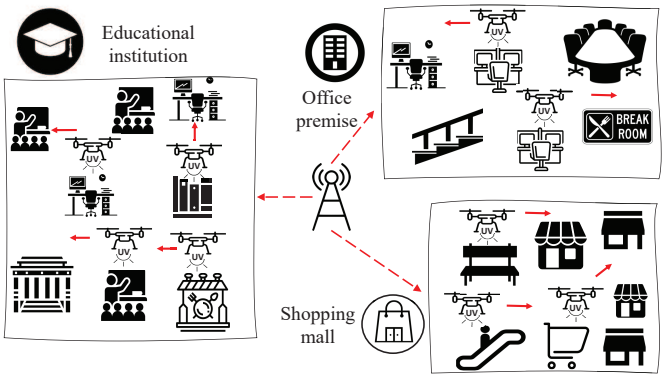


Fig. 1: Considered architecture of UV-C drone-based sterilization system whereby the energy-constrained drones need to be optimally scheduled for efficient sterilization. Various on-premise sterilization use-cases are demonstrated for ease of illustration.

Despite the recent demonstration of a UV-C emitting drone for a small room-sterilization by Digital Aerolus, Inc. [5], a large-scale deployment of such aerial disinfecting devices is constrained by the limited battery-life resulting in a rather short flight-time of the drone. As a consequence, the deployment of such devices with broader scenarios such as school/-

T. Tazrin is with the Department of Computer Science, Lakehead University, Thunder Bay, Canada (e-mail: ttazrin@lakeheadu.ca).
M. M. Fouda is with the Department of Electrical and Computer Engineering, College of Science and Engineering, Idaho State University, Pocatello, ID 83209, USA (e-mail: mfouda@ieee.org).
Z. M. Fadlullah is with the Department of Computer Science, Lakehead University, and Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada (e-mails: zubair.fadlullah@lakeheadu.ca, fadlullz@tbh.net).
Nidal Nasser is with the College of Engineering, Alfaisal University, KSA. Email: nnasser@alfaisal.edu

college campuses, shopping malls, and so forth, is not scalable. For a finite number of drones to cover a target area (e.g., a campus), various aspects need to be taken into consideration to ensure proper scalability of the system. One of the most crucial factor includes an efficient utilization of the available resources which can be achieved by ensuring minimal energy usage by the UV-C drones during the disinfection procedure. In this vein, we construct a sub-optimal heuristic-based UV-C drone scheduling model called UV-CDS (UV-C drone-based sterilization) that can be utilized to minimize the drone's energy consumption while disinfecting every location of a certain premise for a fixed number of drones. We formulate a problem to minimize the energy consumed by the drones, considering practical battery-resource and sterilization service constraints. We use a graph-based model, where the vertices of the graph represent the premises to be sterilized and the edges represent the trajectories of the UV-C drones. However, this problem is computationally hard which makes it more complex to be solved for a higher number of premises with a specified number of drones. Therefore, we have employed various heuristics, namely, random and greedy-based heuristics and genetic algorithm [6], to construct a sub-optimal solution which can be regarded as a trade-off between speed and efficiency. We have made some realistic assumptions regarding various features of the drone such as payload capacity and battery power, to evaluate the performance of the models. To the best of our knowledge, constructing an optimized path for the UV-C drones is yet to be explored in the literature for sterilization surfaces aiming to prevent further transmission of COVID-19.

The remainder of the paper is organized as follows. Section II surveys the relevant research work regarding the existing UV-C-based approaches for disinfection purposes. The related work considering coupling UV-C with drones as well as efficient path planning approaches of drones are also surveyed in this section. Our considered system model is presented in section III. The problem of traditional disinfection approaches and the need for an optimal scheduling of UV-C drones is described in section IV. Our proposed solution to the problem is presented in section V. The time and space complexity of the proposed solution are analyzed in section VI. The performance of our proposal is evaluated in various campus scenarios in section VII. Finally, section VIII concludes the paper.

II. Related Work

In this section, we first survey the relevant research work using Ultraviolet (UV) rays for sterilization and then describe how drones have been used in the literature to facilitate sterilization. UV waves are recently used for eradicating COVID-19 in various scenarios. For instance, Hamzavi et al. [7] mentioned the use of UV-C light for respirator disinfection so that respirators, which were scarce during the pandemic, could be safely reused. They proposed reasonable doses of UV germicidal irradiation that can be used to destroy the novel coronavirus in N95 masks and make them reusable by care-givers and frontline workers. The International Advanced Research Centre for Powder Metallurgy and New Materials constructed a UV-C disinfection trolley [8], which was used in several hospitals. The UV-C disinfection trolley consists of 6 UV-C germicidal tubes and is expected to sterilize the entire surrounding environment that includes walls, beds, floor, and room air. A UV-C disinfection robot was also introduced in [9], which introduced UV-C disinfection robots in hospitals, airlines, and public transports for the purpose of sanitization. The robot can sense motion and measure temperature to detect surrounding humans, and operate safely. Additionally, in [10], a scanner chamber with far-UVC was introduced, that employs a lower range of wavelengths between 207 and 222 nm, to disinfect people from pathogens including the novel coronavirus (COVID-19) while minimizing the adverse effect on the human skin. Apart from combating COVID-19, UV-C is used for various other disinfection purposes. The Gallium Nitride-based UV-C light-emitting diode (LED) were proven to be an effective way of treating contaminated water in [11] after several suitability tests has been performed under various conditions. In [12], Curtis et al. designed a Crystalline ultraviolet C (C-UVC) filter unit, which was observed to decrease the particles in the air in operative settings and sterilize the surroundings. C-UVC filters were also demonstrated to recirculate sterilized air in the operating room.

Additionally, UV light-carrying drones have been employed in various scenarios that transformed the existing paradigm of UAV-based missions for performing reconnaissance and constructing flexible communication networks [13]–[16]. For instance [17], the UltraViolet Direction And Ranging (UVDAR) sensor was utilized in multi-UAV flights to administer the discipline between leader and follower UAVs by sensing their relative orientation information. A UAV carrying a UV-LED light source was employed in the "Telescope Array" experiment to calibrate the fluorescence detectors. Since UV light-carrying drones are already being used for various purposes, they have been identified as a promising solution for disinfecting surfaces to mitigate COVID-19 spread. With this concept, the autonomous technology company called "Digital Aerolus" recently constructed a drone to disinfect large areas [5]; however without any consideration for efficient scheduling to minimize impact on humans in the vicinity. In addition, other disinfecting robots have been introduced in several countries to combat COVID-19 in public areas. For example, Xenex introduced a "LightStrike" disinfection robot to clean virus and bacteria using UV disinfection technology [18]. Also, Nanyang Technological University in Singapore introduced the eXtremeDisinfection robot (XDBOT) [19], which can be operated using mobile platforms. XDBOT was claimed to reach inaccessible areas of a room, e.g., under the bed and so forth.

The aforementioned UV-light/rod carrying drone-based disinfection techniques could be supported by a myriad of scheduling algorithms in the recent literature. In the remainder of the section, the scheduling algorithms are surveyed which ensure that the drones are exploited to efficiently carry out specific tasks to their maximum capacity. For instance, a dynamic trajectory control algorithm for multiple UAVs, to optimally plan their flight paths while improving the communication delay and throughput, was explored in [20]. In [21],

an energy-efficient data collection method called ECO-UDC was proposed that can be used in disaster-affected areas. The model tends to minimize energy consumption while also reducing the path length. A minimal trajectory planning technique was introduced in [22], where multiple objectives such as data collection from each sensor node and the time required for energy harvesting were divided into sub-problems to minimize the average age of information. Two heuristic algorithms were applied, i.e., dynamic programming and ant colony optimization (ACO), and the latter resulted in a sub-optimal performance. In [23], a genetic algorithm-based algorithm, using a customized fitness function accounting for the various environmental aspects, was designed to construct an intelligent camera control path planning. Furthermore, in the research work conducted in [24], a genetic algorithm was first developed to derive the states and path segments of a multi-UAV scenario. Then, these states and path segments were used as path planning features and labels to train a Convolutional Neural Network (CNN) model to provide fast, path planning inferences.

III. Proposed System Model: Drone-aided on-premises disinfection technique

In this section, we discuss our considered UV-C drone-based sterilization (UV-CDS) system model to sterilize institutional premises. To construct a feasible model for disinfecting an area, we have considered a set of locations L within a premise P. The aim of the drone is to disinfect as many locations l within an area as possible within its capacity where $l \in L$. Typically, drones are considered to be capacitated due to its energy constraints for longer flight paths. Therefore, to make them relevant to our particular problem, UV-C light was chosen for sterilizing purpose in the premises. Since drones containing disinfecting spray are a very popular choice for combating pests in agricultural fields, the concept could be similarly extended toward disinfecting locations within P. However, this simple approach is not without its shortcomings rendering it impractical in real-life scenarios. For example, the capacity to carry a large quantity of spray required for a particular l and the energy required for spraying those across the area are excessively high. Therefore, the UV-C light may be coupled with the UAV as a compact and practical mode of disinfection for our considered scenario. However, the process of travelling to a place and emit UV-C light still requires a drone to spend a significant amount of energy. Furthermore, it is unlikely for a single drone to cover all the areas in the premise P. Hence, multiple drones are required to achieve the stated goal of a sterilized environment. Our considered system model consists of the number of drones N_D assigned for P, the x and y coordinates of the target areas, and the size and availability of l. Accordingly, the best path for each drone needs to be determined by the UV-CDS model. For this purpose, the area of each location, a, and the distances from l_i to l_j are two important factors that should be taken into account where $\{l_i, l_j\} \in L$.

Now, we describe our detailed UV-CDS system model. Initially, the drones commence their flight from a base station, bs (such as a hangar), and follow the path assigned to them by the UV-CDS model. The drones will hover over possibly infected locations (l) and emit UV-C rays from a height of approximately three feet to sterilize an area. Since the size of the area is previously determined, the drone can accordingly adjust its hovering time and radiation amount to adequately sterilize that area. Once the area is sterilized, the drone moves to the next area. Throughout this procedure, it is critical to ensure that the drone's remaining energy is sufficient to travel to the next location l_n and thoroughly sterilize it. When the drone's energy level falls below a maximum tolerable energy threshold denoted by E_{max}, it needs to return to the base station bs. E_{max} is set as a ratio (ρ) of the total energy capacity of the drone. This is done to guarantee that the drone returns to bs before depleting all its energy. At bs, the drone is then sterilized and recharged so that it may visit the next route assigned to it. To calculate the total energy E_T consumed by the drone, both the energy required E_D to travel a distance D and the energy required to emit UV-C ray E_{UV} need to be taken into account as follows:

$$E_T = E_D + E_{UV}. \tag{1}$$

Based on the maximum energy consumption limit of each drone and the energy required to travel a path from l_i to l_j and disinfect a premise location l_j, each individual path is determined. Since the paths of the drones are pre-determined by the UV-CDS model, the communication energy, τ is considered constant throughout the whole sterilization task and is, therefore, omitted.

To calculate the energy required to travel a distance to reach the next area E_D, the kinetic energy (i.e., the energy of motion) of the drone can be calculated based on:

$$E_D = \frac{1}{2} m_d v_d^2, \tag{2}$$

where m_d and v_d denote the mass and velocity of the drone, respectively. The velocity of the drone depends on its displacement s as follows:

$$v_d^2 = u_d^2 + 2\alpha s, \tag{3}$$

where α and u_d indicate the acceleration and the initial velocity of the drone. The mass of the drone includes the components of the drone in addition to the UV-C panel attached to it. On the other hand, to calculate the energy consumed by the UV-C panel, the area of the location to be disinfected should be taken into consideration, i.e., $E_{UV} = e_{uv} a_l$ where e_{uv} is the energy required to emit UV-C per squared meter and a_l denotes the area of location l.

Having constructed the UV-CDS system model, we focus on formulating the problem in the following section.

IV. Problem Formulation

In this section, we present a formal problem formulation. First, we describe the preliminaries required for the problem formulation. Let the locations to be disinfected by the UV-C drones within a premise be modelled as a graph, $G(V, A)$, where V denotes the set of vertices (e.g., specific locations inside a campus as well as the base station) and A is the

set of edges (e.g., the paths connecting the locations). As demonstrated in Fig. 2, the red arrows represent the edges in A, v_0 indicates the base station, and $v_1 ... v_n$ represent the locations to be sterilized where $\{v_0, v_1, v_2 ... v_n\} \in V$. Each premise contains n_l locations. The x and y coordinates of the center of each location along with the area of the location are pre-determined from the map of the premise. A weight is assigned to each member in A depending on the energy required $E_{D(v_i,v_j)}$ to travel from v_i to v_j, where $v_i, v_j \in V$.

Given the graph $G(V, A)$, our research challenge can be formally formulated as an optimization problem to minimize the energy consumption $E_{D(r,v_i,v_j)}$ of drones, where each drone r travels from v_i to v_j and is also within its energy-limit while routing to the next location, v_j. Let N_D be the number of drones required for an area. N_D is given as an input to the model. An energy-limit E_{max} is assigned to the problem so that the drones have a reasonable amount of remaining energy to safely return to the base station. Therefore, the problem can be expressed as follows:

$$
\begin{aligned}
\min \quad & \sum_{r,i,j \in X} x_{rij} E_{D(i,j)} \\
\text{s.t.} \quad & \sum_{r \in D} \sum_{i \in V, i \neq j} x_{rij} = 1, && \forall\ j \in L \\
& \sum_{j \in V} x_{r0j} \geq 1, && r \in R \\
& \sum_{i \in V, i \neq j} x_{rij} = \sum_{i \in V, i \neq j} x_{rji}, && j \in V, r \in R \\
& \sum_{j \in V} x_{r0j} = \sum_{i \in V} x_{ri0}, && r \in R \\
& if \quad x_{rij} = 1, \quad then \quad u_{ri} + E_{T(i,j)} = u_{rj}, \\
& \qquad r, i, j \in X, i \neq 0, j \neq 0 \\
& if \quad x_{rij} = 1, \quad then \quad u_{rj} >= E_{T(i,j)}, && r, i, j \in X \\
& u_{ri} \leq E_{max}, && i \in L, r \in R
\end{aligned}
\tag{4}
$$

where $x_{r,i,j} \in \{0, 1\}$, signifies whether drone r will take the path from v_i to v_j. X denotes a set containing N_D number of drones and the edges of the graph, i.e., $X \in \{R, A\}$ where R is the set of N_D drones. u_{ri} indicates the cumulative energy consumption of drone r up to a particular location v_i. It is important to keep track of the cumulative energy so that each drone's route limitation can be considered in the calculation. Since the energy required to disinfect a location, E_{UV}, cannot be reduced, the energy required to travel from one location to another, E_D, needs to be reduced while constructing the routes. Hence, the objective function states that the total energy consumed by all the assigned drones while travelling a distance from v_i to v_j should be minimized. The first constraint ensures that each location is visited only once by one drone. The second constraint states that each drone must leave the base station at least once for the sterilization task. The third constraint is given to confirm that the number of drones arriving at each location and entering the base station is equal to the number of drones leaving the base station. The fourth constraint ensures that the number of times the drones leave the base station is equal to the number of times the drones return to the base station. It is important to ensure that all the drones safely return to the base station after completing the disinfection task. The fifth constraint calculates the cumulative energy of drone r till point v_j by adding the total energy consumed in v_j to the cumulative energy till v_i if the path from v_i to v_j is active for drone r. The sixth constraint states that if a path between i and j is active, the cumulative energy till v_j must be more than or equal to the total energy from v_i to v_j for drone r. The last constraint denotes that each individual drone's cumulative energy must be less than or equal to the maximum energy capacity, E_{max}.

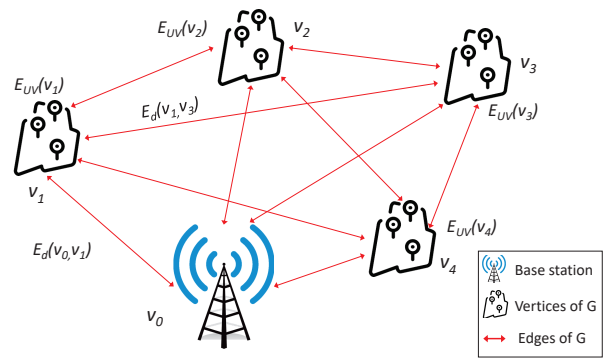


Fig. 2: A graph-based representation of campus locations.

The mathematical model presented in eq. (4) to create an optimal route for the UV-C drones is NP-hard. This was deduced from the Capacitated Vehicular Routing Problem (CVRP), due to their similar properties, that is an NP-hard problem [25] of multiple routes planning for υ number of vehicles to serve η customers. The vehicles are limited by their carrying capacity due to the weight and dimension of the load as well as the vehicle. Each customer has a demand that needs to be satisfied by the vehicle, and each vehicle can serve one customer at a time. The objective of CVRP is, therefore, to construct the minimum cost routes to serve the maximum number of customers within its capacity.

Theorem IV.1. *The route assignment problem for the UV-C drones is NP-hard.*

Proof. In an example scenario of a CVRP, suppose B is a set of υ capacitated vehicles and Ω is a set of η customers to be served. Each vehicle has the same capacity θ and the cost of serving each customer, $\omega \in \Omega$, is denoted by ξ_ω. To compare the CVRP problem with our UV-C drone route assignment problem, the example scenario of CVRP is mirrored into our problem. In our case, B is the set of UV-C drones used to sterilize a scenario and Ω is the number of locations to be sterilized. Each drone is capacitated by its energy which is similar to the limited capacity of an individual vehicle, $\beta \in B$. The drones consume ξ_ω amount of energy to sterilize the location ω which is the cost of that route as considered. Therefore, the objective of the UV-C drone route assignment problem is to construct minimal energy consuming routes while sterilizing the maximum number of locations. Since the energy consumed by the drones while sterilizing each location is constant, the cost of the route depends on the energy required to travel from one location to

another which corresponds to the fixed demand of a customer. Therefore, it can be concluded that the energy minimization of each drone while moving from one location to another is directly proportional to the cost minimization of a capacitated vehicle. □

Hence, obtaining an optimal solution for this UV-C drone route construction problem in polynomial time is challenging. Therefore, there is a trade-off between the optimal route in a considerably high execution time or computation of a sub-optimal route in a relatively low execution time. To enforce the latter, various heuristics can be implemented, thus computing a sub-optimal solution in a shorter time period. Thus, heuristic methods can be considered more practical for constructing our UV-CDS model to make the computation efficient and scalable. In this vein, in the following section, we investigate several techniques that can be considered as potential candidates for solving this problem.

V. Proposed Method

Our aim is to relax the formulated problem in Sec. IV by developing heuristics to find a suitable schedule for the drones in a reasonable amount of time. An optimal solution can be derived for smaller number of vertices/locations in scenarios such as in Fig. 2 using commercially available solvers, e.g., Gurobi [26]. To find an optimal solution for eq. (4) using this solver, a bounded time, t_b is set. However, with an increasing number of locations, the complexity of the problem increases which makes it difficult to find an optimal solution. Therefore, the problem stated is addressed with various heuristic algorithms to obtain a sub-optimal solution. To construct an efficient UV-CDS, it needs to be made sure that the algorithm to be implemented can provide a solution for all scenarios in an uncomplicated manner, which can be achieved by the heuristics. Three heuristics for route construction for drones, i.e., a randomized approach, an energy minimizing greedy approach, and a genetic algorithm-based scheduling approach, are presented in this section to compute sub-optimal solutions.

A. Randomized Drone Scheduling in UV-CDS

In our developed randomized route construction, the route is constructed such that the drones visit all unexplored locations randomly one by one, given their co-ordinates, until the locations are all visited. At first L, E_T, E_{max}, and N_D are initialized where E_T is a list containing the energy consumed by each drone for the route and the rest of the variables defined earlier. The list $route$, containing the route of the drones, is initialized as empty. While all the locations are not added to $route$, it will generate a random index, and if the index is already visited, it selects the next unvisited index. The drone starts from the base station and keeps on moving forward randomly until it reaches the energy limit for the nth drone, i.e., $E_{T\,n} \geq E_{max}$. Once the drone crosses the energy assigned to it, it returns to the base station, and starts a new route after getting recharged.

Even though the random algorithm utilizes a simple and adaptable approach used in various contexts, it is not a good fit for our particular purpose due to the energy constraint. The problem with this algorithm is that it does not regard the energy required to reach destination prior to travelling. Hence, the energy distribution might not always be the best one. Therefore, it will spend most of its energy in travelling rather than disinfecting the areas; thus making the procedure inefficient.

B. Greedy Drone Scheduling with Energy Minimization in UV-CDS

To overcome the challenges arising from the randomized algorithm-based drone route construction in UV-CDS, we develop a greedy heuristic approach to account for the energy minimization problem while constructing routes. The greedy approach was adopted in various studies to construct optimal routes [27], [28]. To implement this concept in our scenario, the total energy while travelling (E_D) is considered as the cost of each route. The steps are similar to random algorithm, however, instead of generating a random index, the greedy algorithm calculates energy required to travel from current location loc to all the unvisited routes and selects the location which consumes the least energy. While constructing the route, the algorithm also needs to ensure that the drone does not cross the overall energy limitation (E_{max}) assigned to them. If the drone crosses the energy limit for a specific route, it simply returns to the base station, and a new route is started from the base station.

While the greedy algorithm may be able to perform well for various cases, it is not without shortcomings. Since it selects the most suitable node at a given moment instead of finding a solution globally, it is only practical when each sub-problem's optimal solution leads to the overall optimal solution, i.e., if the problem has an optimal substructure. Thus, the locally optimal solution may not always lead to the globally optimal solution [29]. Therefore, we need to develop a more compact and generalized solution that will be able to find the best route in all cases without any conditions.

C. Genetic Algorithm-based Drone Scheduling in UV-CDS

To avoid the shortcoming of the greedy-based approach, we now aim to develop a more advanced heuristic approach using the genetic algorithm. The reason behind considering the genetic algorithm is its efficiency for achieving a sub-optimal solution for the stated disinfection problem within reasonable requirements. It was previously adopted in various studies where significant performance improvement was demonstrated [30]–[32]. The genetic algorithm calculates the best route based on its predecessors' and its own fitness, as shown in Algorithm 1. The inputs of the algorithm are the x, y coordinates of each location (x_c, y_c), and the algorithm provides the best route as its output. At first, the variables such as population size S_p, generations Gen, elite size S_e, and mutation rate μ_r are initialized. Then, $locList$ is created which contains the x_c and y_c of the locations, where each location is addressed as a $gene$. Once the variables are all initialized, the genetic algorithm is implemented from line 19. Initially, a population is created from $locList$ of size S_p

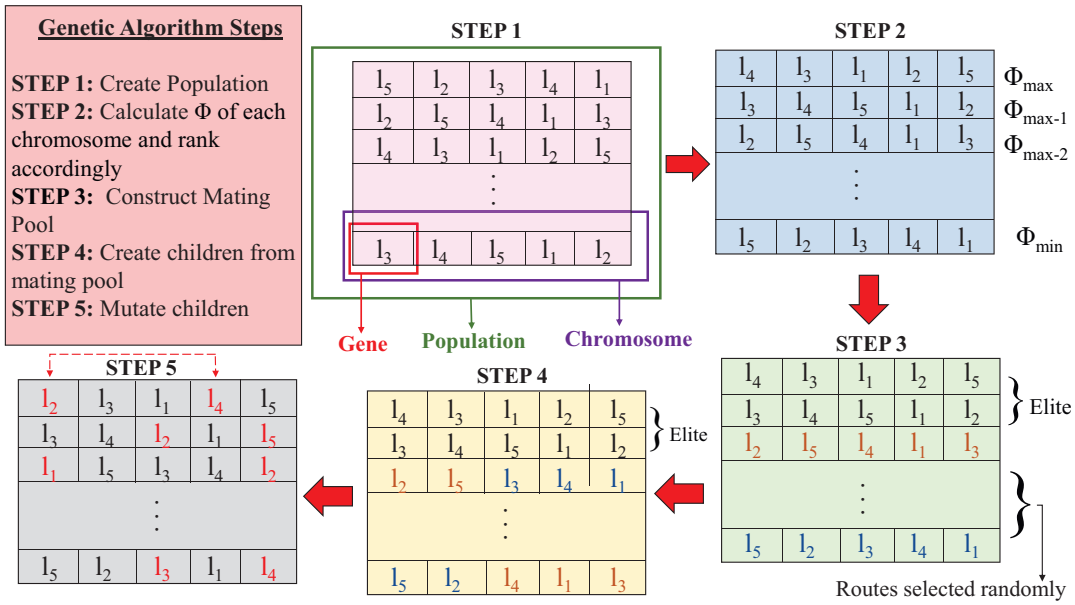


Fig. 3: Step-by-step representation of the developed genetic algorithm for generating a new population.

using the function $createRoute$. This function constructs a route based on our requirements. A variable e_t is kept for tracking the total energy, which is calculated by adding the existing e_t to the total energy of the new location $E_T(loc)$. Once the drone consumes E_{max} energy, it returns to the base station (bs) and starts a new route by adding bs to the route. The population contains a list of S_p randomized routes, also known as $chromosomes$, that can be considered as potential candidates for the calculation. After the population is initialized, a series of further steps are carried out for G generations to calculate the final population. At first, the fitness function, Φ, is calculated using the following:

$$\Phi_{ij} = 1/E_D(i,j). \tag{5}$$

Since we need to minimize E_D, Φ is inverted so that the performance of a route is proportional to the fitness. In this case, the fitness function of all the routes in one scenario was calculated altogether, so that the effectiveness of the whole scheme, instead of individual routes, may be determined at once.

After calculating Φ, the chromosomes are ranked according to their fitness value. From the ranked population, the first S_e chromosomes are chosen for creating the mating pool, (M_P). S_e can be selected as a ratio of the total population. The remaining chromosomes are then assigned a weight and they are selected randomly by the weights for the mating pool. The mating pool consists of a list of high-performing parents, which are supposed to construct a more energy-efficient chromosome. This method of selecting the parents based on their Φ and S_e is called elitism. Then the selected parents in the mating pool are used to create new chromosomes or children using the crossover technique also known as breeding technique. In the crossover method, partial routes are selected from both parents from M_P and the new set of chromosomes containing the children genes are found. Another final and important step is the mutation of the newly created population. During the mutation, the locations or genes within a chromosome are randomly swapped. This is important as it adds variation to the population by circumventing local convergence. The process is repeated for G generations, and a final mutated population is found from the last generation. The final population is once more ranked according to Φ. The first element in the ranked list is taken as the best route for the problem. The summarized steps of the complete algorithm is visually represented in Fig. 3.

Due to the genetic algorithm's scalability and ability to provide a global solution, it can be regarded as the most viable scheduling algorithm for the disinfecting drones in our considered UV-CDS model. In the following section, we perform a rigorous analysis of the candidate algorithms to measure their performance in terms of time and space complexity.

VI. Algorithmic Analysis

In this section, the time and memory consumption of the developed algorithms are analyzed to evaluate their computational overheads.

First, the randomized route construction algorithm for the drones in UV-CDS is rather simple as it selects the locations arbitrarily rather than following any specific rules. The algorithm selects a location randomly for each drone, while all the locations are not yet sterilized. If the randomly selected location has been visited before, it chooses the next available location. Therefore, in the worst case, it requires a time of $O(n^2)$ where n is the number of locations. The space complexity of the algorithm is $O(m \times N_D)$ where m is the length of routes.

Next, we analyze the algorithmic complexity of the greedy route construction algorithm for drones with energy minimization. This algorithm calculates the required energy for travelling from the current location to all the unvisited locations and the location that consumes the least amount of energy to reach is visited next, for each drone. Therefore, this algorithm

Algorithm 1: Pseudo code of genetic algorithm-based drone scheduling in UV-CDS.

Input : x_c, y_c
Output: route

Function `createRoute`($locList$):
  $e_t \leftarrow 0$, $route \leftarrow \emptyset$
  for $loc \in locList$ **do**
    $e_t \leftarrow e_t + E_T(loc)$
    if $e_t \geq E_{max}$ **then**
      Append bs to route, $e_t \leftarrow 0$
    Append loc to $route$
  return $route$

Function `crossover`(M_P):
  $children \leftarrow \emptyset$
  for $Parent_1$ & $Parent_2$ *in* M_P **do**
    $child \leftarrow \emptyset$
    $C_1 \leftarrow$ Select random subset of $Parent_1$
    $C_2 \leftarrow$ Select remaining subset of $Parent_2$
    Append C_1 & C_2 to $child$, Append $child$ to $children$
  return $children$

Initialize S_p, Gen, S_e, μ_r
$locList \leftarrow [xc, yc]$
for i *in* S_p **do**
  $population \leftarrow createRoute(locList)$
for g *in* G **do**
  Calculate Φ for each route in population
  $rankedList \leftarrow$ Rank population based on Φ
  $M_P \leftarrow$ select S_e routes from $rankedList$
  $children \leftarrow crossover(M_P)$
  $population \leftarrow$ Mutate $children$ by randomly swapping two locations in a route
$finalRank \leftarrow$ Rank final $population$ in descending order based on Φ
$route \leftarrow$ First route in $finalRank$
return $route$

also requires a time of $O(n^2)$ is the worst case scenario. Additionally, the algorithm consumes a memory of $O(m \times N_D)$ while executing, similar to the randomized approach.

For the genetic algorithm-based scheduling, there is a series of steps such as ranking the routes, R, selecting the elite chromosomes for the mating pool, S, construct children using crossover function C and finally mutating the children, M. Therefore, for making a population Pop in each generation, the time complexity required can be represented as follows:

$$O(Pop) = O(R) + O(S) + O(C) + O(M). \tag{6}$$

To rank the routes, the computation requires calculating the fitness function of each route in the population. The fitness function is determined by visiting each location in a route and calculating the distance between them to obtain E_D. Therefore, ranking the routes takes a time of $O(S_p \times L_R)$ where S_p is the size of the population and L_R is the length of the route.

Subsequently, the elite chromosomes are taken from the ranked population. Therefore, the time taken to select the elite chromosomes for the mating pool can be expressed as $O(S_e)$ where S_e denotes the elite size. To select the other routes randomly except for the elite routes, a time of $O(S_p(S_e - S_p))$ is required for S_p chromosomes. Therefore, the overall complexity for the selection procedure is given by:

$$O(S) = O(S_e) + O(S_p(S_e - S_p)). \tag{7}$$

In the mating pool containing a list of possible parents, a crossover method is implemented to create the children. The elite ones are selected as children with $O(S_e)$ complexity. To perform a crossover, a chunk, g_1, is selected from the first parent (P_1). This requires $O(g_1)$ time to execute. Similarly, a portion of the second parent (P_2), i.e., g_2, is selected which needs $O(g_2)$ time. Therefore, the time complexity of the entire process of breeding is represented by:

$$O(C) = O(S_e) + O(S_m - S_e) \times O(g1 + g2), \tag{8}$$

where S_m represents the size of the mating pool.

Lastly, the mutation steps require a time of $O(L_r S_p)$ for accessing each location of each route. Overall, the population, Pop, is generated Gen times. Therefore, the overall time complexity of the genetic algorithm, GA, can be expressed as follows:

$$O(GA) = O(Gen \times Pop)$$
$$O(GA) = O(GenR) + O(GenS) + O(GenC) + O(GenM). \tag{9}$$

Additionally, since, the routes are considered as one list during the calculation, the memory consumption of the genetic algorithm is $O(m)$, where m denotes the length of the whole route. Therefore, it can be concluded that the genetic algorithm consumes comparatively more time and less memory than the greedy and random algorithm for computing the routes for the disinfecting drones. However, for constructing UV-CDS, it is important to take into account the aforementioned tradeoff by selecting an algorithm capable of minimizing the energy consumption of the drones by constructing sub-optimal routes while consuming a reasonable amount of time and memory. The simulation-based results in the following section will further corroborate the presented complexity analysis.

VII. Performance Evaluation

In this section, we evaluate the performance of our proposal based on a university campus use-case, which can be regarded as a hotspot for COVID-19 transmission. Our considered model and the developed algorithms were implemented in the campus map by making realistic assumptions for each location in the campus. The total area of the campus was considered to be $2000 \times 1000\ m^2$ as shown in Fig. 4. The figure demonstrates an instance of the used map with an example scenario having 20 locations. In this scenario, it can be seen that a drone needs to take two turns to completely sterilize the entire institution due to the energy restriction of the drone. Here, the path of the drone is determined by the Gurobi solver for $N_D = 1$. The red square represents the base station from where the drone leaves and returns to, and the other markers represent

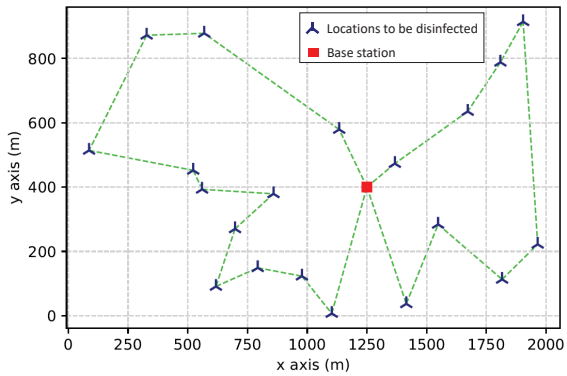
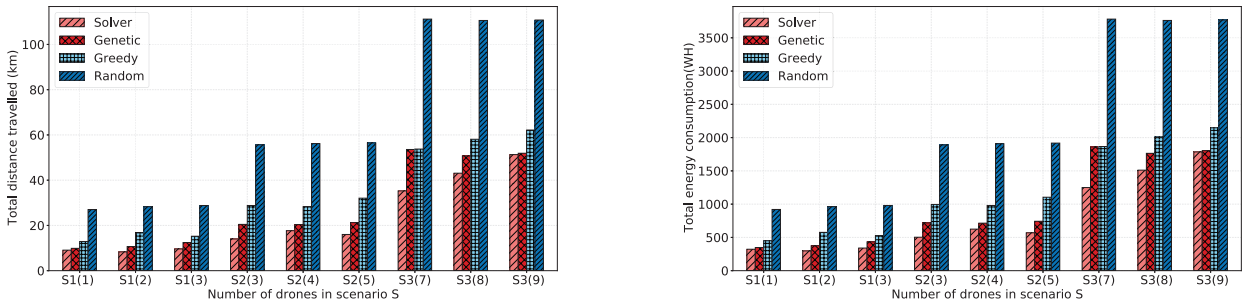


Fig. 4: A map demonstrating the computed routes in an example scenario of a premise containing 20 locations.

the locations to be disinfected in the scenario. Next, to find the effectiveness of the proposed algorithms, we describe the various parameters, such as the total energy consumed and distance travelled by the drones for each approach, and then compared their performances. For the genetic algorithm-based drone scheduling, the hyper-parameters were set as follows: $S_p = 500$, $Gen = 500$, $S_e = 20\%$ of S_p, and $\mu_r = 0.01$. For all the generated random values, the random seed was fixed so that it produces the same values for all executions. A random seed of 31 was used for this case. Additionally, for the randomized approach, the random seed was varied within a range of 30 to 50, and the average of all the execution results was reported as result. The results were all generated in a device having an Intel Core i7, 3.00 GHz Central Processing Unit (CPU), 16GB Random Access Memory (RAM), powered by NVIDIA RTX 2060 Graphics Processing Unit (GPU).

To synthesize the dataset required for experiments, the locations of institutions were randomly assigned. The area of each location was considered between 50m^2 to 150m^2 which approximate represent a regular-sized classroom and large auditoriums, respectively. Based on [5], it was assumed that an area of 1cm^2 requires 3mJ or 8.33×10^{-7}Watt hour (Wh) of energy. Therefore, for each meter squared of area, the required energy was 8.33×10^{-3}Wh. For calculating the energy required for travelling of the drone, the kinetic energy was calculated using eqs. (2) and (3). s is the distance between two locations and the acceleration of the drone was considered as 15m/s^2. Therefore, the energy consumed per meter was 0.033Wh. URUAV GRAPHENE 6S 22.2V 6000mAh 100C Lipo Battery XT90 [33] having an energy capacity of 133.2Wh was taken as the battery of the drone. This battery is commonly used for heavy duty tasks. Therefore, it was considered as the energy source for the UV-C drones. E_{max} was calculated by considering $\rho = 75\%$ of the total capacity of the battery, i.e., approximately 100Wh for this particular case. Various types of drones can be used for the purpose of disinfection, such as agricultural (Ag) drones (e.g., PrecisionHawk Lancaster 5, senseFly eBee SQ, Sentera Omni Ag, and so on) or more lightweight ones like DJI Matrice 300 RTK, Autel EVO II, and so forth. It is important to ensure that the drones used in the process have the capability to support the battery requirements specific to the considered scenario. Particularly for larger scenarios, a battery unit with a higher capacity or customized configuration (e.g., with a higher number of cells combined in a series) may be employed to have a greater energy capacity for our considered disinfection mission. Apart from that, the Euclidean distance between the locations was calculated for distance measurement. Based on these values, six scenarios, denoted by S1, S2, S3, S4, S5, and S6, were implemented. The number of locations in scenarios S1, S2, S3, S4, S5 and S6 was set to 25, 50, 100, 150, 200, and 1000, respectively. The scenarios were implemented for a fixed number of drones which depends on n_l. In our case, we assumed that for each $n_l = 25$, $N_D = 2$. To test with various N_D values for one scenario, the value of N_D was varied in the range $N_D \pm 1$. It was assumed that if the number of locations are not all visited and the drone is out of 75% of the energy, it would return to the base station for recharging before becoming mission-ready again to sterilize the remainder of the non-sterilized locations.

Fig. 5 represents the total energy consumed and distance travelled by varying the values of N_D for the three scenarios S1, S2, and S3. For these scenarios, t_b was considered to be 1500s for the solver. In the figures, the bars of Si(j) denote the amount of total energy consumed and distance travelled by i drones to completely perform disinfection in the scenario Si. For instance, the bars of S1(1) and S1(2) demonstrate the total amount of energy consumed and distance travelled by a single UAV and two drones, respectively, to complete the same task in the scenario S1 and so forth. It can be noticed, in Fig. 5(a), that the path calculated by the Gurobi solver takes the shortest distance for all cases. On the other hand, the randomized approach results in the highest distance travelled by the drones. In contrast to the randomized approach, the other two algorithms require the drones to travel much lower distances. Particularly, the genetic algorithm-based scheduling demonstrates remarkably close performance to the optimal case in terms of the distance travelled by the drones. The genetic algorithm performs the second best with distances after the solver. The greedy approach performs moderately well compared to its randomized counterpart. On the other hand, in Fig. 5(b), it can be seen that the solver-derived paths consumed the least energy compared to other methods. The genetic algorithm also consumed less energy compared to the greedy and randomized approaches in most cases. Also, the greedy approach consumes comparatively less energy than the randomized algorithm in all the scenarios. Additionally, the variation of N_D does not have much effect in both energy and distance. However, with an increasing value of N_D, the energy consumption and distance also slightly increase in most cases. The energy required by the drones to disinfect each scenario was also analyzed, as demonstrated in Fig 6. As shown in this figure, the energy required to disinfect a maximum number of locations, i.e., scenario S6, which disinfects a total area of around 89000m^2 was approximately 750Wh, which can be distributed among multiple drones for plausible implementation. Therefore, it may be concluded that large areas can be disinfected if this model is efficiently used. Fig. 7 shows the kinetic energy with respect to the velocity of the drones. With increasing velocity, the energy required for travelling (E_D) also increases exponentially. Therefore, it is important to find a relevant velocity for the drones that will



(a) Performance in terms of distance travelled of the drones.

(b) Performance in terms of energy consumption of the drones.

Fig. 5: Results of two example scenarios in terms of the distance travelled and energy consumption of the drones carrying out disinfection missions at the target premise.

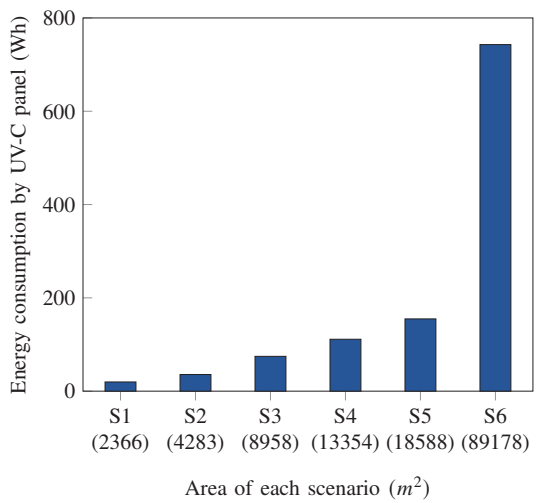


Fig. 6: Energy consumption of the UV-C panel for the considered scenarios.

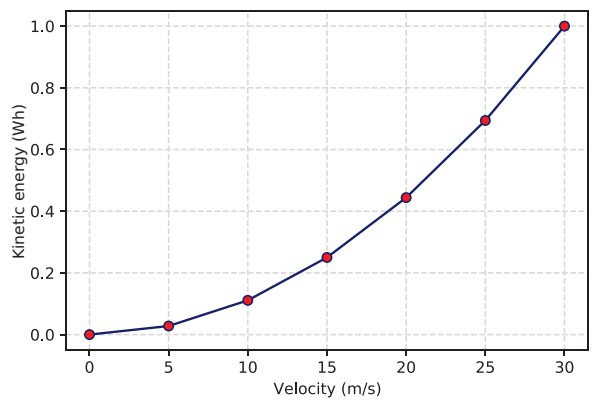


Fig. 7: Kinetic energy versus increasing velocity of drones.

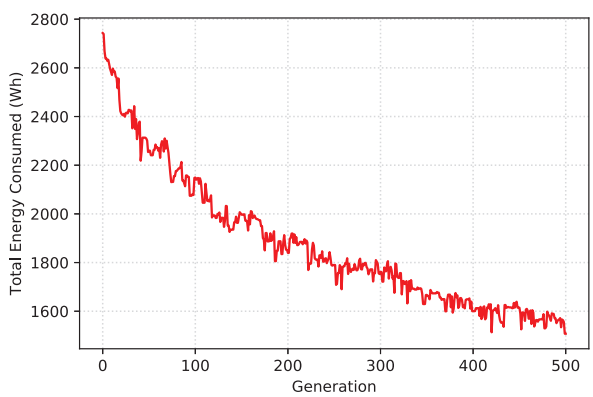


Fig. 8: Variation of energy consumed by the drones in genetic algorithm with growing generations.

be able to complete the task with reasonable energy and time.

Table I represents the energy consumption and distance travelled for the remaining scenarios, i.e., S4, S5, and S6. For these scenarios, t_b was considered to be 3000s for the solver. It can be noticed from the results that the trend continued for a larger number of locations. However, for S6, the solver was unable to give a feasible solution, even for a longer period of time. Although the energy consumed and distance travelled were a bit higher for the heuristic approaches in S6 due to the larger number of locations, they could provide solutions unlike the solver. Thus, despite the sub-optimal solutions given by the heuristic approaches, they can be applied to any scenario for the purpose of scheduling UV-C drones. Among the considered heuristics, the genetic algorithm-based drone scheduling demonstrated the best performance in terms of accomplishing the task with the least amount of distance and energy.

The execution time of the solver and genetic algorithm was also measured, since they performed the best as shown in Table II. It can be seen that the genetic algorithm could solve the problem in all scenarios within a reasonable amount of time with the highest being approximately 1922 sec for S6. On the other hand, the solver was not able to finish the task in the given time limit. Despite running for approximately 4 hours, no optimal solution was found for scenario S1. To further investigate the genetic algorithm, a graph was plotted in Fig. 8 demonstrating how the energy changed with generation for scenario S3 with 8 drones. It can be noticed that the total energy required while travelling, E_D, drastically decreased with evolved generations within a short time. The energy shown in generation 0 is the initial energy of the population, set by the algorithm; and later at 500 generations, the energy consumption significantly dropped to around 1500Wh.

Therefore, it can be concluded that even though the solver performs the best, it has some shortcomings. With an increas-

TABLE I: Energy consumption and distance travelled in scenario S4, S5 and S6.

Scenario	Number of locations	Drone (s)	Energy consumed (Wh)				Distance travelled (m)			
			Solver	Genetic algorithm	Greedy algorithm	Randomized algorithm	Solver	Genetic algorithm	Greedy algorithm	Randomized algorithm
S4	150	11	2394.7	3115.7	3176.8	5821.7	68.5	90.1	92	171.3
		12	2552.9	3014.7	3339.4	5890.5	73.2	87.1	96.8	173.4
		13	3419	3073.2	3487.5	5919.1	99.2	88.9	101.3	174.2
S5	200	15	2777.6	4439.3	5208.3	7650.8	78.7	128.6	151.6	224.9
		16	2723.6	4598.2	5411.4	7748.4	77.1	133.3	157.7	227.8
		17	3159.4	4664.0	5615	7736.4	90.1	135.3	163.8	227.4
S6	1000	39	Infeasible	28450.4	32797.6	37452.1	Infeasible	831.2	911.6	1101.3
		40		28173.2	32950	37739.3		822.9	916.2	1109.9
		41		27909.3	33095.6	37496.1		815	920.6	1102.6

ing number of locations, it becomes more complex due to the increased number of variables, and hence it is not able to perform well for larger premises. On the other hand, the genetic algorithm-based drone schedule yields a sub-optimal performance within a reasonable time for all cases. Although the randomized and greedy implementations require less time to execute, the performance of the genetic algorithm is superior to those of the randomized and greedy approaches in terms of energy and distance minimization that corroborate with the analysis in section VI. Hence, the genetic algorithm emerges as the most viable algorithm for our considered UV-CDS system model in terms of performance with a slight trade-off with execution time. Thus, our proposed genetic algorithm-based drone scheduling can be employed in larger premises along with a higher number of locations requiring disinfection. This can be accomplished by increasing the number of drones for the task and using batteries with a higher capacity to ensure that the drones are able to improve disinfection area coverage. The complexity of the proposal will not be drastically affected by the increased area since it does not affect the creation of population in the genetic algorithm. Thus, the model can be considered scalable in terms of premises having larger areas.

VIII. Conclusion

In this paper, we proposed the UV-CDS (Ultraviolet-C Drone-based Sterilization) system and discussed the need to develop an energy-efficient flight path scheduling for the UV-C drones to perform on-premises sterilization. We showed that scheduling problem is computationally hard, and developed several heuristics based on a randomized approach, a greedy approach, and a genetic algorithm-based scheduling for the disinfecting drones to obtain reasonable solutions at the expense of acceptable computational time and memory overheads. Our developed heuristics' performances were evaluated in six scenarios in university campus use-case that consisted of varying numbers of infected locations and a fixed number of drones for each scenario. Simulation results demonstrated that the genetic algorithm emerges as the most viable energy-efficient scheduling technique for sterilizing various premises.

TABLE II: Simulation time for the two best performing models.

Scenario	Genetic algorithm (sec)	Gurobi solver
1	233.6	-
2	268.3	-
3	324.3	-
4	407.4	-
5	466.1	-
6	1922.8	Infeasible

References

[1] Z. Fadlullah, M. M. Fouda, A. S. K. Pathan, N. Nasser, A. Benslimane, and Y. D. Lin, "Smart IoT solutions for combating the COVID-19 pandemic," *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 10–11, Sep. 2020.

[2] S. Sakib, T. Tazrin, M. M. Fouda, Z. M. Fadlullah, and M. Guizani, "DL-CRC: Deep learning-based chest radiograph classification for COVID-19 detection: A novel approach," *IEEE Access*, vol. 8, pp. 171 575–171 589, Sep. 2020.

[3] M. Buonanno, G. Randers-Pehrson, A. W. Bigelow, S. Trivedi, F. D. Lowy, H. M. Spotnitz, S. M. Hammer, and D. J. Brenner, "207-nm UV light-a promising tool for safe low-cost reduction of surgical site infections. I: in vitro studies," *PloS one*, vol. 8, no. 10, p. e76968, Oct. 2013.

[4] 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, Mar. 2018.

[5] A. FERNG, "World's First Indoor Disinfection Drone Ready to Fight COVID-19," Apr. 2020, last accessed: 29 Nov. 2020. [Online]. Available: https://www.medgadget.com/2020/04/worlds-first-indoor-disinfection-drone-ready-to-fight-covid-19.html

[6] Z. Wang, H. Duan, and X. Zhang, "An improved greedy genetic algorithm for solving travelling salesman problem," in *2009 Fifth International Conference on Natural Computation*, vol. 5, 2009, pp. 374–378.

[7] I. H. Hamzavi, A. B. Lyons, I. Kohli, S. Narla, A. Parks-Miller, J. M. Gelfand, H. W. Lim, and D. Ozog, "Ultraviolet germicidal irradiation: possible method for respirator disinfection to facilitate reuse during COVID-19 pandemic," *Journal of the American Academy of Dermatology*, no. 6, pp. 1511–1512, Jun. 2020.

[8] B. Sarada, R. Vijay, R. Johnson, T. N. Rao, and G. Padmanabham, "Fight against COVID-19: ARCI's technologies for disinfection," *Transactions of the Indian National Academy of Engineering*, vol. 5, no. 2, pp. 349–354, Jul. 2020.

[9] M. Guettari, I. Gharbi, and S. Hamza, "UVC disinfection robot," *Environmental Science and Pollution Research*, Oct. 2020, doi: 10.1007/s11356-020-11184-2.

[10] S. Sahu, S. K. Pandey, and A. Mishra, "Disinfectant chamber for killing body germs with integrated far-uvc chamber (for covid-19)," *International Journal of Design Engineering*, vol. 10, no. 1, pp. 1–9, Feb. 2021.

[11] M. Würtele, T. Kolbe, M. Lipsz, A. Külberg, M. Weyers, M. Kneissl, and M. Jekel, "Application of gan-based ultraviolet-C light emitting diodes--UV LEDs--for water disinfection," *Water research*, vol. 45, no. 3, pp. 1481–1489, Jan. 2011.

[12] G. L. Curtis, M. Faour, M. Jawad, A. K. Klika, W. K. Barsoum, and C. A. Higuera, "Reduction of particles in the operating room using ultraviolet air disinfection and recirculation units," *The Journal of arthroplasty*, vol. 33, no. 7, Supplement, pp. S196–S200, Jul. 2018.

[13] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, F. Ono, and R. Miura, "On a novel adaptive UAV-mounted cloudlet-aided recommendation system for LBSNs," *IEEE Transactions on Emerging Topics in Computing*, vol. 7, no. 4, pp. 565–577, 2019.

[14] F. Tang, Z. M. Fadlullah, N. Kato, F. Ono, and R. Miura, "Ac-poca: Anticoordination game based partially overlapping channels assignment in combined uav and d2d-based networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 1672–1683, 2018.

[15] N. Kato *et al.*, "Optimizing space-air-ground integrated networks by artificial intelligence," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 140–147, 2019.

[16] J. Liu, Y. Shi, Z. M. Fadlullah, and N. Kato, "Space-air-ground integrated network: A survey," *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2714–2741, 2018.

[17] V. Walter, N. Staub, A. Franchi, and M. Saska, "UVDAR system for visual relative localization with application to leader–follower formations of multirotor UAVs," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2637–2644, Jul. 2019.

[18] "Xenex LightStrike Robot Destroys SARS-CoV-2 (Coronavirus) in 2 Minutes; First & Only UV Disinfection Technology Proven to Deactivate COVID-19 Virus," Apr. 2020, last accessed: 29 Nov. 2020. [Online]. Available: https://www.businesswire.com/news/home/20200430005656/en/

[19] "XDBOT set for Covid-19 cleaning duties in Singapore," Apr. 2020, last accessed: 29 Nov. 2020. [Online]. Available: https://www.theengineer.co.uk/xdbot-set-for-covid-19-cleaning-duties-in-singapore/

[20] Z. M. Fadlullah, D. Takaishi, H. Nishiyama, N. Kato, and R. Miura, "A dynamic trajectory control algorithm for improving the communication throughput and delay in uav-aided networks," *IEEE Network*, vol. 30, no. 1, pp. 100–105, 2016.

[21] S. Jia, Z. M. Fadlullah, N. Kato, and L. Zhang, "ECO-UDC: An energy efficient data collection method for disaster area networks," in *2016 IEEE International Conference on Network Infrastructure and Digital Content (IC-NIDC)*. IEEE, Sep. 2016, pp. 130–134.

[22] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, "AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks," *IEEE Internet of Things Journal*, vol. 8, no. 2, pp. 1211–1223, 2021.

[23] Z. Chen, J. Zhou, R. Sun, and L. Kang, "A new evolving mechanism of genetic algorithm for multi-constraint intelligent camera path planning," *Soft Computing*, pp. 1–20, Jan. 2021.

[24] Y. Pan, Y. Yang, and W. Li, "A deep learning trained by genetic algorithm to improve the efficiency of path planning for data collection with multi-uav," *IEEE Access*, vol. 9, pp. 7994–8005, Jan. 2021.

[25] S. N. Kumar and R. Panneerselvam, "A survey on the vehicle routing problem and its variants," *Intelligent Information Management*, vol. 4, no. 3, pp. 66–74, May 2012.

[26] G. Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[27] X. Geng, Z. Chen, W. Yang, D. Shi, and K. Zhao, "Solving the traveling salesman problem based on an adaptive simulated annealing algorithm with greedy search," *Applied Soft Computing*, vol. 11, no. 4, pp. 3680–3689, Jun. 2011.

[28] K. Karabulut and M. F. Tasgetiren, "A variable iterated greedy algorithm for the traveling salesman problem with time windows," *Information Sciences*, vol. 279, pp. 383–395, Sep. 2014.

[29] Y. Zhang and M. S. Waterman, "An eulerian path approach to global multiple alignment for DNA sequences," *Journal of Computational Biology*, vol. 10, no. 6, pp. 803–819, Jul. 2004.

[30] E. Ruiz, V. Soto-Mendoza, A. E. R. Barbosa, and R. Reyes, "Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm," *Computers & Industrial Engineering*, vol. 133, pp. 207–219, Jul. 2019.

[31] W. Ho, G. T. Ho, P. Ji, and H. C. Lau, "A hybrid genetic algorithm for the multi-depot vehicle routing problem," *Engineering applications of artificial intelligence*, vol. 21, no. 4, pp. 548–557, Jun. 2008.

[32] M. Albayrak and N. Allahverdi, "Development a new mutation operator to solve the traveling salesman problem by aid of genetic algorithms," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1313–1320, 2011.

[33] "URUAV GRAPHENE 22.2V 6000mAh 100C 6S Lipo Battery XT90 Plug for RC Racing Drone." [Online]. Available: https://m.uruav.com/URUAV-GRAPHENE-22_2V-6000mAh-100C-6S-Lipo-Battery-XT90-Plug-for-RC-Racing-Drone-p-506.html

Tahrat Tazrin received the B.Sc. degree in Computer Science and Engineering from BRAC University, Dhaka, Bangladesh, in 2018. She is currently pursuing the graduation degree with the Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada. Her research interest in data analytics applications with a particular focus on developing logic-in-sensor EEG headsets.

Mostafa M. Fouda (SM'14) is currently an Assistant Professor with the Department of Electrical and Computer Engineering at Idaho State University, ID, USA. He also holds the position of Associate Professor at Benha University, Egypt. He received his Ph.D. degree in Information Sciences from Tohoku University, Japan in 2011. His research interests include cyber security, machine learning, blockchain, IoT, 6G networks, and smart grid communications. He is an Editor of IEEE Transactions on Vehicular Technology and an Associate Editor of IEEE Access.

Zubair Md Fadlullah (SM'13) is currently an Associate Professor with the Computer Science Department, Lakehead University, and a Research Chair of the Thunder Bay Regional Health Research Institute (TBRHRI), Thunder Bay, Ontario, Canada. He was an Associate Professor at the Graduate School of Information Sciences, Tohoku University, Japan, from 2017 to 2019. He received his Ph.D. degree in Information Sciences from Tohoku University, Japan in 2011. His main research interests span over the cyber physical system layers in sensing, communication, and computing problems and elegant solutions.

Nidal Nasser (SM-IEEE) completed his Ph.D. in the School of Computing at Queen's University, Kingston, Ontario, Canada, in 2004. He is currently a Professor of Software Engineering in the College of Engineering at Alfaisal University, Saudi Arabia. He worked in the School of Computer Science at University of Guelph, Guelph, Ontario, Canada. Dr. Nasser was the founder and Director of the Wireless Networking and Mobile Computing Research Lab @ Guelph. He is currently the founder and Director of the Telecommunications Computing Research Lab @ Alfaisal University. Dr. Nasser received several outstanding research awards as well as several best paper awards.