Results in Control and Optimization 9 (2022) 100174

Contents lists available at ScienceDirect

Results in Control and Optimization

journal homepage: www.elsevier.com/locate/rico

Optimal waypoint assignment for designing drone light show formations

Dharna Nar *, Radhika Kotecha

K. J. Somaiya Institute of Engineering and Information Technology, Mumbai, India

ARTICLE INFO

Keywords:
Swarm drones
Multi-UAVs
Optimal assignment
Swarm drones light show
Static drone formatics

ABSTRACT

Advancements in technology and new drone policies have boosted the adoption of drones by various industries. With active research and development, ingenious applications using drones are being developed. One such innovative and creative application of swarm drones is the drone light show. Light shows are generally performed outdoors in a dedicated clear open air space, which attracts the usage of drones for the execution. A fleet of multiple UAVs illuminates the night sky with picturesque performances in a synchronized manner. Use of drones in light shows provides an eco-friendly and reusable alternative for unique ways of advertisements, celebrations, storytelling, and entertainment. To choreograph the performance of swarm drones means allocating waypoints to each drone for every formation and the complexity of designing increases with the number of drones. An intelligent optimal assignment system is much required to allocate waypoints to drones for each transition in order to create formations. We propose a static swarm-intelligence-based assignment approach named Constrained Hungarian Method for Swarm Drones Assignment (CHungSDA) for optimally assigning multi-UAVs to waypoints. This approach uses Hungarian algorithm as a base with added constraints specific to the targeted application. Several simulations of the proposed approach — carried out for different designs have shown promising results. The work will be of best reference to the UAV manufacturers for their functional design of UAVs and development of Ground Control Station Software to communicate with multiple UAVs, along with other companies that aim to create visual illustrations using UAVs.

1. Introduction

With the latest technological evolution and ancillary functions, drones have become intelligent, compact, and less expensive for deployment and tinkering. These advancements have laid down a plethora of opportunities for more Unmanned Aerial Vehicle (UAV) innovations. Any application of a drone consists of two major components, the hardware and the software. Hardware includes mechanical and electronic components to build a drone that helps it to fly and maneuver. Sensors mounted on the drone help in data acquisition, sensing their environment, communication, localization, and detecting obstacles. The software helps the drone in making informed decisions, improves flight stability, avoids obstacles and navigates its course to their desired path [1,2]. Level of autonomy of the drone and the communication of the drone with the remote controller, ground control station (GCS) or peer UAVs in case of swarm drones applications is handled by the software logic with the help of communication sensors.

Some of the popular applications of drones include target search, land mapping, security and surveillance, reforestation, military, environmental monitoring, visual arts [2], delivery [3], and flying cellular networks [4]. Commercial drones are being deployed in various industries to help companies improve safety, enhance efficiency and accuracy of operations, save time and money.

* Corresponding author.
E-mail addresses: nar.dharna@gmail.com (D. Nar), kotecha.radhika7@gmail.com (R. Kotecha).

https://doi.org/10.1016/j.rico.2022.100174
Received 22 April 2022; Received in revised form 29 July 2022; Accepted 19 September 2022
Available online 28 September 2022

Some of the top industries and sectors that use drones are agriculture, mining, oil and gas, energy, warehousing and logistics and delivery [3,4], films and media and entertainment [2]. A single drone has the potential to perform these tasks. However, the major drawback of deploying a single drone is that it has a limited flight time and payload capacity. In order to complete a given task in a stipulated amount of time, a fleet of UAVs — also known as swarm drones can be deployed. A swarm of drones [3,5,6] can be defined as a fleet of multiple UAVs that work in a coordinated manner to perform a specific task. Swarm drones have found great potential in multiple applications described above. One of the commercial applications of multi-UAVs is light show, which is the subject of this work.

A swarm drone light show is the use of multiple UAVs, often quadrotors, flying in a coordinated fashion for public display. All the drones in the fleet are equipped with LEDs, and the display is held at night. The displays may be for entertainment, storytelling, or advertisement purposes, where the drones use swarming behaviour to display illuminated visual art. The drone show is choreographed in such a way that a fleet of multi-UAVs arrange themselves in order to produce formations of beautiful designs, images and transitions. Using the emerging technology, multi-UAVs can be employed to create eco-friendly light shows. Drone light shows differ from fireworks since fleets of drones are reusable and do not produce air and noise pollution. Also, swarm drone light shows offer flexibility to be held indoors where fireworks display is simply not possible. Drones in the fleet are custom built and can be deployed in multiple shows. The procurement of multiple drones, although an expensive affair, is a one-time big investment with a comparatively smaller maintenance cost. This investment in a reusable fleet of drones can easily start to pay off after just a few shows and is a sustainable approach for occasions or organizations offering light shows often.

1.1. Motivation and problem description

Most common applications of drones are performed in an environment loaded with obstacles like buildings, trees, overhead wires and light poles. An attentive drone pilot or a precise obstacle detection and path planning is required to perform drone operations. As compared to it, drone light shows have an advantage of being performed on an open ground with a clear air-space and permissible geozone, which is one of the prime motivations of this work.

The deployment of a successful drone light show requires various technology components, which are still at an early stages of development. As the technology advances, swarm drone light shows have the potential to get bigger, brighter, denser and for a longer duration. While the common factors that are presently affecting the dissemination of drone light shows are high deployment cost, need for regulatory approval, expensive procurement of drones, safety requirements, limited flight time, labour-intensive operations, etc., one of the major hindrances is the lack of efficient show design tools.

In order to produce a swarm drone light show, the first step to be executed by this tool is to design and simulate the sequence of formations of the show. The effectiveness of the formation and transitions from one design to another depends on the number of drones in the fleet, the waypoints assigned to each drone, and the trajectories of the drones. In order to address these challenges, a drone show designer follows a sequence of steps to design and simulate the entire show:

1. Decide the number of drones, n — to be deployed for creating design formations in the show considering the economic feasibility.
2. Ideate on the creation of different designs with the n drones in a fleet.
3. Assign waypoints to each drone in the fleet to create the desired design for each intermediate transformation. (This is usually done by using an image overlay of the desired design to create a mesh object and then assign drones to the vertices of the mesh object to create that design with drones.)
4. Simulate the formations and validate the show in software. (Herein, the designer closely examines the trajectory of each drone to ensure smooth flights without head-on collisions).

Step 3 is a crucial step in designing a drone light show and an optimal assignment of the waypoints to the drones is critical. Specifically, drones have a limited flight time. In order to simulate all the desired formations in a given amount of time, vertices of the mesh object should be assigned as waypoints to the drones in such a way that the sum of all the distances travelled by all drones to transform from one design to another is minimum. Step 3 can be performed manually if the number of drones in the swarm is less and the designs are simple. As the number of drones in the swarm increases, the design formation complexity also increases. Manual designing is a challenging, tedious, and time-inefficient task. Given a limited flight time, if the total distance to be travelled by all the drones for each transformation is not minimum, it will lead to a shorter duration of the drone light show.

Thus, the proposed work targets to address this issue by developing an optimal assignment method for the targeted application of designing drone light shows. That is, the work aims to develop a decision support model — specifically, an optimal assignment approach for designing a static coordinated swarm drone light show.

The important characteristics of an optimal assignment for this particular application are:

1. Only one waypoint should be assigned to each drone in each formation.
2. All the drones in the swarm should be assigned a waypoint for creating a formation.
3. The cost, i.e., sum of the distances of all the trajectories of drones for transforming into each new formation should be minimum.

Hence, this work proposes an optimized approach of waypoint allocation to the drones for each intermediate transformation for creating designs in drone light shows.

1.2. Contributions of this work

1. The work proposes an efficient swarm drone light show designing tool.
2. The proposed approach assures design with minimal total cost (total distance to be travelled by the swarm drones) through an optimized approach for waypoint allocation for every transformation.
3. The proposed algorithm outperforms other methods for *n*-*n* assignment for the considered application, and it has been demonstrated through experimental results.
4. Although this algorithm discusses an optimal assignment of drones for the application of drone light show, the proposed approach could be adopted in other applications where deployment of static swarm drones is required.

The paper is organized as follows: Section 2 provides an overview of the related work available in the literature. Section 3 presents the proposed approach for an optimal assignment of multiple drones to the waypoints in order to design and simulate swarm drones' light show. Experimental results that demonstrate the performance of the proposed approach are stated and discussed in Section 4. Section 5 concludes the work and highlights the scope for future research and applications.

2. Related work

Swarms of drones can be classified based on their behaviour, level of autonomy, movements, and formations [5]. In terms of behaviour, swarm drones can be categorized as static and dynamic. In static swarm implementation, the number of drones in the fleet and their mission trajectories are pre-planned and no changes occur in real-time while executing the task. Dynamic swarm implementation allows the addition or removal of new UAVs in the fleet and their mission trajectories can be altered in real-time while executing the task. In terms of movement, some of the popular and efficiently implemented movements include stigmergy, flocking, and random movements. The formations are greatly inspired by the observation of the collective behaviour of insects, birds, and fishes. In terms of autonomy, drone swarms can be categorized as semi-autonomous and fully-autonomous. This classification can be further split into single-layered where each drone is its own leader and uses decentralized algorithms to perform given tasks and multi-layered swarms with a dedicated master at each layer or cluster. These leader drones report the status of their cluster to masters at a higher level of the hierarchy. The ground control station is the highest level in this hierarchy. Depending on the application for which drone swarms are being deployed, an appropriate technique can be chosen.

The entire logistics and technology stack for deploying a swarm drones light show is explained in [2]. The study of flight formation techniques has been widely explored [7] by applying social foraging in order to find optimal assignment solutions for developing swarm-based intelligence. Some of these techniques are Particle Swarm Optimization (PSO) [1,8] and Ant Colony Optimization [9].

The following subsections provide a brief review of widely used algorithms and related work for destination assignment in the context of drones.

2.1. Hungarian algorithm

The Kuhn–Munkres algorithm is a combinatorial optimization algorithm that solves the assignment linear-programming problem in polynomial time. This algorithm is also known as the Hungarian matching algorithm or Bipartite Graph Matching. The Hungarian Algorithm locates maximum-weight matchings or minimum-weight matchings in the bipartite graph to discover the optimal solution to the assignment or association problem. An assignment problem is a specific type of transportation problem where the goal is to distribute resources across an equal number of activities in order to reduce overall costs or increase overall allocation profits.

In the assignment problem, workers are represented as source and jobs as destination. The goal is to assign jobs to workers such that the transportation cost is minimum and profits earned are maximum. The supply amount at each source is exactly equal to 1 i.e. assign each worker to one and only one job. Let x_{ij} denote the assignment of worker *i* to job *j*. The value of $x_{ij} = 1$ if worker *i* is assigned to job *j* otherwise $x_{ij} = 0$. The cost of transporting worker *i* to job *j* is c_{ij}. Then, the objective function is as shown in Eq. (1)

$$minimize\ Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{1}$$

subject to constraints mentioned in Eqs. (2), (3) and (4)

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for i} = 1, \dots, \text{n} \tag{2}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for j} = 1, \dots, \text{n} \tag{3}$$

$$x_{ij} \in \{0, 1\} \quad \text{for j} = 1, \dots, \text{n},\ \text{i} = 1, \dots, \text{n} \tag{4}$$

The Hungarian technique is a straightforward solution approach that was developed because of the fact that all supply and demand amounts equal 1.

The two important characteristics of the method are as follows:

1. The cost matrix is a square matrix.
2. The optimum solution provides only one assignment in a row or a column of the cost matrix.

The Hungarian method works on the principle of reducing the given cost matrix to a matrix of opportunity cost by using the method of least-cost assignment. The opportunity cost shows the relative costs of allocating resources to a task as opposed to choosing the most or least efficient use of those resources.

For our targeted application, *n* drones have to be assigned from their current position to *n* waypoints, which create a particular formation in air space such that the total distance travelled by drones for each design transformation is minimum.

This algorithm was originally proposed by H. W. Kuhn is 1955 [10] and refined by J. Munkres in 1957 [11]. This algorithm is old but widely used for solving assignment problems. Researchers have studied and applied the Hungarian algorithm in Geospatial analytics [12]. Here, the algorithm is applied for optimizing the cost of assignment of delivery agents to deliver food to the customers. Various scenarios of balanced and unbalanced assignment are discussed and the results have shown that in terms of optimal allocation for equal assignment, the Hungarian algorithm gives the best possible solution. In the study conducted by a few researchers [13], the problem to optimize the scheduling of employees in a home industry engaged in bag productions for doing certain tasks is solved using the Hungarian method so that the minimum time for the bag productions will be obtained. The implementation of the Hungarian method in this application has shown that there was an increase in the production of bags by 9.09%.

The complexity of the Hungarian assignment algorithm is $O(n^3)$, where n is the total number of drones. The Alternate method proposed in literature [14] also provides the same optimal solution as the Hungarian Algorithm. Researchers [15] have also proposed an efficient GPU-based parallel algorithm for the augmenting path search, which is the most time-intensive step of the Hungarian algorithm. Further, it has been found that a minor alteration in the matrix operation can help reduce the execution time of the classical Hungarian algorithm [16]. Authors from [17] have proposed an Accelerating Hungarian Algorithm, which results in creating more than one zeroes in a single iteration that helps in the faster execution of the code. An improvement in the original Kuhn–Munkres algorithm by utilizing the sparsity structure of the cost matrix is proposed in [18]. Experimental results of two proposed algorithms sparsity-based KM (sKM) and parallel KM (pKM) have shown that sKM greatly improves computational performance whereas pKM provides a parallel way to solve assignment problems at a considerable loss. However, the code execution time is also dependent on the system configuration. Also, these optimizations are mainly required for resource-intensive applications or are useful when the number of drones are too large. For the targeted application of drone light shows, the classical Hungarian method proves sufficient and promising.

2.2. Branch and Bound algorithm

Branch and Bound (B&B) algorithm [19] is another classical method used to find optimized solutions of combinatorial optimization problems. This method [20] explores the entire search space of possible solutions and provides an optimal solution. We first build a rooted decision tree with all possible solutions. The root node represents the entire search space. Here, each child node is a partial solution and part of the solution set. Before constructing the rooted decision tree, an upper and lower bound is set for a given problem. At each level, a decision has to be made about which node to include in the solution set. Instead of following FIFO order, a live node with least cost is chosen. In this way, the best and optimal solution is found.

The cost function can be calculated using two approaches.

1. For each drone, we choose a waypoint with minimum cost i.e distance from a list of unassigned waypoints (take minimum distance from each row).
2. For each waypoint, we choose a drone with lowest cost for that waypoint from list of unassigned drones (take minimum entry from each column)

B&B has exhibited promising results for assignment problems and hence is popular among researchers.

2.3. Genetic algorithms

Assigning *n* drones to *n* waypoints to create a design can be modelled as a Quadratic Assignment Problem QAP or a Generalized Assignment Problem GAP. Genetic Algorithms (GA) – a metaheuristic algorithm is widely used for solving search and optimization problems with near-optimal solutions [21,22]. GA is based on the evolutionary ideas of natural selection and the principle of 'survival of the fittest' laid down by Charles Darwin. GAs have also been employed to find solutions to optimization problems. Unlike other algorithms, GA uses guided random search, i.e., finding the optimal solution by starting with a random initial cost function and then searching only in the space with the least cost. The task of assigning *n* drone to *n* waypoints using GA can be modelled as follows:

1. Construct N candidate vectors (chromosomes) as solutions to form an initial population. For each vector the ith index represents the ith drone and the value j represents the jth waypoint where i = 1,..,n and j = 1,..,n.
2. Calculate the fitness value of each chromosome. Here the fitness function would be the sum of the distances for each assignment of a chromosome.

3. For reproduction, two parent solutions are selected using the Roulette Wheel Selection method.
4. A crossover operator is applied to the selected parents to generate a child solution followed by a mutation procedure with a smaller probability value.
5. The fitness value of the child is calculated and then the chromosome with the fitness value lower than the child's is replaced with this child in the population.
6. The selection, crossover, mutation and replacement steps are repeated until a termination condition is fulfilled. This condition is usually based on the number of generations created without improving the best solution, or the number of fitness evaluations.

The efficiency of GA algorithm greatly depends upon the choice of genetic operators (selection, crossover and mutation) and associated parameters. However, there may be constraints on the execution time, and it is required that near-optimal solutions be found within a relatively short timeframe.

Researchers have applied GA to solve QAP, which has shown effective results in reasonable time [21]. Constructive Genetic Algorithm CGA, a modified GA technique was proposed in [22] to solve GAP. In this method, a bi-objective search is used to reduce the computations. The experimental results have shown near-optimal solutions. Another attempt [23] proposes application of GA to improve an existing heuristic method for solving Iterative Capacitated Transshipment Problem ICTP, which is a special case of GAP. The heuristic weight is tweaked by using a set of parameters suggested by GA for assigning multi-UAVs to multiple targets.

2.4. Other approaches

An Exact Algorithm is proposed in [24] for assigning tasks to multiple unmanned surface vehicles and has compared the results with Hungarian Algorithm (HA), Auction algorithm (AA), Genetic algorithm (GA) and Ant Colony Optimization (ACO). In terms of optimal solution Auction algorithm (AA) and Hungarian algorithm provide the same results, however it is shown in [25] that the computational time required by the auction algorithm to complete the assignment is exponentially increases with the increase of the number of agent, while the Hungarian shows its obvious superiority in scalability.

The assignment of drones to create a new formation from its current state can be modelled as a transportation problem TP. The goal is to find the lowest overall transportation cost of a good in order to meet the demand at destinations while utilizing resources available at origins. Algorithm based on a fuzzy linear programming approach has been proposed by researchers in [26] for solving interval-valued trapezoidal fuzzy numbers transportation problem based on comparison of interval-valued fuzzy numbers by the help of signed distance ranking. Solutions to fuzzy linear programming problems involving trapezoidal fuzzy numbers have been explored by researchers with promising results [27,28]. The shortest path problem is proposed to solve with fuzzy arc weights using an artificial bee colony (ABC) algorithm [29], which have shown acceptable results in a wireless sensor network.

There has been in-depth research done in dynamic swarm intelligence [1,3,5] and communication architecture [30–32], however, designing static swarm-intelligence with an optimal formation is less explored and developed.

Drones are aircrafts capable of completing missions with the autonomous flight capability. Even though global destination or trajectory is pre-defined there may arise a need for reactive or local path planning to tackle the uncertainties in the environment, limitations of the system structure.

A UAV can fail to accurately locate itself due to limitations in the positioning system. Once the localization errors accumulate to a certain degree, the mission might fail or the drone might get dangerously close to its neighbouring drone. The method proposed by researchers in [33] focusses on correcting the error during the flight process by using the improved genetic algorithm (GA) and A* algorithm and trajectory planning to ensure the UAV has the shortest trajectory length from the starting point to the ending point under multiple constraints and the least number of error corrections. Another uncertainty is the wind effect, which usually causes the drone to drift and tilt from its originally planned trajectory. Various effects of wind on the UAV are analysed and presented by researchers in [34]. In order to counter the effects on the position, velocities, and attitude of the UAV, the Reject External Disturbance flight mode was proposed, and modelling variable reference adaptive controller was developed. This method could prevent the UAV from path derivation without using the path controller. Our targeted application requires a dedicated and obstacle-free air space. Even so, drones need to be equipped with obstacle detection and bypassing systems for safety precautions. The researchers in [35] have proposed an innovative and efficient Circular Arc Trajectory Geometric Avoidance (CTGA) Algorithm for unmanned aerial vehicles (UAVs) which works by generating circular arc trajectories to avoid obstacles. However, the scope of this research is limited to one of the major challenges in drone light shown design, which is — the global waypoint assignment of n drones to n waypoints to create each formation in drone light show.

The classical method for assignment i.e. the Hungarian algorithm gives a complete optimal solution compared to metaheuristic methods like ACO, PSO and GA which provides a near-optimal solution in a reasonable time. For problems that require optimal solution, like the application targeted in this paper, classical approaches are preferred over metaheuristic approaches. A near-optimal assignment for such application may lead to consequences like trajectory intersection of drones, non-minimal path cost leading to an inefficient flight time usage and shorter duration drone light show. Also, a comparative study of time required for n assignments shows that the Hungarian algorithm finds an optimal solution in less time as compared to GA method.

Various classical and heuristic methods for solving assignment problems are discussed in [36]. Recent research has shown the implementation of the Hungarian algorithm for various swarm drone applications. Researchers have implemented the Hungarian algorithm for Drone-Station Matching in Smart Cities [37] and have also demonstrated the efficient vertical take-off of UAV swarms using Hungarian Algorithm [38].

GA and B&B are two most popular algorithms for assignment problems, and hence, we compare our proposed approach with implementation of these algorithms for optimal design of drone light show. The results from their experiments have been promising and reliable, which leads to the selection of the Hungarian method as our base algorithm to address the optimization problem of assigning multiple-UAVs to waypoints in order to create designs for a drone light show.

The proposed work attempts to solve the problem of optimal assignment of drones to waypoints for designing static formations for swarm drone light shows and is discussed in the next section.

3. Proposed approach

In this section, we propose an approach called **Constrained Hungarian Method for Swarm Drones Assignment (CHungSDA)** for assigning multi-UAVs to waypoints in order to create designs for a drone light show. Section 3.1 discusses the concept and the terminologies used in CHungSDA. The outline of the proposed algorithm is shown in Section 3.2 accompanied by a detailed explanation of it in Section 3.3. The stepwise numerical solution, the graphical display of assignments, the finalized optimal waypoints, and cost of assignments provide a detailed understanding of the proposed approach.

3.1. Concept and terminologies

The desired design to be created by swarm drones is converted into a mesh object and split into a number of vertices that are equal to the number of drones in the fleet. These vertices act as waypoints for drones in the fleet. We create an array $D = [[p_x^{d_1}, p_y^{d_1}, p_z^{d_1}], [p_x^{d_2}, p_y^{d_2}, p_z^{d_2}], \dots, [p_x^{d_n}, p_y^{d_n}, p_z^{d_n}]]$ where d_i represents the ith drone and $i \in [1, n]$. p_x, p_y and p_z represent the *x, y,* and *z* coordinates respectively of the ith drone. Similarly we create an array $W = [[p_x^{w_1}, p_y^{w_1}, p_z^{w_1}], [p_x^{w_2}, p_y^{w_2}, p_z^{w_2}], \dots, [p_x^{w_n}, p_y^{w_n}, p_z^{w_n}]]$ where w_j represents the jth waypoint and $j \in [1, n]$. p_x, p_y and p_z represent the *x, y* and *z* coordinates respectively of the jth waypoint. These two arrays *D* and *W* are given as input to our algorithm.

The proposed approach encompasses two constraints to ensure that the design is flawless with respect to the creation of waypoints. Constraint 1 assures if the total number of waypoints are equal to the total number of drones in the fleet — only then all the drones will be able to create the desired design.

Localization of multi-UAVs is done with the help of Global Positioning System (GPS) [39]. The positioning accuracy of each drone will vary depending on the type of GPS used. Also, various drone light show organizers will use different types of drones. The dimensions of the drones will vary as well. Considering these two scenarios, we define a parameter δ_{min}, which is the minimum distance to be maintained between any two drones to avoid collisions. Hence, Constraint 2 checks if all the waypoints in the design are unique and maintain minimum collision avoidance distance between each other. The distance between a waypoint to every other waypoint in the design should more than δ_{min}. We display the indices of the waypoints which are at a distance of less than δ_{min} from each other. This helps in manually tweaking the mesh object vertices to rectify the error. Once both the constraints are validated, the Hungarian method is applied to obtain the optimal swarm drones' assignment.

By implementing this algorithm, the assignment of drones to the waypoints will be found such that each waypoint of the design is assigned to one drone and each drone is assigned to one waypoint, such that the sum of all distances is minimum.

The Hungarian matching algorithm can be used to find minimum weight matchings in bipartite graphs. This algorithm can be represented as a bipartite graph. For this work, the bipartite graph is represented as $G = (V = (D, W); E)$ with *n* drone position vertices *(D)* and *n* waypoint vertices *(W)* and each edge *(E)* where $V = D \cup W$, $E \in DxW$, $D \cap W = \emptyset$.

c_{ij} represents non-negative weight which is the cost of assigning ith drone to jth waypoint where $1 \leq i \leq |D|$ and $1 \leq j \leq |W|$. $|D|$ and $|W|$ denote the numbers of elements in *D* and *W*, respectively. For the work stated in this paper, $|D|$ and $|W|$are equal because the same number of drones and waypoints are used to create a design. The weight represents the Euclidean distance between the current position of the ith drone and the position of the jth waypoint and is shown in Eq. (6).

Fig. 1 shows the schematic view of bipartite graphs for four drones and four waypoints of the design.

For *n* drones, the current position of the drones and waypoints of the design, which form a bipartite graph can also be represented by an adjacency matrix as shown in Table 1. In the next step, the matrix formulation of the Hungarian algorithm, a non-negative *n* × *n* matrix is considered and the elements of ith row and jth column represent the Euclidean distances between ith drone position and jth waypoint, represented as c_{ij}.

Table 1
Matrix interpretation of Hungarian algorithm for drone assignment.

	w_1	w_2	w_3	w_4
d_1	c_{11}	c_{12}	c_{13}	c_{14}
d_2	c_{21}	c_{22}	c_{23}	c_{24}
d_3	c_{31}	c_{32}	c_{33}	c_{34}
d_4	c_{41}	c_{42}	c_{43}	c_{44}

The drones are represented as d_1, d_2, d_3 and d_4 and the waypoints represented as w_1, w_2, w_3 and w_4. Table 1 shows the matrix implementation of the Hungarian algorithm for four drones and four waypoints.

The Hungarian algorithm will operate on this matrix to find the cost minimizing assignment of drones to the waypoints. Therefore, our goal is to match d_i to w_i such that each drone in D is connected to exactly one waypoint in W while minimizing total assignment cost. Here, the assignment cost is the sum of euclidean distance between each assignment for a single intermediate design transition. For achieving the same, we define a function $F(i)$ representing assignments of ith drone to jth waypoint such that the total cost is minimized as shown in Eq. (5):

$$F(i) = min \sum_{i=1}^{n} c_{d_i w_j} \tag{5}$$

3.2. Proposed algorithm

The proposed algorithm Constrained Hungarian Method for Swarm Drones Assignment (CHungSDA) is presented in Algorithm 1 followed by a stepwise detailed explanation through a numerical in Section 3.3. The proposed approach uses the Hungarian algorithm as a base assignment algorithm and further adds some key features to it corresponding to the targeted application.

Algorithm 1: CHungSDA(D, W, δ_{min})

Input: Array of current position of all the drones, D; array of waypoints to form a design, W; minimum distance required between any two waypoints, δ_{min}.
Output: An optimal matching dictionary of drone-waypoint pairs, M.
1. Let |D| be the number of drones and let |W| be the number of waypoints
2. **if** |D| = |W| **then**
3. Create a $n \times n$ Euclidean distance matrix, Mat_W, where each element represents the distance between i^th^ and j^th^ waypoints
4. **if** $\forall w_{i,j \in i \neq j} \in$ Mat_W >= δ_{min} **then**
5. Create a $n \times n$ cost distance matrix, C, where each element c_{ij} represents the Euclidean distance between i^th^ drone and j^th^ waypoint as shown in Eq. (2).

$$c_{ij} = \sqrt{(p_x^{d_i} - p_x^{w_j})^2 + (p_y^{d_i} - p_y^{w_j})^2 + (p_z^{d_i} - p_z^{w_j})^2} \quad \ldots (6)$$

6. **for** $\forall$ row(i) $\in$ C
7. row_{min} = min($C_{i:}$)
8. $\forall$ c $\in C_{i:}$, $c = row_{min} - c$ to obtain at least one zero element in each row.
9. **for** $\forall$ col(j) $\in$ C
10. col_{min}= min($C_{:j}$)
11. $\forall$ c $\in C_{:j}$, $c = col_{min} - c$ to obtain at least one zero element in each column.
12. Let l_{min} be the minimum number of lines that cover all zeros in C
13. **if** $l_{min} < n$ **then**
14. Let $c_{ij,uc}$ be an uncovered element by any line
15. Let $c_{ij,int}$ be an element at the intersection of 2 lines
16. $c_{ij,uc} = c_{ij,uc} - c_{min}$
17. $c_{ij,int} = c_{ij,int} + c_{min}$
18. Go to Step 12
19. **if** $l_{min} = n$ **then**
20. M = {d_i:w_j} $i \in R : 1 \leq i \leq n$, $j \in R : 1 \leq j \leq n$ s.t. C_{ij}=0 and it appears only once $\forall$ row(i), $\forall$ col(j)
//an optimal solution is found
21. **else**
22. Output (i, j) where $\forall w_{i,j \in i \neq j} \in$ Mat_W < δ_{min}
23. **end if**
24. **end if**

3.3. Numerical illustration of CHungSDA

The proposed algorithm CHungSDA is explained using a numerical illustration demonstrated below.

Input: Let δ_{min}= 3, D = [(5, 0, 20), (10, 0, 20), (15, 0, 20), (10, 0, 15), (10, 0, 10)], and W = [(10, 0, 25), (20, 0, 18), (16, 0, 5), (4, 0, 5), (0, 0, 18)].

(Lines 1–2) **Constraint 1:** Number of drones, |D| = 5 and number of waypoints, |W| = 5. Since |D| = |W|, the Constraint 1 is satisfied.

(Lines 3–4)/(Line 22) **Constraint 2:** Let Mat_W be as follows:

Table 2 shows a matrix which represents distances between every pair of waypoints in the design. As it can be seen that the value of all the non-diagonal elements is greater than δ_{min} i.e 3, Constraint 2 is satisfied.

(Line 5) **Creation of Cost Matrix:**

(Lines 6–8) **Row Reduction:** The minimum value of each row from Table 3 is highlighted in Table 4. This value is subtracted from each element of that row to obtain at least one zero element in each row. The resulting matrix is shown in Table 5.

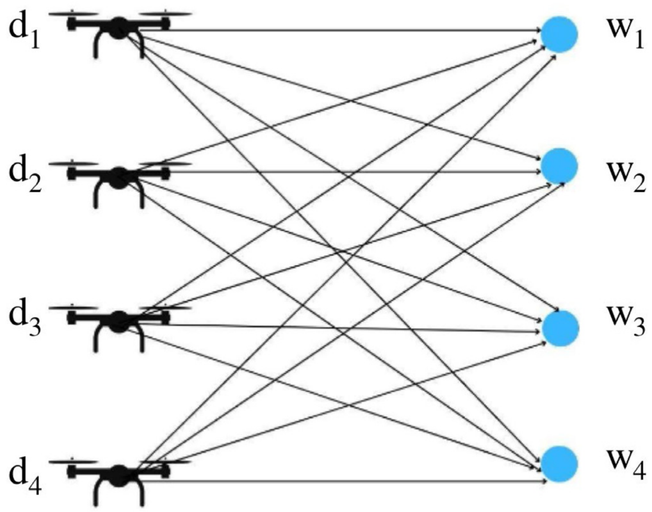

Fig. 1. Schematic View of the Bipartite Graph of 4 Drones and Waypoints.

Table 2
Waypoints distance matrix, Mat_W.

	w_1	w_2	w_3	w_4	w_5
w_1	0	12.20	20.88	20.88	12.20
w_2	12.20	0	13.60	20.61	20.0
w_3	20.88	13.60	0	12.0	20.61
w_4	20.88	20.61	12.0	0	13.60
w_5	12.20	20.0	20.61	13.60	0

Table 3
Cost matrix C.

	w_1	w_2	w_3	w_4	w_5
d_1	7.1	15.1	18.6	15.0	5.4
d_2	5.0	10.2	16.1	16.1	10.2
d_3	7.1	5.4	15.0	18.6	15.1
d_4	10.0	10.4	11.7	11.7	10.4
d_5	15.0	12.8	7.8	7.8	12.8

Table 4
Identifying minimum values from each row.

	w_1	w_2	w_3	w_4	w_5
d_1	7.1	15.1	18.6	15.0	5.4
d_2	5.0	10.2	16.1	16.1	10.2
d_3	7.1	5.4	15.0	18.6	15.1
d_4	10.0	10.4	11.7	11.7	10.4
d_5	15.0	12.8	7.8	7.8	12.8

Table 5
Resulting matrix after row reduction.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	13.2	9.6	0
d_2	0	5.2	11.1	11.1	5.2
d_3	1.7	0	9.6	13.2	9.7
d_4	0	0.4	1.7	1.7	0.4
d_5	7.2	5	0	0	5

Table 6
Identifying minimum values from each column.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	13.2	9.6	0
d_2	0	5.2	11.1	11.1	5.2
d_3	1.7	0	9.6	13.2	9.7
d_4	0	0.4	1.7	1.7	0.4
d_5	7.2	5	0	0	5

Table 7
Resulting matrix after column reduction.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	13.2	9.6	0
d_2	0	5.2	11.1	11.1	5.2
d_3	1.7	0	9.6	13.2	9.7
d_4	0	0.4	1.7	1.7	0.4
d_5	7.2	5	0	0	5

Table 8
Coverage of zeros — Iteration 1.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	13.2	9.6	0
d_2	0	5.2	11.1	11.1	5.2
d_3	1.7	0	9.6	13.2	9.7
d_4	0	0.4	1.7	1.7	0.4
d_5	7.2	5	0	0	5

Table 9
Resultant matrix after algorithmic operations — Iteration 1.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	11.5	7.9	0
d_2	0	5.2	9.4	9.4	5.2
d_3	1.7	0	7.9	11.5	9.7
d_4	0	0.4	0	0	0.4
d_5	8.9	6.7	0	0	6.7

(Lines 9–11) **Column Reduction:** The minimum value of each column is highlighted in Table 6. This value is subtracted from each element of that column to obtain at least one zero element in each column. The resulting matrix is shown in Table 7. In our example, the minimum value found in every column is 0.

(Lines 12–18) **Finding minimum number of lines required to cover all zeros:** As we can see in Table 8, all the zeros can be covered by drawing lines on column 1, column 2, column 5 and row 5. Thus l_{min}, the minimum number of lines that cover all zeros of matrix C is 4 which is less than the order of matrix i.e 5. The optimal solution is not yet possible. In that case, we find the smallest value from all the uncovered elements i.e $c_{ij,uc}$ which is 1.7. Let $c_{ij,int}$ be elements at the intersection of 2 lines which are 7.2, 5 and 5 as shown in red in Table 5. $c_{ij,uc}$ is subtracted from each uncovered element in the matrix and added to all the intersecting elements $c_{ij,int}$. The resulting matrix is shown in Table 9.

As we can see in Table 10, all the zeros can be covered by drawing lines on row 4, row 5 and column 1, column 2 and column 5. Thus l_{min} is 5 which is equal to the order of the matrix. The optimal assignment solution is possible.

(Line 19–20) **Final Assignment:** Select zero elements from the resulting matrix such that each row and column is selected only once. As shown in Table 11, the selected zero elements are highlighted in pink which will be used to assign drones to the corresponding waypoints. Index(*i,j*) of the selected zero element will represent the assignment of ith drone to the jth waypoint. The total assignment cost can be calculated by adding the element values of the original cost matrix at indices found in Table 11. The

Table 10
Coverage of zeros — Iteration 2.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	11.5	7.9	0
d_2	0	5.2	9.4	9.4	5.2
d_3	1.7	0	7.9	11.5	9.7
d_4	0	0.4	0	0	0.4
d_5	8.9	6.7	0	0	6.7

Table 11
Selection of zeros.

	w_1	w_2	w_3	w_4	w_5
d_1	1.7	9.7	11.5	7.9	0
d_2	0	5.2	9.4	9.4	5.2
d_3	1.7	0	7.9	11.5	9.7
d_4	0	0.4	0	0	0.4
d_5	8.9	6.7	0	0	6.7

Table 12
Optimal assignment of waypoints to drones by CHungSDA.

	w_1	w_2	w_3	w_4	w_5
d_1	7.1	15.1	18.6	15.0	5.4
d_2	5.0	10.2	16.1	16.1	10.2
d_3	7.1	5.4	15.0	18.6	15.1
d_4	10.0	10.4	11.7	11.7	10.4
d_5	15.0	12.8	7.8	7.8	12.8

Table 13
Summary of assignment by CHungSDA.

Drones	Waypoints	Cost
d_1	w_1	5.4
d_2	w_2	5.0
d_3	w_3	5.4
d_4	w_4	11.7
d_5	w_5	7.8
Total cost = 35.3		

individual assignment cost is highlighted in Table 12. The Optimal assignment and the cost summary is shown in Table 13. The output is stored as an optimal matching dictionary of drone-waypoint pairs, M.

The final assignment can be seen in Fig. 2. The red dots represent the current drone formation of the alphabet T and the blue dots represent the waypoints of the next formation of pentagon design. The lines joining the red dots to blue dots represent the optimal assignment of drones to waypoints to transform from T to pentagon design. Figs. 3 and 4 show drones forming T and pentagon designs respectively. This algorithm can be repeated for all the design formations in sequential order. The two arrays, the drone's current positions, and mesh object vertex locations are updated after every optimal assignment algorithm is executed.

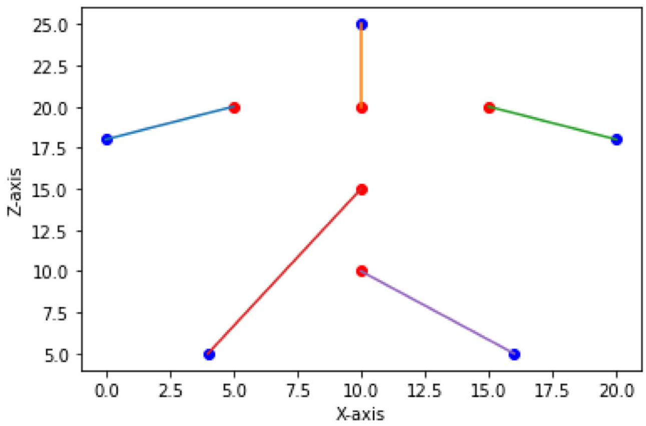

Fig. 2. CHungSDA Implementation of Presented Numerical.

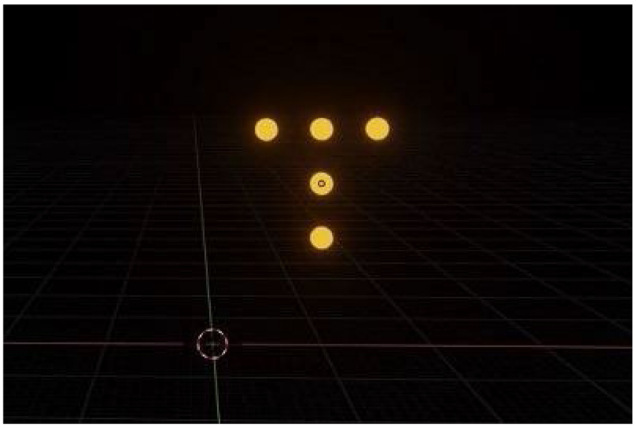

Fig. 3. Light Show Scene 1.

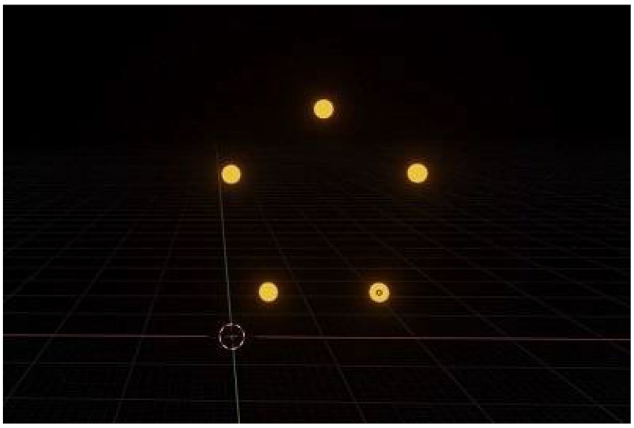

Fig. 4. Light Show Scene 2.

4. Experiments

In this section, we implement and check the assignment results of the proposed CHungSDA algorithm in a simulated environment, which is similar to the real-world scenario where drone shows are ideally held. The algorithm should be able to assign global optimal waypoints to drones during each transformation to create formations. We conducted several experiments with varying fleet sizes and

Table 14
Simulation parameters.

Algorithm	Parameters/ Key inputs	Values
Branch and bound	Base data Type of optimization	Input: Cost matrix C Minimization (of assignment cost)
Genetic algorithm	Fitness function Reproduction rate Crossover rate Mutation rate Selection method Termination condition Maximum generations Population size Initial population	Minimization (of assignment cost) 0.2 0.78 0.02 Roulette wheel selection No change in allocation for 10 consecutive generations or maximum generations reached 500 100 N candidate vectors (chromosomes). For each vector the ith index represents the ith drone and the value j represents the jth waypoint where i = 1,…,n and j = 1,…,n.
CHungSDA	Base data Initial position of drones	Input: Number of drone (n) Minimum separation between adjacent waypoints: δ_{min} • The initial positions of the drones will be on the ground, in the form of a grid, with minimum separation δ_{min} between adjacent drones. • For each intermediate transition, the position of the drones are fetched from their respective placement by the designer or by the previous assignment in the simulator, which corresponds to the local coordinate frame system of the simulator.

simple to complex design formations in a 3D coordinate system environment. The simulation environment, parameters initialization are discussed in Section 4.1. The experimental results and discussion are given in Section 4.2.

4.1. Simulation parameters

The proposed approach is implemented in Python programming language and all the simulations were performed in Blender software using the local coordinate frame system on Windows 10 Home, Intel(R) Core(TM) i5-8300H CPU @ 2.30 GHz, GeForce GTX 1050 2 GB, 16 GB RAM Laptop. The initialization parameters and values are tabulated in Table 14.

4.2. Results and discussion

Drone assignments to create formations have been tested with a varying number of drones with GA, Branch and Bound algorithm and our proposed CHungSDA algorithm. The total distance travelled by drones as per the assigned waypoints per transition is tabulated in Table 15. It can be observed from the results that the most optimal solution i.e. the lowest cost in terms of distance travelled is obtained from our proposed approach.

It can be observed from the results that the most optimal solution i.e. the lowest cost in terms of distance travelled is obtained from the proposed approach.GA provides a near optimal solution and the cost is comparatively higher than the other two methods. B&B method also provides a near optimal solution and the cost obtained is better i.e. lower than GA but it is still higher than the proposed method.

The running time complexity of Hungarian method is $O(n^3)$, where n is the total number of drones. CHungSDA is a variant of the Hungarian method, but it is assured that the time complexity is not increased due to added constraints in the proposed approach. Although the execution time increases as the number of drones in the fleet increases, maintaining the time complexity same as the baseline method is acceptable because the light show designs are finalized well in advance to the actual show time. Further, the storage requirements per design are for the $n \times n$ matrix only. That is, the algorithm running time would not affect the light show performance during the implementation of the application.

Table 15
Total distance travelled by drones per transformation in meters.

Design	No. of drones	GA	Branch and bound	Proposed approach - CHungSDA
1	5	42	37	35
2	10	81	72	67
3	20	106	98	92
4	50	219	204	204
5	100	492	423	400
6	500	2136	2015	1996
7	1000	4237	3915	3915

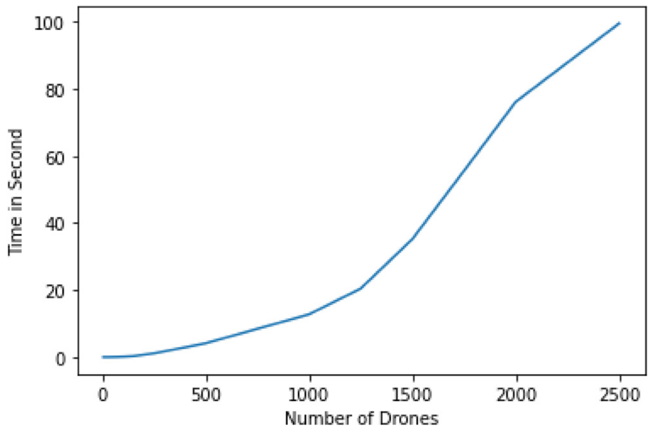

Fig. 5. Execution Time of CHungSDA for n Drones.

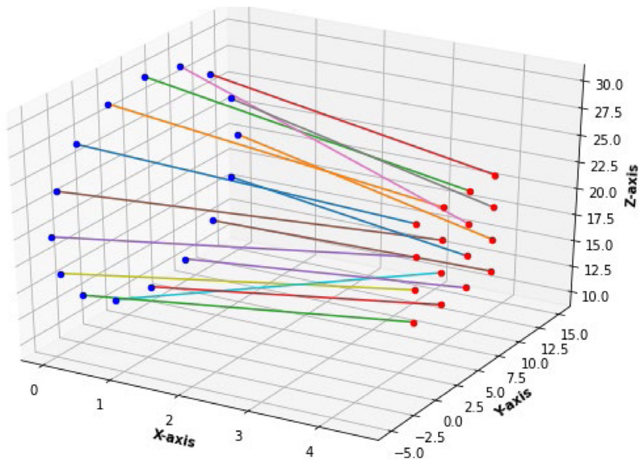

Fig. 6. CHungSDA Example Implementation 1.

Fig. 5 shows the average execution time for optimal assignment by CHungSDA for a varied number of drones and different designs (simple, average, complex). However, for very complex designs, the worst case time complexity remains cubic.

Figs. 6 to 10 represent the implementation of CHungSDA algorithm and show the optimal assignment of the waypoints to the drones in the fleet to create the desired design. In these figures, the red dots represent the current position of the drones and blue dots represent the waypoints (vertices) for the design creation.

Fig. 6 shows the optimal assignment of 16 drones to form a circle. In Fig. 7, 16 drones currently form a heart design and the optimal assignment is shown by lines to form an infinite symbol. Fig. 8 represents 3D perspective and Fig. 9 represents 2D perspective of an optimal assignment of 16 drones from a circle to the heart design. In Fig. 10, the red dots represent the current formation of a moon design and the blue dots represent the waypoints of the next formation of star design. The lines joining the red dots to blue dots represent the optimal assignment of drones to waypoints to transform from moon to star design. Figs. 11 and 12 show drones forming moon and star designs respectively.

Simulations of simpler designs (like Figs. 6 to 12) are dedicatedly shown for the reader's easy understanding. Simulations have also been performed on more complex designs (like Fig. 15).

Figs. 13–15 show a simulation of 3 different designs created by a fleet of 240 drones. The drones were optimally assigned to the waypoints of the design by using the proposed CHungSDA algorithm. Positive and negative tests were also conducted on the waypoints of the design to successfully pass the pre-conditional checklist. The simulation video of this drone light show is available at CHungSDASimulation.

Several simulations of CHungSDA conducted on a varying number of drones and the formation of different designs prove the efficacy of the proposed approach.

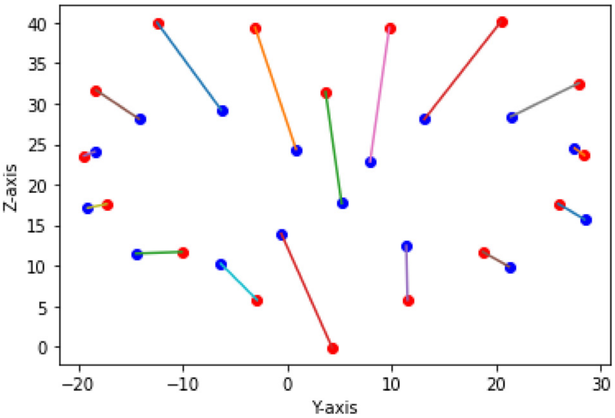

Fig. 7. CHungSDA Example Implementation 2.

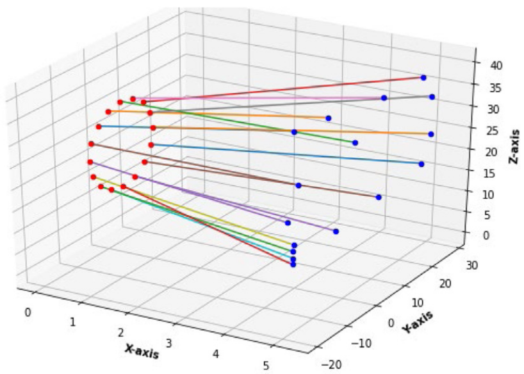

Fig. 8. CHungSDA Example Implementation 3 (3D Perspective).

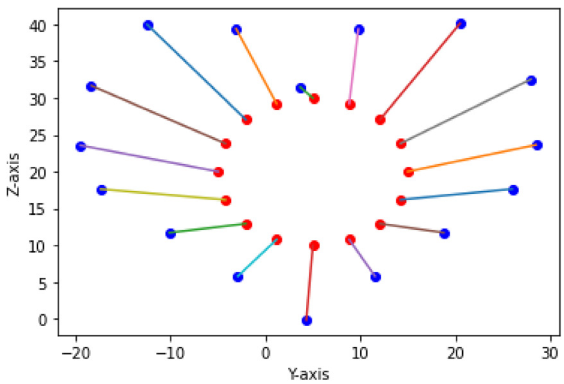

Fig. 9. CHungSDA Example Implementation 3 (2D Perspective).

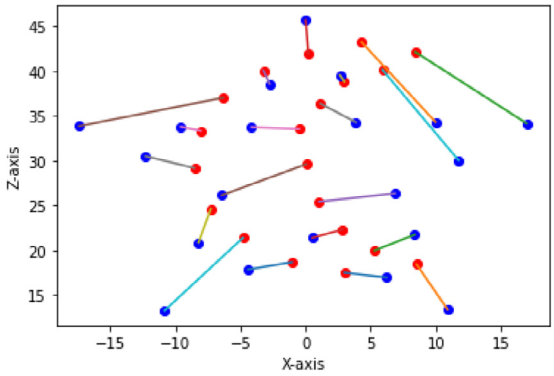

Fig. 10. CHungSDA Example Implementation 4.

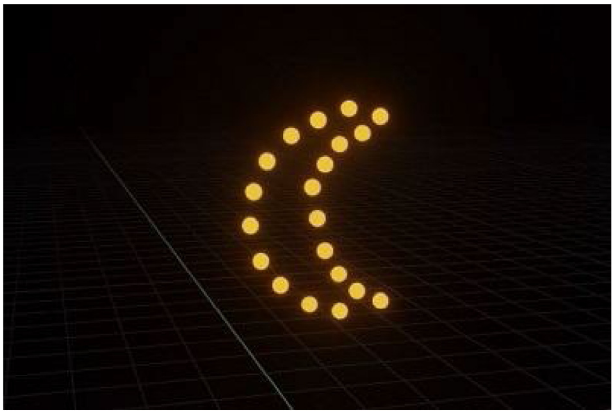

Fig. 11. CHungSDA Example Implementation 4 - Scene 1.

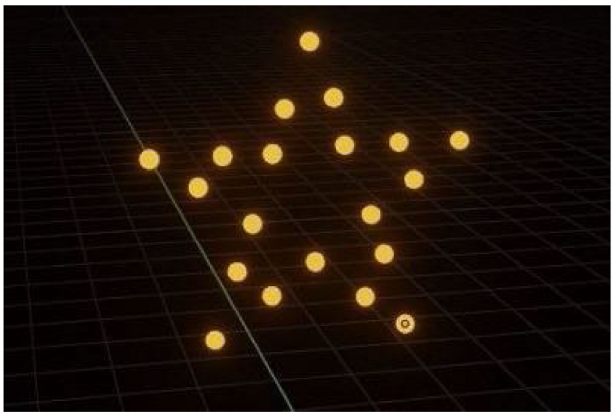

Fig. 12. CHungSDA ExampleImplementation 4 - Scene 2.

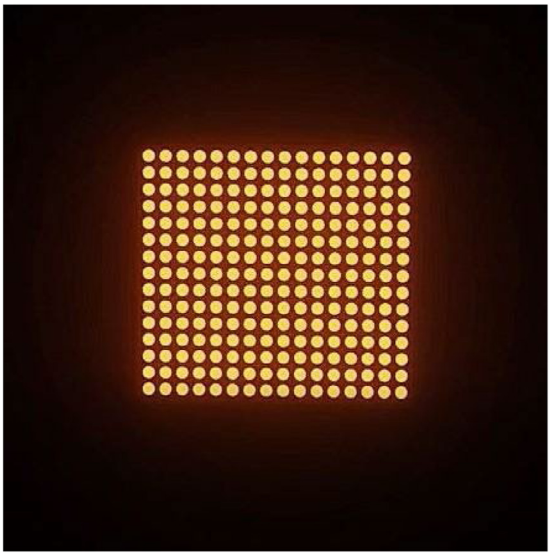

Fig. 13. CHungSDA Example Implementation 5 - Scene 1.

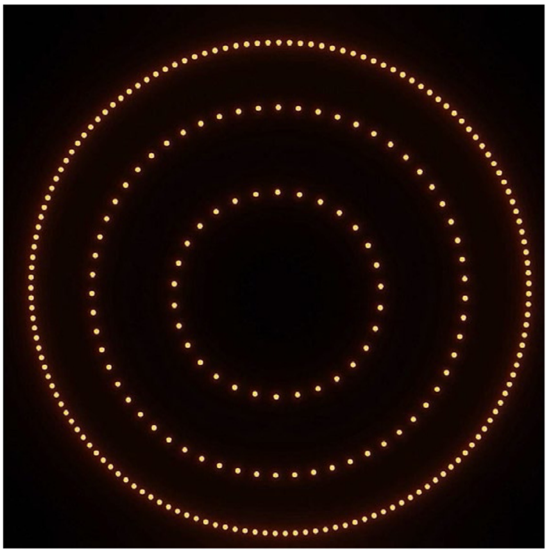

Fig. 14. CHungSDA Example Implementation 5 - Scene 2.

Fig. 15. CHungSDA Example Implementation 5 - Scene 3.

5. Conclusion and future scope

Light show designs are a spectacular application of swarm drones. In order to design a light show with multi-UAVs, it is required to assign waypoints to UAVs in different formations. The implementation results show that the proposed approach Constrained Hungarian Method for Swarm Drones Assignment (CHungSDA) provides an optimal global assignment solution that assigns drones to the vertices of the design for each swarm drones design transformation. This method reduces the overall path cost for the assignment while ensuring that the trajectory of the drones does not intersect. CHungSDA can be executed multiple times for a series of design formations leading to the generation of a drone light show. The output of this algorithm generates multiple waypoints in a 3D cartesian coordinate system for each individual drone in the fleet. The generated waypoints can be saved as a mission file for a show. Drone light shows can be organized indoors as well as in outdoor environments. Different localization methods would be required to position the drone in the given environment. The Ground Control Station used for deploying multiple UAVs for a task will import the mission file produced using CHungSDA and transform the waypoints into the required localization format. The concepts and results from this paper will provide insights in the development of Ground Control Stations (GCS) for deploying static missions to fleets of drones to perform a given task in a coordinated manner.

CHungSDA is implemented for global assignment in a simulated environment similar to the real-world scenario of open ground and obstacle-free air space where drone light shows are organized. It considers that the trajectory of drones is linear along the shortest distance from the allocated waypoint to the drone's current position. However, sometimes due to unavoidable environmental factors like high wind speed, drones might drift from their originally allocated position or trajectory, which might lead to close proximity or collision with neighbouring drones. In such cases, a system capable of reactive path planning to sense and avoid collision would be necessary. The future scope of this work could be to enhance implementation of the CHungSDA algorithm with AI-based real-time obstacle detection and reactive trajectory planning. The AI-based system would bypass the obstacles, help in reaching the globally assigned waypoints by CHungSDA and enhance the overall efficiency and robustness of the system. The optimal assignment problem can be further explored and adapted using the fuzzy assignment solutions for the problem targeted.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Spanogianopoulos S, Zhang Q, S. Spurgeon. Fast formation of swarm of UAVs in congested urban environment. IFAC-PapersOnLine 2017;50(1):8031–6.
[2] Lanteigne A, Kibru E, Azam S, Shammary S. Design of a drone light-show production system.
[3] Kuru K, Ansell D, Khan W, H. Yetgin. Analysis and optimization of unmanned aerial vehicle swarms in logistics: An intelligent delivery platform. IEEE Access 2019;7:15804–31.
[4] Abdelkader M, Güler S, Jaleel H, J. Shamma. Aerial swarms: Recent applications and challenges. Curr Robot Rep 2021;2(3):309–20.
[5] Alkouz B, A. Bouguettaya. Formation-based selection of drone swarm services. In: MobiQuitous 2020-17th EAI international conference on mobile and ubiquitous systems: computing, networking and services. 2020, p. 386–94.
[6] Tahir A, Böling J, Haghbayan M, Toivonen H, J. Plosila. Swarms of unmanned aerial vehicles — A survey. J Ind Inform Integr 2019;16:100–6.

[7] Bandala A, Dadios E, Vicerra R, L. Lim. Swarming algorithm for unmanned aerial vehicle (UAV) quadrotors - swarm behavior for aggregation, foraging, formation, and tracking. J Adv Comput Intell Intell Inform 2014;18:745–51.
[8] Belkadi A, Ciarletta L, D. Theilliol. Particle swarm optimization method for the control of a fleet of unmanned aerial vehicles. J Phys Conf Ser 2015;659(1):012–5.
[9] Chowdhury S, Marufuzzaman M, Tunc H, Bian L, W. Bullington. A modified ant colony optimization algorithm to solve a dynamic traveling salesman problem : A case study with drones for wildlife surveillance. J Comput Des Eng 2019;6(3):368–86.
[10] Kuhn HW. The Hungarian method for the assignment problem. Nav Res Logist Q 1955;2:83–97.
[11] Munkres J. Algorithms for the assignment and transportation problems. J Soc Ind Appl Math 1957;10:196–210.
[12] Ghosh P, Kulkarni S, Rajan D, Sanfransisco J. Team Geospoc. Application of Hungarian algorithm in geospatial analytics : A cost optimization approach.
[13] Rusdiana S, Oktavia R, E. Charlie. Application of Hungarian method in optimizing the scheduling of employee assignment and profit of home industry production. J Res Math Trends Technol 2019;1(1):24–33.
[14] Akpan N, U. Abraham. A critique of the Hungarian method of solving assignment problem to the alternate method of assignment problem by mansi. Int J Sci: Basic Appl Res 2016;29:43–56.
[15] Date K, R. Nagi. GPU-accelerated Hungarian algorithms for the linear assignment problem. Parallel Comput 2016;57:52–72.
[16] Wright M. Speeding up the Hungarian algorithm. Comput Oper Res 1990;17:95–6.
[17] Munapo E. Development of an accelerating Hungarian method for assignment problems. Eastern-Eur J Enterprise Technol 2020;4(4106):6–13.
[18] Cui H, Zhang J, Cu C, Q. Chen. Solving large-scale assignment problems by Kuhn-Munkres algorithm. Ameii 2016;822–7.
[19] Kianfar K. Branch-and-bound algorithms. In: Wiley encyclopedia of operations research and management science. 2011.
[20] Morrison D, Jacobson S, Sauppe J, Sewell E. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optim 2016;19:79–102.
[21] Azarbonyad H, Babazadeh R. A genetic algorithm for solving quadratic assignment problem(QAP). 2014, ArXiv, abs/1405.5050.
[22] Lorena L, Narciso M, Beasley J. A constructive genetic algorithm for the generalized assignment problem. Evol Optim 2000.
[23] Juell P, Perera A, K. Nygard. Application of a genetic algorithm to improve an existing solution for the generalized assignment problem. In: Proceedings of the 16th international conference on computer applications in industry and engineering, Vol. 216. 2003.
[24] Xue K, Huang Z, Wang P, Z. Xu. An exact algorithm for task allocation of multiple unmanned surface vehicles with minimum task time. J Mar Sci Eng 2021;9(907).
[25] Ismail S, L. Sun. Decentralized Hungarian-based approach for fast and scalable task allocation. In: International conference on unmanned aircraft systems, Vol. 2. 2017, p. 3–28.
[26] A. Ebrahimnejad. Fuzzy linear programming approach for solving transportation problems with interval-valued trapezoidal fuzzy numbers. Sādhanā 2016;41:299–316.
[27] Ebrahimnejad A, Nasseri H, Lotfi F. Bounded linear programs with trapezoidal fuzzy numbers. Int J Uncertain Fuzziness Knowl-Based Syst 2010;18:269–86.
[28] Ebrahimnejad A, Nasseri H. Linear programmes with trapezoidal fuzzy numbers: A duality approach. Int J Oper Res 2012;13:67–89.
[29] Ebrahimnejad A, Tavana M, Alrezaamiri H. A novel artificial bee colony algorithm for shortest path problems with fuzzy arc weights. Measurement 2016;93:48–56.
[30] Zhu X, Bian C, Chen Y, Chen S. A low latency clustering method for large-scale drone swarms. IEEE Access 2019;7:186260-186267.
[31] Chen X, Tang J, Lao S. Review of unmanned aerial vehicle swarm communication architectures and routing protocols. Appl Sci 2020;10(3661).
[32] Campion M, Ranganathan P, Faruque S. A review and future directions of UAV swarm communication architectures. In: IEEE international conference on electro information technology. 2018, p. 903–8.
[33] Zhou H, Xiong H, Liu Y, Tan N, Chen L. Trajectory planning algorithm of UAV based on system positioning accuracy constraints. Electronics 2020;9(2):250.
[34] Bohang W, Zain A, Daobo W, Paolo M. Controller for UAV to oppose different kinds of wind in the environment. J Control Sci Eng 2020;1–10.
[35] Guo J, Liang C, Wang K, Sang B, Wu Y. Three-dimensional autonomous obstacle avoidance algorithm for UAV based on circular arc trajectory. Int J Aerosp Eng 2021;21:3–226.
[36] Güneri Ö, Durmuş B, D. Aydın. Different approaches to solution of the assignment problem using R program. J Math Stat Sci 2019;5(2411–2518):129–45.
[37] Mirzaeinia A, Bradley S, M. Hassanalian. Drone-station matching in smart cities through Hungarian algorithm: Power minimization and management. In: AIAA propulsion and energy 2019 forum. 2019.
[38] Hernández D, Cecilia J, Calafate C, Cano J, P. Manzoni. The kuhn-munkres algorithm for efficient vertical takeoff of UAV swarms. In: 2021 IEEE 93rd vehicular technology conference. 2021, p. 1–5.
[39] Patrik A, Utama G, Agung A, Gunawan S, Chowanda A, S. Suroso J. GNSS - Based navigation systems of autonomous drone for delivering items. J Big Data 2019;6:1–14.