Task assignment and path planning of multiple unmanned aerial vehicles using Integer Linear Programming

Mr.Imran Saeed Mirza

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering
Department of Electrical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2023

การมอบหมายงานและการวางแผนเส้นทางของยานพาหนะทางอากาศไร้คนขับหลายคันโดยใช้โปรแกรมเชิงเส้นจำนวนเต็ม

นายอิมรัน ซาอิด เมียร์ซ่า

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิศวกรรมไฟฟ้า ภาควิชาภาควิชาวิศวกรรมไฟฟ้า

คณะคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2566

Thesis Title	Task assignment and path planning of multiple unmanned aerial vehicles using Integer Linear Programming
By	Mr.Imran Saeed Mirza
Field of Study	Electrical Engineering
Thesis Advisor	Prof. Dr.LUNCHAKORN WUTTISITTIKULKIJ
Thesis Co Advisor	

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Engineering

-- Dean of the FACULTY OF ENGINEERING

(Prof. Dr.SUPOT TEACHAVORASINSKUN)

Thesis COMMITTEE

-- Chairperson

(Asst. Prof. Dr.SUREE PUMRIN)

-- Advisor

Prof. Dr.LUNCHAKORN WUTTISITTIKULKIJ

-- Committee

Assoc. Prof. Dr.CHARNCHAI PLUEMPITIWIRIYAWEJ

-- External examiner

Dr.Pisit Vanichchanunt

คำสำคัญ: **การมอบหมายงาน, การวางแผนเส้นทาง หลาย UAV; อัลกอริทึม ILP**

นายอิมรัน ซาอิด เมียร์ซ่า :
การมอบหมายงานและการวางแผนเส้นทางของยานพาหนะทางอากาศไร้คนขับหลายคันโดยใช้โปรแกรมเชิงเส้นจำนวนเต็ม (Task assignment and path planning of multiple unmanned aerial vehicles using Integer Linear Programming) อาจารย์ที่ปรึกษาหลัก: ลัญฉกร วุฒิสิทธิกุลกิจ

วิทยานิพนธ์ฉบับนี้นำเสนอสูตรโปรแกรมเชิงเส้นจำนวนเต็มเพื่อการมอบหมายงานและการวางแผนเส้นทางให้กับอากาศยานไร้คนขับหลายลำกลุ่มโดรนจะได้รับมอบหมายให้เยี่ยมภารกิจให้ครบทุกเป้าหมายและเดินทางกลับไปยังตำแหน่งเริ่มต้นโดยมีเป้าหมายให้ได้ผลรวมของระยะทางที่โดรนทุกตัวเดินทางมีค่าน้อยที่สุดในขณะเดียวกันไม่อนุญาตให้มีทัวร์ย่อยใดๆเพื่อให้ได้ผลลัพธ์ที่มีประสิทธิภาพมากที่สุดมีการทดลองทดสอบเพื่อตรวจสอบความถูกต้องของสูตรที่ได้นำเสนอโดยใช้จำนวนโดรนและจำนวนงานที่หลากหลายผลการทดสอบแสดงให้เห็นว่าการเพิ่มจำนวนโดรนโดยทั่วไปสามารถช่วยลดระยะทางรวมและลดระยะเวลาในการทำภารกิจให้เสร็จสิ้นได้ข้อดีของวิธีที่เสนอคือให้ผลเฉลยที่เหมาะสุดแต่ใช้เวลาในการคำนวณเพื่อค้นหาผลเฉลยก็นานด้วยดังนั้นจึงได้ทำการทดลองโดยกำหนดจำนวนโดรนและภาระงานที่ขนาดแตกต่างกันจากการทดสอบโดยใช้คอมพิวเตอร์โน้ตบุ๊กที่มีหน่วยประมวลผลเพนเทียม10หน่วยความจำแรม8GBและหน่วยความจำเอสเอสดีขนาด256GBพบว่าระบบที่มีงานไม่เกิน27การหาผลเฉลยเหมาะสมที่สุดทำได้ภายในไม่กี่นาทีโดยไม่ขึ้นกับจำนวนโดรนที่ใช้งานแต่สำหรับงานจำนวนมากที่เกินขีดจำกัดนี้จะไม่สามารถได้ผลเฉลยแม้ว่าจะใช้เวลาจำลองไปแล้วเกินหนึ่งชั่วโมงก็ตาม

สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต	....
ปีการศึกษา	2566	ลายมือชื่อ อ.ที่ปรึกษาหลัก	....
		ลายมือชื่อ อ.ที่ปรึกษาร่วม	....

6470313521 : MAJOR ELECTRICAL ENGINEERING
KEYWORDS: Task assignment, path planning; Multi-UAV; ILP Algorithm

Mr.Imran Saeed Mirza : Task assignment and path planning of multiple unmanned aerial vehicles using Integer Linear Programming Advisor: Prof. Dr.LUNCHAKORN WUTTISITTIKULKIJ

In this thesis, we propose an Integer Linear Programming (ILP) formulation to perform task assignment and path planning in applications of distributed multiple unmanned aerial vehicle (multi-UAV) environment in which a set of drones are assigned to ensure that each task is visited once while all drones return to its initial location. The objective is to minimize the sum of distances traversed by all drones, while opting out the possibilities of any sub tours, thereby achieving optimal solutions. Several experiments with various different number tasks and drones are carried out to verify the correctness of our proposed formulation. Numerical results show that introducing more drones for numerous tasks can typically reduce the total distance and shorten the mission completion time. Despite it optimality advantage, the computational time requirement to find the solution is the main concern. Therefore, we conducted extensive simulation experiments to determine its limit with respect to the number of drones and tasks. Based on out tests using a computer notebook with Pentium 10, 8 GB RAM and 256 GB SSD, the algorithm will find optimal solutions under few minutes for systems with no more than 27 tasks, regardless of the number of drones being deployed. For, larger number of tasks beyond this limit, solutions are not available even after one hour of simulation time.

Field of Study	Electrical Engineering	Student's Signature ....
Academic Year	2023	Advisor's Signature ....
		Co-advisor's Signature ....

ACKNOWLEDGEMENTS

I'd like to begin by expressing my gratitude to my advisor, Assoc. Prof. Dr. Lunchakorn Wuttisittikulkij, for providing me with an opportunity to study at Chulalongkorn University and for his invaluable counsel and direction. I am appreciative of Asian Scholarship Program's financial assistance in allowing me to complete my master's degree in a fascinating area of engineering. I certainly appreciate the opinions and points raised by the committee members throughout my thesis review.

I want to thank all the seniors and friends in my research group for being so helpful and nice to me when I was studying. I also want to express my gratitude to Dr. Shashi Shah, my senior, for his suggestions and assistance with my thesis. Finally, my loving parents and siblings who have supported and encouraged me during my lengthy journey through school life make me feel very blessed and thankful.

Imran Saeed Mirza

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS (if available)

iThesis	ไอทีสิส
CAA	Conventional auction algorithm.
CBBA	Consensus based bundle algorithm.
DTAP	Dynamic task allocation algorithm.
DRL	Deep Reinforecment learning.
ILP	Integer Linear Programming.
MPS	Mission planning system.
MSOS	Modified symbolic organisms' search.
MUTAPP	Multi-agent target assignment and path planning
MADDPG	Multi-agent deep deterministic policy gradient.
MAPD	Multi-agent pick and drop.
STAPP	Simultaneous target assignment and path planning.
SGWO	Simplified grey wolf optimizer.
SPSO	Spherical vector-based particle swarm optimization.
UAV	Unmanned aerial vehicle.
WPC	Wireless powered communication.

LIST OF Tables

Page

CHAPTER 1

Preface

1.1. Introduction

Task assignment and path planning are critical aspects of coordinating multiple Unmanned Aerial Vehicles (UAVs) in various applications, ranging from surveillance and reconnaissance to disaster response and package delivery. The efficient allocation of tasks and optimal path planning for UAVs can significantly enhance their performance, reduce mission completion time, and improve overall mission success rates. In recent years, researchers have focused on developing intelligent algorithms and optimization techniques to address these challenges. One such approach is the utilization of Integer Linear Programming (ILP) for task assignment and path planning of multiple UAVs.

The ILP framework provides a mathematical modeling technique to formulate complex optimization problems involving discrete decision variables, objective functions, and a set of constraints. By representing the task assignment and path planning problem as an ILP model, it becomes possible to find an optimal solution considering various factors such as resource constraints, number of tasks and distance. The ILP approach offers the advantage of rigorous mathematical optimization, allowing for systematic exploration of the solution space.

The task assignment problem involves determining which tasks should be assigned to each UAV, considering factors such as task importance, UAV capabilities, and mission requirements. Path planning, on the other hand, focuses on finding optimal routes or trajectories for the UAVs to navigate from their current locations to the assigned tasks. The path planning problem is challenging due to several factors, including limited UAV resources, distance, and potential conflicts with other UAVs or obstacles. ILP-based path planning models can incorporate these constraints and generate collision-free paths that optimize criteria such as travel distance, mission completion time, or energy consumption.

The integration of task assignment and path planning within an ILP framework enables a comprehensive optimization approach for multiple UAVs. By jointly considering task assignment and path planning, the ILP model can account for dependencies and interactions between tasks and paths, leading to more efficient and coordinated UAV operations.

Several research studies have investigated the application of algorithms in the context of task assignment and path planning for multiple UAVs. In (2017) [1] the realm of multi-AUV task allocation, an inventive auction algorithm is employed for hunting task assignments, featuring advancements in auctioneer selection and bidding value calculation to enhance security and robustness, yielding superior outcomes compared to traditional and alternative allocation methods. In (2020) [2] presented paper addresses multi-robot task allocation and path planning in a two-dimensional warehouse setting, utilizing a market auction-based allocation algorithm and an enhanced Astar algorithm considering task costs and inter-task distances to optimize total distance and runtime; simulation demonstrates effective task execution and collision-free paths. Yang et al. (2020) [3] developed an ILP-based approach for multi-UAV surveillance task assignment, incorporating multiple objectives and constraints.

Moreover, the use of ILP in UAV task assignment and path planning has been extended to specific domains and applications. For instance, Chen et al. (2018) [4] focused on the allocation of sensing tasks to UAVs in a target tracking scenario, considering both sensing and communication constraints. Guo et al. (2020) [5] proposed an ILP-based approach for task assignment and path planning of UAVs in precision agriculture applications, optimizing crop coverage and minimizing travel distance. These studies demonstrate the versatility and effectiveness of ILP in addressing task assignment and path planning challenges in various UAV applications.

In summary, task assignment and path planning of multiple UAVs play a crucial role in optimizing mission performance and resource utilization. The distance factor and the surety of completion of task without any repetition or making loops have been done in this research. The following sections of this paper will delve into the details of ILP-based approaches, algorithms, and case studies related to task assignment and path planning for multiple UAVs. This ILP-based drone routing and task assignment solution offers several notable advantages. It exhibits exceptional flexibility and adaptability, accommodating diverse scenarios and parameters, making it suitable for various applications. The inclusion of visualizations using Matplotlib enhances the code's accessibility and facilitates a clear understanding of optimized drone routes. It is well-structured and easy to modify, making it accessible to a wide audience. By leveraging optimization techniques through ILP formulation, the code efficiently minimizes the total distance traveled. It further stands out for its capability to generate random scenarios, allowing for simulations. Additionally, this code is forward-thinking, with potential for future enhancements and addresses practical challenges through the Miller-Tucker-Zemlin (MTZ) method for subtour elimination. The code's effectiveness depends on specific problem instances and constraints, but it serves as a versatile tool for addressing a broad range of drone routing and task assignment challenges.

1.2. Problem Statement

The efficient assignment of multiple tasks to UAVs is crucial for optimizing mission performance and resource utilization. Existing research has explored various approaches for UAV task assignment, path planning and minimum distance. However, there is a need for a comprehensive solution that considers the global mission planning, computational complexity, path planning, and resource allocation aspects. This research aims to develop an architecture utilizing ILP to enable mission planning, agent-to-task assignment, and optimal utilization of UAVs in complex environments. The proposed algorithm will address the challenges associated with multi-task assignment for UAVs and provide an optimal solution for improved mission execution and resource management.

A key aspect of this research is to identify how large the problem can be resolved by the proposed approach to achieve the optimal solution.

1.3. Research Objective

This thesis aims to develop and implement an integer linear programming (ILP) formulation for optimal task assignment and path planning of multiple UAVs. The algorithm will simultaneously address task assignment and path planning, resulting in global optimal solution efficient execution of multiple tasks. The proposed technique will incorporate an objective function that considers task assignment performance, and a constraint penalty term will ensure a balanced workload distribution among UAVs while optimizing resource utilization. The results demonstrate the proposed algorithm in multi-task assignment and path planning for UAVs. The thesis will provide a detailed methodology, simulations, and presentation of results to enhance understanding of path planning and task allocation. The computational time requirement of our approach will be systematically evaluated with various problem sizes so that the limit of the problem sizes can be identified.

1.4. Significance of Research

This research focuses on the significance of developing an ILP-based technique for efficient task allocation, resource management, and path planning in UAVs. These versatile aerial vehicles are increasingly utilized in various domains, such as deliveries, warfare, and passenger transportation. However, effectively managing their multitasking capabilities poses a complex challenge. By leveraging ILP, the research aims to optimize task allocation and path planning, thereby enhancing operational efficiency and effectiveness. Computational requirements are the key aspect of this research. Such that we will be able to determine the largest size of the problems one can solve with a reasonable time, which is suitable for practical use.

CHAPTER 2

Survey

2.1. Literature Review and Background

Historically the use of UAVs was first reported in 1849, serving as balloon carrier to initiate wars on Austrian forces. In early ages drones were only used as Warcraft. With the passage of time much research has been conducted and many other tasks are also achieved by using this technology. In the present era, this technology has been used to deliver packages, for transportation of goods and as well as passengers, and in space missions also to provide necessary stuff to space station without any human intervention.

In addressing the complex challenge of multi-UAV target assignment and path planning (MUTAPP) [6], this study introduces a novel approach, termed Simultaneous Target Assignment and Path Planning (STAPP). By leveraging the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm, STAPP transforms the MUTAPP problem into a trainable multi-agent system. This enables the real-time resolution of target assignment and path planning in dynamic UAV environments, facilitated by a streamlined neural network and further enhanced through innovative training techniques. Experimental validation underscores the efficacy of the proposed STAPP method. In order to cluster the search space, one method compares the distance traveled by each UAV, and the other [7] employs a cost function that approximates the traveled distance.

Although several collision avoidance approaches have been reported, there is a lack of highlighting the key components shared by these approaches. In this subject of [8] to provide researchers with a state-of-the-art overview of various approaches for multi-UAV collision avoidance. [9] Addressing the intricate Multi-agent Pickup and Delivery (MAPD) challenge prevalent in automated warehouse logistics and mail sortation, this study introduces an innovative coupled approach. By integrating a marginal-cost assignment heuristic and a meta-heuristic enhancement strategy based on Large Neighborhood Search, task assignments are guided by actual delivery costs, thereby outperforming conventional lower-bound estimate methods. The method's efficacy is substantiated through numerical simulations, demonstrating enhanced efficiency and substantial improvements over recent approaches, even extending to scenarios involving multiple tasks per robot. To accomplish this task, a novel system framework is designed and proposed to accomplish simultaneous moving target tracking and path planning by a quadrotor UAV with an onboard embedded computer, vision sensors, and a two-dimensional laser scanner [10]. The area between the UAV group range and the group communication range is called the insecurity range and, in the region, multi-UAV communication can cause serious information leakage. To resolve this problem [11] consider two aspects, namely, cooperative control and secure communication. UAV path planning problem is an important component of UAV mission planning system, which needs to obtain optimal route in the complicated field. To solve this problem, a novel hybrid algorithm called HSGWO-MSOS is proposed by combining simplified grey wolf optimizer (SGWO) and modified symbiotic organisms search (MSOS) [12]. [13] propose a Deep Reinforcement Learning (DRL) approach for UAV path planning based on the global situation information. UAV path planning is modelled as the optimization problem, in which fitness functions include travelling distance and risk of UAV, three constraints involve the height of UAV, angle of UAV, and limited UAV slope [14]. [13] study the trajectory and resource allocation design for downlink energy-efficient secure UAV communication systems, where an information UAV assisted by a multi-antenna jammer UAV serves multiple ground users in the existence of multiple ground eavesdroppers. The UAV provides services to IoT terminals as an aerial BS based on the wireless-powered communication (WPC) technique. According to this system a synergetic scheme for UAV trajectory planning and subplot allocation. [15] present a new algorithm named spherical vector-based particle swarm optimization (SPSO) to deal with the problem of path planning for UAVs in complicated environments subjected to multiple threats.

While talking about unmanned vehicles. There are two of the main categories that are comprises as unmanned vehicles one of them is ground unmanned vehicles and other is aerial unmanned vehicles. UAVs are those which work in the concept of auto pilot system. This system is evolving in the modern era with the passage of time. In this thesis the focus is the problems faced due to aerial unmanned vehicles system and their solution both mathematically and in terms of engineering. UAVs are mostly drones, modern aircraft without any human involvement (both as a pilot or a guide). Before going further, it is necessary to learn which kind of engineering comes under the flag of Unmanned aerial vehicles. A person who wants to work in this prime field i.e., field of UAVs must understand controller modifications, control systems and communication systems associated with UAVs. Before going into technical details, first we learn about the basic terminologies. There is a lot of confusion going on while using the words drone and UAVs. While watching the news, the word drone is used many times or in simple words drone is the military term used primarily by armed forces. UAV is the same thing, this word is most common in technological field i.e., the field where people are working or researching in this field. In short both these words have the same meanings.

2.1.1. Conventional Auction Algorithm [17]

The Conventional Auction Algorithm (CAA) is an iterative combinatorial optimization algorithm used for solving a variety of problems, including resource allocation, task scheduling, and network routing. The algorithm is based on a bidding mechanism where each agent in the system makes bids for the resources or tasks it needs to complete its objectives.

The CAA proceeds through a series of rounds, where in each round, agents submit bids for the resources they need. The bids are then evaluated, and the agents are allocated the resources based on their bids. The algorithm uses a pricing mechanism to determine the value of the resources in each round, and agents are incentivized to bid truthfully by paying the amount they bid for the resource.

The CAA is a decentralized algorithm that does not require a central authority to coordinate the resource allocation. It is known to be efficient and can converge to a stable allocation in a finite number of rounds. However, the algorithm does not guarantee an optimal allocation and can suffer from issues such as bid shading and collusion among agents.

In the figure below, it is assumed that possible D2D users are spread in PPP form, and the path-loss impact and log-normal shadowing propagation possibilities between the likely D2D transmitter and receivers in a multicast group are considered. A multicast D2D transmitter in a multicast D2D group can send the same data to numerous D2D receivers after assessing the receiver's channel demand criteria. Because multicast D2D receivers have varying channel characteristics, the D2D transmitter will make an informed selection in favor of those receivers that can meet the channel parameters established by the D2D transmitter.

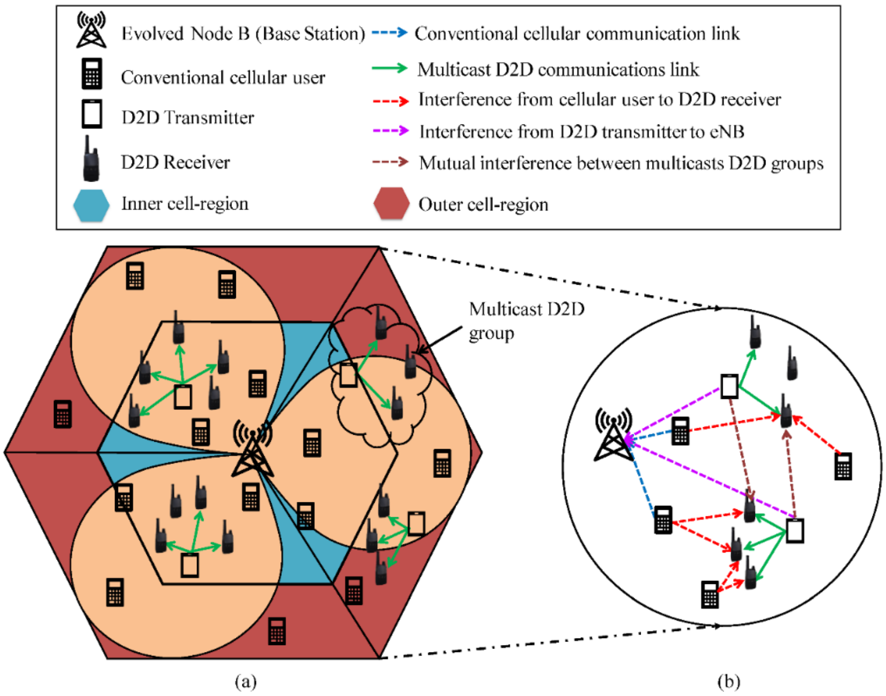


Figure 1 An Example of Conventional Auction Algorithm

2.1.2. Consensus Based Bundle Algorithm [18],[19]

The Consensus-Based Bundle Algorithm (CBBA) is a distributed task allocation algorithm that can be used for multi-agent systems. It is designed to allocate tasks among a group of agents in a decentralized and scalable manner.

The CBBA proceeds through a series of rounds, where each agent proposes a set of tasks that it can accomplish. These sets of tasks are called bundles. Each agent then broadcasts its bundle to the other agents in the system. The agents then use a consensus algorithm to determine the best bundle for each agent. The CBBA is known for its scalability, robustness, and efficiency. It has been applied in various domains, including robotics, UAVs, and sensor networks. The algorithm has been shown to be effective in situations where communication bandwidth is limited, and agents must make decisions quickly.

2.1.3. DTA Based on Sequential Item Auction (DTAP) [20]

The Dynamic Task Allocation based on Sequential Single Item Auctions (DTAP) is a distributed task allocation algorithm designed for multi-agent systems. It is based on a sequence of single-item auctions, where each auction allocates one task to one agent. The DTAP algorithm proceeds through a series of rounds, where each round consists of multiple sequential single-item auctions. In each auction, agents bid for the right to perform a single task. The auction proceeds in a descending price format, where the price of the task decreases until a single agent remains, and the task is allocated to that agent.

After each auction, the allocated task is removed from the set of available tasks, and the next auction begins. The process continues until all tasks are allocated. The DTAP algorithm is known for its simplicity, scalability, and efficiency. It has been applied in various domains, including robotics, UAVs, and sensor networks. The algorithm is particularly effective in situations where communication bandwidth is limited, and agents must make decisions quickly.

Overall, the DTAP algorithm provides a simple and efficient approach to dynamic task allocation in multi-agent systems, making it a popular choice for various applications.

CHAPTER 3

Methodology and Results

3.1. Proposed Integer Linear Programming

ILP is a mathematical optimization technique used to solve optimization problems where the objective function and constraints are linear, and the decision variables are restricted to integer values. ILP can be used to model and solve a wide range of optimization problems in various domains such as finance, logistics, scheduling, and resource allocation. For example, it can be used to optimize production schedules, vehicle routing, portfolio management, and many other problems where decisions need to be made on discrete quantities. ILP is particularly useful in situations where decision variables are required to take integer values or when the optimization problem involves logical constraints, such as selecting one option from a set of options or satisfying a set of conditions. It provides an efficient and effective way to model and solve complex optimization problems with discrete decision variables. It formulates the problem as a mathematical program and seeks an optimal solution by satisfying all constraints. Here are some characteristics of ILP:

Centralized: ILP is a centralized approach where all variables and constraints are formulated as a single optimization problem. It requires a central solver or optimizer to find the optimal solution.

Deterministic: ILP aims to find the exact optimal solution to the given problem. It relies on mathematical techniques to systematically explore the solution space and determine the best feasible solution.

Global Optimization: ILP seeks to optimize a defined objective function while satisfying a set of linear constraints. The objective function can be tailored to various optimization goals, such as maximizing revenue, minimizing costs, or maximizing resource allocation efficiency.

Complexity: ILP can be computationally expensive for large-scale problems, as the search space increases exponentially with the number of variables and constraints. Solving ILP problems requires specialized solvers and techniques.

3.2. Code Description

The code implements an ILP formulation to solve a drone routing problem. The goal is to optimize the allocation of tasks to a fleet of drones to minimize the total distance traveled by the drones. The code utilizes the matplotlib library to generate visual representations of the drone routes and task locations. It creates a grid of subplots to display multiple instances of the problem. Each subplot represents a different scenario with a specific set of randomly generated tasks and drone locations.

The ILP formulation begins by defining the decision variables, including binary variables $x_{i,j}$ to indicate whether a drone travels directly between two tasks, binary variables $y^1_{k,j}$ to represent whether a drone visits a specific task, and binary variables $y^2_{i,k}$ to indicate whether a task is visited by any drone. The objective function is then defined as the sum of the distances traveled by the drones, which includes the distances between drones and tasks (DistDT) and between tasks (DistTT). Several constraints are added to the problem. These constraints ensure that each task is visited exactly once by a drone, each drone starts and ends at its origin, and each task is visited by only one drone. Additionally, the Miller-Tucker-Zemlin (MTZ) method is used to eliminate subtours and ensure that the routes form a single closed loop. After the ILP problem is defined, it is solved using the PuLP library. The solution status and the total distance traveled by the drones are then printed. The code further visualizes the solution by plotting the drone routes and task locations on the corresponding subplot. The routes are displayed using different line styles and colors to distinguish between drone movements and task connections.

The code repeats the above process for different scenarios, varying the task and drone locations, and generates multiple subplots to compare the solutions. The resulting visualization provides insights into the effectiveness of the ILP formulation in solving the drone routing problem and optimizing the total distance traveled by the drones.

3.2.1. Formulation and Results

Table 1 Variables involved in the constraints.

Notations	Description
K	The number of Drones
N	The number of Tasks
$\mathrm{dist}DT_{i,k}$	Distance between task *i* and Drone *k*, $\forall i \in \{1,\dots,N\}$ and $\forall k \in \{1,\dots,K\}$
$\mathrm{dist}TT_{i,j}$	Distance between task *i* and task *j* $\forall i, j \in \{1,\dots,N\}$

The decision variables considered for the ILP formulation are as below:

$$x_{i,j} = \begin{cases} 1, & \text{if drone travels from task } i \text{ to task } j, \forall i, j \in \{1, \ldots, N\} \text{ and } i \neq j \\ 0, & \text{otherwise} \end{cases}$$

$$y^1_{k,j} = \begin{cases} 1, & \text{if drone travels from its origin } k \text{ to task } j, \forall k \in \{1, \ldots, K\} \text{ and } \forall j \in \{1, \ldots, N\} \\ 0, & \text{otherwise} \end{cases}$$

$$y^2_{i,k} = \begin{cases} 1, & \text{if drone travels from task } i \text{ to its origin } k, \forall i \in \{1, \ldots, N\} \text{ and } \forall k \in \{1, \ldots, K\} \\ 0, & \text{otherwise} \end{cases}$$

$$u_i = \text{time instant at which task } i \text{ is visited}, \forall i \in \{1, \ldots, N\}$$

The objective function is given as:

$$\text{minimize} \sum_{j=1}^{N} \sum_{k=1}^{K} y^1_{k,j} \ \text{dist}DT_{j,k} + \sum_{j=1}^{N} \sum_{i=1, i \neq j}^{N} x_{i,j} \ \text{dist}TT_{j,k} + \sum_{i=1}^{N} \sum_{k=1}^{K} y^2_{i,k} \ \text{dist}DT_{i,k}$$

The above objective is to minimize the total distance that a drone(s) takes to complete the task starting from its origin and returning to its origin. The term of $y^1_{k,j}$ shows that drone is starting from its origin *k* and moving to task *j*. The term with $x_{i,j}$ represents that the drone travels from task *i* to task *j*. The term with $y^2_{i,k}$ represents the return of drone(s) after completing the task.

Following are the "constraints" that are considered for the above ILP objective:

$$\sum_{j=1}^{N} \sum_{k=1}^{K} y^1_{k,j} \leq K \tag{1}$$

$$\sum_{k=1}^{K} y^1_{k,j} + \sum_{i=1}^{N} x_{i,j} = 1, \forall j \in \{1, \ldots, N\} \tag{2}$$

$$\sum_{k=1}^{K} y^2_{i,k} + \sum_{j=1}^{N} x_{i,j} = 1, \forall i \in \{1, \ldots, N\} \tag{3}$$

$$\sum_{j=1}^{N} y^1_{k,j} = \sum_{i=1}^{N} y^2_{i,k}, \forall k \in \{1, \ldots, K\} \tag{4}$$

$$u_j - u_i \geq 1 - N(1 - x_{i,j}), \forall i, j \in \{1, \ldots, N\} \text{ and } i \neq j \tag{5}$$

In the task assignment scenario, the constraint (1) dictates that the number of drones commencing from their origin to execute a task must not exceed the limit of *K* drones. This limitation ensures efficient utilization of resources and maintains a manageable workload distribution. By adhering to this constraint, the system can optimize the allocation of tasks among the available drones while considering the limitations of their origin locations. One important thing to note is that when we are considering the case of only one drone the "*k*" is opted out from the equation. Constraint (2) in the task allocation process, it is required that each task is assigned to drone in such a way that they arrive at the task location exactly once. This condition ensures that the tasks are efficiently distributed among the drones, avoiding duplication or omission. By enforcing this constraint, the system can maintain a reliable and accurate execution of tasks, optimizing the overall performance and preventing any inconsistencies or inefficiencies in the assignment process. Constraint (3) ensures that each task is assigned to drone and arrives exactly once, it is also crucial that the drone leave the task location exactly once. This requirement guarantees that the drones complete their assigned tasks and move on to subsequent assignments without any redundant or missed operations. By enforcing this constraint, the system maintains a streamlined workflow, minimizing delays and maximizing the efficiency of task execution. It also enables proper resource allocation and scheduling, allowing for effective coordination among the drones in the system. Constraint (4) as part of the task assignment process, it is essential that each drone, after completing its assigned task, returns to its original starting point. This requirement ensures that the drones maintain a closed loop trajectory, completing their mission and returning to their designated origin. By enforcing this constraint, the system can optimize the utilization of resources and ensure the efficient operation of the drones. It also allows for better planning and coordination of subsequent tasks, as the drones are available at their designated starting locations for future assignments. Constrain (5) the sub-tour elimination constraint, derived from the Miller-Tucker-Zemlin method, is a crucial component in optimizing the task assignment process. This constraint ensures that the solution does not contain any sub-tours, meaning that all drones' routes form a single connected tour. By imposing this constraint, the system prevents inefficient and overlapping routes, leading to a more optimal and streamlined allocation of tasks among the drones. This constraint enhances the overall efficiency and effectiveness of the task assignment algorithm, reducing redundancy and improving the overall performance of the system.

3.3. Examples

Ex#1 Suppose an example with one drone using the constraints that we have introduced earlier in the above section.

Objective:

$$\begin{aligned}
&\text{minimize } y_1^1\text{distDT}_1 + x_{21}\text{distTT}_1 + y_2^2\text{distDT}_2 + y_1^1\text{distDT}_1 + x_{31}\text{distTT}_1 + y_3^2\text{distDT}_3 + y_1^1\text{distDT}_1 + x_{41}\text{distTT}_1 + y_4^2\text{distDT}_4 \\
&+ y_1^1\text{distDT}_1 + x_{51}\text{distTT}_1 + y_5^2\text{distDT}_5 + y_2^1\text{distDT}_2 + x_{12}\text{distTT}_2 + y_1^2\text{distDT}_1 + y_2^1\text{distDT}_2 + x_{32}\text{distTT}_2 + y_3^2\text{distDT}_3 + \\
&y_2^1\text{distDT}_2 + x_{42}\text{distTT}_2 + y_4^2\text{distDT}_4 + y_2^1\text{distDT}_2 + x_{52}\text{distTT}_2 + y_5^2\text{distDT}_5 + y_3^1\text{distDT}_3 + x_{13}\text{distTT}_3 + y_1^2\text{distDT}_1 + \\
&y_3^1\text{distDT}_3 + x_{23}\text{distTT}_3 + y_2^2\text{distDT}_2 + y_3^1\text{distDT}_3 + x_{43}\text{distTT}_3 + y_4^2\text{distDT}_4 + y_3^1\text{distDT}_3 + x_{53}\text{distTT}_3 + y_5^2\text{distDT}_5 + \\
&y_4^1\text{distDT}_4 + x_{14}\text{distTT}_4 + y_1^2\text{distDT}_1 + y_4^1\text{distDT}_4 + x_{24}\text{distTT}_4 + y_2^2\text{distDT}_2 + y_4^1\text{distDT}_4 + x_{24}\text{distTT}_4 + y_2^2\text{distDT}_1 + \\
&y_4^1\text{distDT}_4 + x_{34}\text{distTT}_4 + y_3^2\text{distDT}_3 + y_4^1\text{distDT}_4 + x_{54}\text{distTT}_4 + y_5^2\text{distDT}_5 + y_5^1\text{distDT}_5 + x_{15}\text{distTT}_5 + y_1^2\text{distDT}_1 + \\
&y_5^1\text{distDT}_5 + x_{25}\text{distTT}_5 + y_2^2\text{distDT}_2 + y_5^1\text{distDT}_5 + x_{35}\text{distTT}_5 + y_3^2\text{distDT}_3 + y_5^1\text{distDT}_5 + x_{45}\text{distTT}_5 + y_4^2\text{distDT}_4
\end{aligned}$$

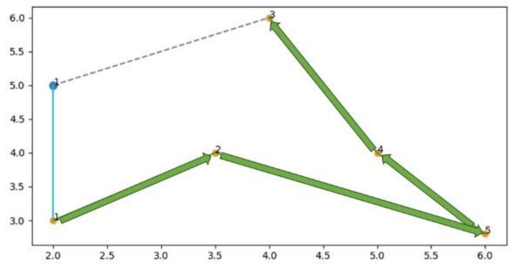


Figure 2 Drone choosing the path of its start.

Following the equation number (1) the new equation that we get for this problem is below:

$$y_1^1 + y_2^1 + y_3^1 + y_4^1 + y_5^1 = 1 \tag{6}$$

Equation (7) is derived from the equation (2).

$$\begin{aligned}
y_1^1 + x_{21} + x_{31} + x_{41} + x_{51} &= 1 \\
y_2^1 + x_{12} + x_{32} + x_{42} + x_{52} &= 1 \\
y_3^1 + x_{13} + x_{23} + x_{43} + x_{53} &= 1 \\
y_4^1 + x_{14} + x_{24} + x_{34} + x_{54} &= 1 \\
y_5^1 + x_{15} + x_{25} + x_{35} + x_{45} &= 1
\end{aligned} \tag{7}$$

Driven equation (8) and (9) is from Equation (3)

$$y_1^2 + y_2^2 + y_3^2 + y_4^2 + y_5^2 = 1 \tag{8}$$

$$
\begin{aligned}
y_1^2 + x_{12} + x_{13} + x_{14} + x_{15} &= 1 \\
y_2^2 + x_{21} + x_{23} + x_{24} + x_{25} &= 1 \\
y_3^2 + x_{31} + x_{32} + x_{34} + x_{35} &= 1 \\
y_4^2 + x_{41} + x_{42} + x_{43} + x_{45} &= 1 \\
y_5^2 + x_{51} + x_{52} + x_{53} + x_{54} &= 1
\end{aligned} \tag{9}
$$

$$u_j - u_i \geq 1 - N(1 - x_{ij}) \tag{10}$$

Solution:

Table 2 Distance calculation between Drone and Task

Drones	Task	Distance (m)
0	1	2.00
0	2	1.80
0	3	2.24
0	4	3.16
0	5	4.57

Table 3 Distance calculation between Task and Task

Task(*i*)	Task(*j*)	Distance (m)	Task(*i*)	Task(*j*)	Distance (m)
1	2	1.80	2	1	1.80
1	3	3.61	2	3	2.06
1	4	3.16	2	4	1.05
1	5	4.00	2	5	2.77
3	1	3.61	4	1	3.16
3	2	2.06	4	2	1.50
3	4	2.24	4	3	2.24
3	5	3.77	4	5	1.56
5	1	4.00	5	2	2.77

5	3	3.77	5	4	1.56

Total Distance = 12.61m

The ILP formulation selects the shortest path to completion of the journey. Distance is calculated initially from drone to task, then from task to task, and finally after making potential pathways. ILP selects the shortest one.

$$y_1^1 + y_2^1 + y_3^1 + y_4^1 + y_5^1 = 1$$

$$y_1^1 = 1$$
$$y_2^1 = y_3^1 = y_4^1 = y_5^1 = 0$$

Entering Equation:

$$x_{21} = 0,\quad x_{12} = 1,\quad x_{13} = 0,\quad x_{14} = 0,\quad x_{15} = 0$$
$$x_{31} = 0,\quad x_{32} = 0,\quad x_{23} = 0,\quad x_{24} = 0,\quad x_{25} = 1$$
$$x_{41} = 0,\quad x_{42} = 0,\quad x_{43} = 1,\quad x_{34} = 0,\quad x_{35} = 0$$
$$x_{51} = 0,\quad x_{52} = 0,\quad x_{53} = 0,\quad x_{54} = 1,\quad x_{45} = 0$$

Leaving Equation:

$$y_3^2 = 1$$
$$y_1^2 = y_2^2 = y_4^2 = y_5^2 = 0$$

Subtour-Elimination Equation:

Table 4 Subtour Equation

u_1	**1**
u_2	2
u_3	5
u_4	4
u_5	3

$$u_j - u_i \geq 1 - N(1 - x_{ij})$$

If x_{12} =1

$u_2 - u_1 \geq 1 - 5(1 - x_{12})$
$u_2 - u_1 \geq 1 - 5(1 - 1)$
$u_2 - u_1 \geq 1 - 5(0)$
$u_2 - u_1 \geq 1 - 0$ From Table we can confirm this which shows that $u_2 \geq u_1$
$u_2 - u_1 \geq 1$
$u_2 - u_1 \geq 1$
$u_2 \geq 1 + u_1$

1. x_{13} =1

$u_3 - u_1 \geq 1 - 5(1 - x_{13})$
$u_3 - u_1 \geq 1 - 5(1 - 1)$
$u_3 \geq 1 + u_1$

2. x_{14} =1

$u_4 - u_1 \geq 1 - 5(1 - x_{14})$
$u_4 - u_1 \geq 1 - 5(1 - 1)$
$u_4 \geq 1 + u_1$

3. x_{15} =1

$u_5 - u_1 \geq 1 - 5(1 - x_{15})$
$u_5 - u_1 \geq 1 - 5(1 - 1)$
$u_5 \geq 1 + u_1$

If x_{25} =1

$u_5 - u_2 \geq 1 - 5(1 - x_{25})$
$u_5 - u_2 \geq 1 - 5(1 - 1)$ From the table and figure (2) it can be seen that $u_5 \geq u_2$
$u_5 \geq 1 + u_2$

4. x_{21} =1

$u_1 - u_2 \geq 1 - 5(1 - x_{21})$
$u_1 - u_2 \geq 1 - 5(1 - 1)$
$u_1 \geq 1 + u_2$ Not possible

5. x_{23} =1

$u_3 - u_2 \geq 1 - 5(1 - x_{23})$
$u_3 - u_2 \geq 1 - 5(1 - 1)$
$u_3 \geq 1 + u_2$

6. x_{24} =1

$u_4 - u_2 \geq 1 - 5(1 - x_{24})$
$u_4 - u_2 \geq 1 - 5(1 - 1)$
$u_4 \geq 1 + u_2$

7. x_{31} =1

$u_1 - u_3 \geq 1 - 5(1 - x_{31})$
$u_1 - u_3 \geq 1 - 5(1 - 1)$
$u_1 \geq 1 + u_3$ Not possible

8. x_{32} =1

$u_2 - u_3 \geq 1 - 5(1 - x_{32})$
$u_2 - u_3 \geq 1 - 5(1 - 1)$
$u_2 \geq 1 + u_3$ Not possible

9. x_{34} =1

$u_4 - u_3 \geq 1 - 5(1 - x_{34})$
$u_4 - u_3 \geq 1 - 5(1 - 1)$
$u_4 \geq 1 + u_3$ Not possible

10. x_{35} =1

$u_5 - u_3 \geq 1 - 5(1 - x_{35})$
$u_5 - u_3 \geq 1 - 5(1 - 1)$
$u_5 \geq 1 + u_3$ Not possible

11. x_{43} =1

$u_3 - u_4 \geq 1 - 5(1 - x_{43})$
$u_3 - u_4 \geq 1 - 5(1 - 1)$
$u_3 \geq 1 + u_4$

12. x_{41} =1

$u_1 - u_4 \geq 1 - 5(1 - x_{41})$
$u_1 - u_4 \geq 1 - 5(1 - 1)$
$u_1 \geq 1 + u_4$ Not possible

13. x_{42} =1

$u_2 - u_4 \geq 1 - 5(1 - x_{42})$
$u_2 - u_4 \geq 1 - 5(1 - 1)$
$u_2 \geq 1 + u_4$ Not possible

14. x_{45} =1

$u_5 - u_4 \geq 1 - 5(1 - x_{45})$
$u_5 - u_4 \geq 1 - 5(1 - 1)$
$u_5 \geq 1 + u_4$ Not possible

15. x_{54} =1

$u_4 - u_5 \geq 1 - 5(1 - x_{54})$
$u_4 - u_5 \geq 1 - 5(1 - 1)$
$u_4 \geq 1 + u_5$

16. x_{51} =1

$u_1 - u_5 \geq 1 - 5(1 - x_{51})$
$u_1 - u_5 \geq 1 - 5(1 - 1)$
$u_1 \geq 1 + u_5$ Not possible

17. x_{52} =1

$u_2 - u_5 \geq 1 - 5(1 - x_{52})$
$u_2 - u_5 \geq 1 - 5(1 - 1)$
$u_2 \geq 1 + u_5$ Not possible

18. x_{53} =1

$u_3 - u_5 \geq 1 - 5(1 - x_{53})$
$u_3 - u_5 \geq 1 - 5(1 - 1)$
$u_3 \geq 1 + u_5$

Ex#2. This example includes the two drones. It clarifies each constraint.

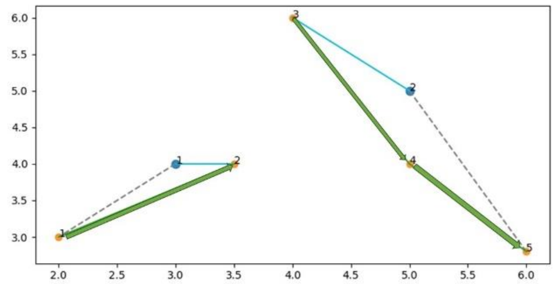


Figure 3 Two drones with 5 tasks

Table 5 Distance Calculation between Drone and Task case#2

Drone 1	Task	Distance (m)	Drone 2	Task	Distance (m)
0	1	1.41	1	1	3.61
0	2	0.50	1	2	1.80
0	3	2.24	1	3	1.41
0	4	2.00	1	4	1.00
0	5	3.23	1	5	2.42

Table 6 Distance Calculation between Task and Task case#2

Task(*i*)	Task(*j*)	Dist (m)	Task(*i*)	Task(*j*)	Dist (m)	Task(*i*)	Task(*j*)	Dist (m)
1	2	1.80	2	1	1.80	3	1	3.61
1	3	3.61	2	3	2.06	3	2	2.06
1	4	3.16	2	4	1.50	3	4	2.24
1	5	4.00	2	5	2.77	3	5	3.77
4	1	3.16	4	2	1.50	4	3	2.24

4	5	1.56	5	1	4.00	5	2	2.77
5	3	3.77	5	4	1.56			

Total Distance = 11.35m

$$y_{1,1}^{1}+y_{1,2}^{1}+y_{1,3}^{1}+y_{1,4}^{1}+y_{1,5}^{1}+y_{2,1}^{1}+y_{2,2}^{1}+y_{2,3}^{1}+y_{2,4}^{1}+y_{2,5}^{1}\leq 2 \quad (11)$$

$$\begin{aligned}
y_{1,1}^{1}+x_{21}+x_{31}+x_{41}+x_{51}&=1\\
y_{1,2}^{1}+x_{12}+x_{32}+x_{42}+x_{52}&=1\\
y_{1,3}^{1}+x_{13}+x_{23}+x_{43}+x_{53}&=1\\
y_{1,4}^{1}+x_{14}+x_{24}+x_{34}+x_{54}&=1\\
y_{1,5}^{1}+x_{15}+x_{25}+x_{35}+x_{45}&=1\\
y_{2,1}^{1}+x_{21}+x_{31}+x_{41}+x_{51}&=1\\
y_{2,2}^{1}+x_{12}+x_{32}+x_{42}+x_{52}&=1\\
y_{2,3}^{1}+x_{13}+x_{23}+x_{43}+x_{53}&=1\\
y_{2,4}^{1}+x_{14}+x_{24}+x_{34}+x_{54}&=1\\
y_{2,5}^{1}+x_{15}+x_{25}+x_{35}+x_{45}&=1
\end{aligned} \quad (12)$$

$$y_{1,1}^{2}+y_{2,1}^{2}+y_{3,1}^{2}+y_{4,1}^{2}+y_{5,1}^{2}+y_{1,2}^{2}+y_{2,2}^{2}+y_{3,2}^{2}+y_{4,2}^{2}+y_{5,2}^{2}\leq K \quad (13)$$

$$u_j-u_i\geq 1-N(1-x_{ij}) \quad (14)$$

Solution

$$y_{1,2}^{1}=y_{2,3}^{1}=1$$
$$y_{1,1}^{1}=y_{1,3}^{1}=y_{1,4}^{1}=y_{1,5}^{1}=y_{2,1}^{1}=y_{2,2}^{1}=y_{2,4}^{1}=y_{2,5}^{1}=0$$

$$\begin{aligned}
&x_{21}=1,\ x_{12}=0,\ x_{13}=0,\ x_{14}=0,\ x_{15}=0\\
&x_{31}=0,\ x_{32}=0,\ x_{23}=0,\ x_{24}=0,\ x_{25}=0\\
&x_{41}=0,\ x_{42}=0,\ x_{43}=0,\ x_{34}=1,\ x_{35}=0\\
&x_{51}=0,\ x_{52}=0,\ x_{53}=0,\ x_{54}=0,\ x_{45}=1
\end{aligned}$$

$$y_{1,1}^{2}=y_{5,2}^{2}=1$$
$$y_{2,1}^{2}=y_{3,1}^{2}=y_{4,1}^{2}=y_{5,1}^{2}=y_{1,2}^{2}=y_{2,2}^{2}=y_{3,2}^{2}=y_{4,2}^{2}=0$$

u_1	**2**

19. $x_{21} = 1$

$u_1 - u_2 \geq 1 - 5(1 - x_{21})$

$u_1 - u_2 \geq 1$

$u_1 \geq 1 + u_2$

20. $x_{12} = 1$

$u_2 - u_1 \geq 1 - 2(1 - x_{12})$

$u_2 - u_1 \geq 1$

$u_2 \geq 1 + u_1$ Not possible

u_2	*1*

21. $x_{43} = 1$

$u_3 - u_4 \geq 1 - N(1 - x_{43})$

$u_3 - u_4 \geq 1$

$u_3 \geq 1 + u_4$ Not possible

22. $x_{53} = 1$

$u_3 - u_5 \geq 1 - N(1 - x_{53})$

$u_3 - u_5 \geq 1$

$u_3 \geq 1 + u_5$ Not possible

u_3	**1**
u_4	2
u_5	3

23. $x_{34} = 1$

$u_4 - u_3 \geq 1 - 5(1 - x_{34})$

$u_4 - u_3 \geq 1$

$u_4 \geq 1 + u_3$

24. $x_{54} = 1$

$u_4 - u_5 \geq 1 - 5(1 - x_{54})$

$u_4 - u_5 \geq 1$

$u_4 \geq 1 + u_5$ Not possible

25. $x_{35} = 1$

$u_5 - u_3 \geq 1 - 5(1 - x_{35})$

$u_5 - u_3 \geq 1$

$u_5 \geq 1 + u_3$

26. $x_{45} = 1$

$u_5 - u_4 \geq 1 - 5(1 - x_{45})$

$u_5 - u_4 \geq 1$

$u_5 \geq 1 + u_4$

From the Miller-Tucker-Zemlin approach, we built the subtour-elimination equation. But to make this clearer, we may give an example that explains this strategy in more detail.

This case will demonstrate more about the subtour-elimination equation.

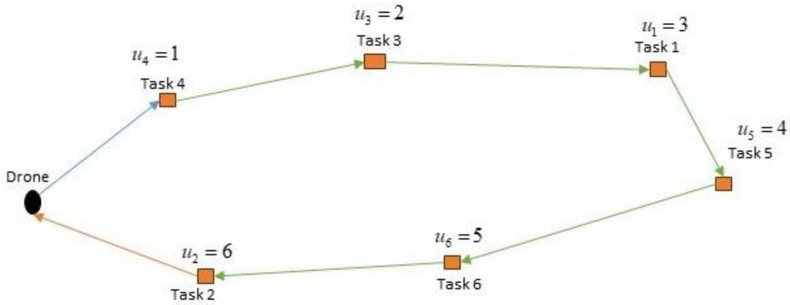


Figure 4 Subtour-Elimination Equation case#3

If there are two drones included how do the elimination equation looks like. It is concluded that there is no effect on two drones' equation each case will have their own "u_j" and "u_i" values in this scenario.

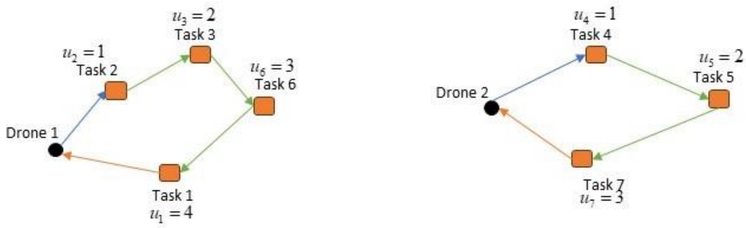


Figure 5 Subtour-Elimination Equation case#4

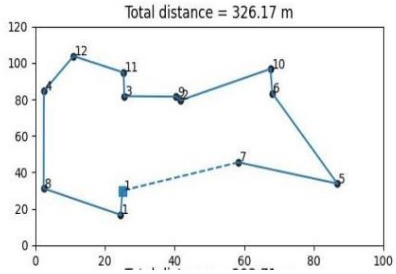
The values of elimination equation variables can be seen in the Figure 5 clearly for two drones' scenario.

.

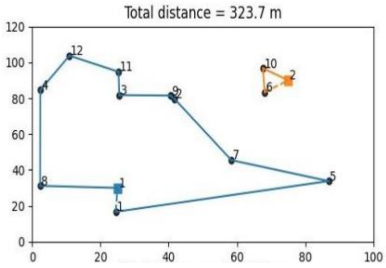
3.4. Results

The results reported demonstrate how the drones perform in various settings using Integer Linear Programming (ILP). The graph has twelve jobs, and we divided it into four sections to demonstrate how our program works. Referring to figure (a) has one drone, the figure (b) has two drones, and the figure (c) and figure (d) have three and four drones, respectively. We are examining how the journey should be done and what implications it should have on the distance traveled. Increasing the number of drones with the same number of jobs will affect how the program behaves.

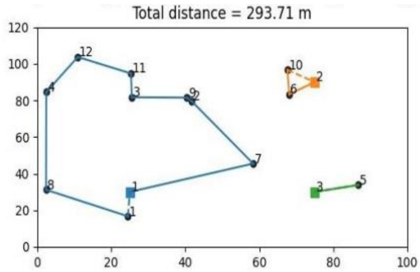
Following multiple trials, we obtained certain findings demonstrating the nature of our program based on Integer Linear Programming.



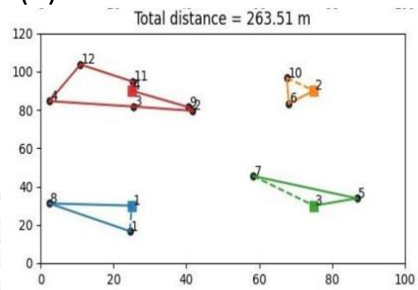
(a) One drone



(b) Two drones



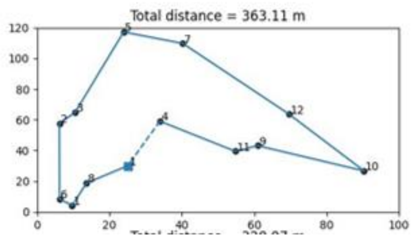
(c) Three drones



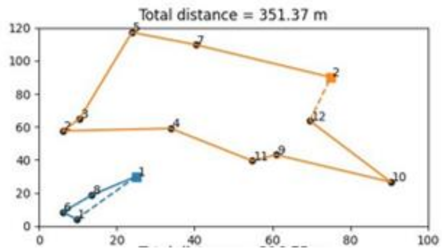
(d) Four drones

Figure 6 Experiment no.1

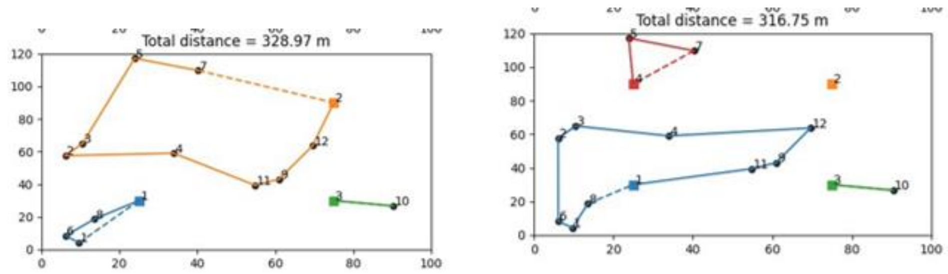
In this experiment, it can be observed that when there is just one drone, it completes all the duties and its distance is measured which is 326.17m, which can be seen in Figure 8(a). When we raise the number of drones by one, we can observe a difference in the distance, and it becomes 323.7m Figure 8(b) decreases almost three meter. When we increase the number of drones by one more, the distance becomes shorter which is in Figure 8(c) 293.71m decreases 30m than earlier. Finally, when we use four drones with twelve jobs, we can see a significant change by 30m in the distance of previous scenario and 63m from the first journey which is 263.51 in Figure 8(d). From this it can be concluded that increasing the number of drones can decrease the distance and cost of the journey. This scenario shows how the increase in drone is beneficial.



(a) One drone



(b) Two drone



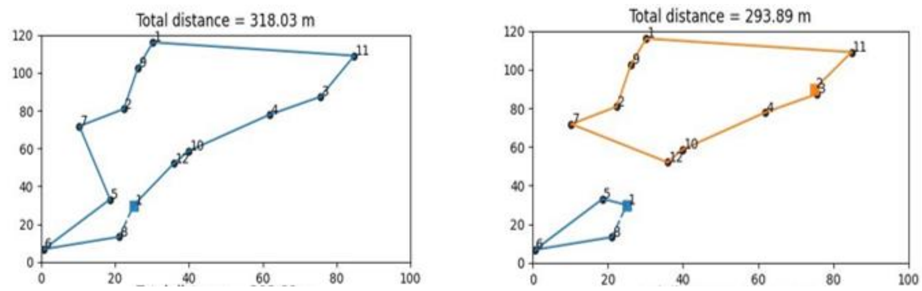
(c) Three drone (d) Four drone

Figure 7 Experiment no.2

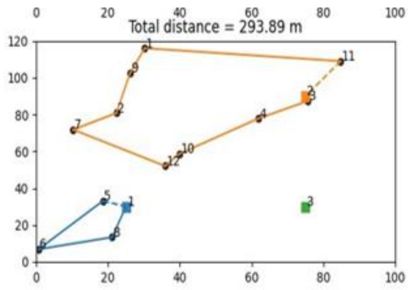
In experiment #4, it is demonstrated that employing only three drones reduces the distance and eliminates the need for a fourth. The reason is that the whichever task is near to the drone it takes that then go for next as we can see in the Figure(a) the one drone is taking responsibility of all tasks but when we increased the drone by one in Figure(b) drone one only takes task 1,6,8 where other tasks 2,3,4,5,7,9,10,11,12 go with drone two because it is more nearer to it. Figure(c) shows that after the addition of drone three the task 10 goes with it because it is nearer than the other two drones. Drone two is not used in Figure(c) because the other three drones are near to tasks and are giving a satisfactory result.

Table 7 Distance Calculation between Drone and Task

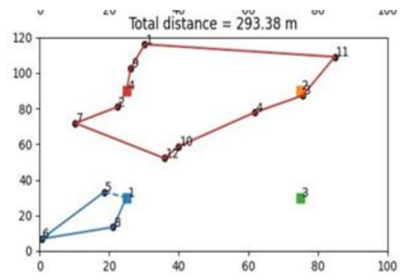
Drone	Distance (m)
1	363.11
2	351.37
3	328.97
4	316.75



(a) One Drone (b) Two drones



(c) Three drones



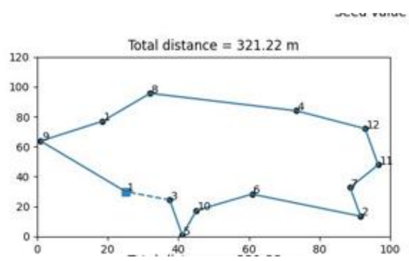
(d) Four drones

Figure 8 Experiment no. 3

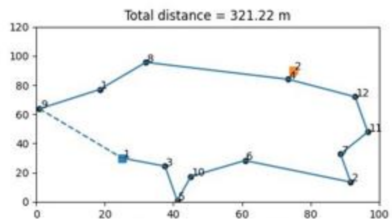
In this experimental setup, the specific task locations, yet the common theme across all scenarios is the satisfactory performance of the two drones in each graph, rendering the use of the remaining two drones unnecessary. However, a noteworthy change occurs in the last graph as the drone assigned for travel shifts from number 2 to number 4. This alteration is attributed to the inherent characteristics of the tasks. For relatively straightforward tasks, the journey's completion via drone number 4 is more expedient compared to drone number 2. As a result, this modification leads to a reduction in overall travel distance, albeit the reduction may not be substantial but remains significant in optimizing the operation.

Table 8 Distance Calculation

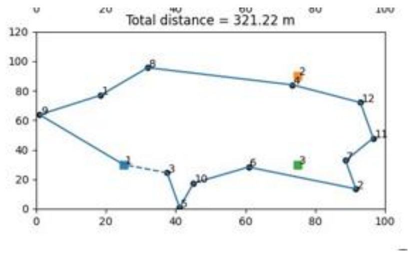
Drone	Distance (m)
1	318.03
2	293.89
3	293.89
4	293.38



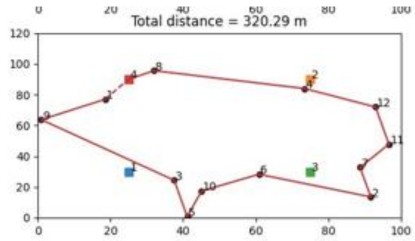
(a) One drone



(b) Two drones



(c) Three drones



(d) Four drones

Figure 9 Experiment no. 4

In this specific scenario, a single drone is exclusively employed to execute a journey, and it consistently yields satisfactory results. This optimized approach of utilizing a sole drone for journey completion is notable for its efficacy in achieving the desired objectives while minimizing operational complexity and resource allocation. By harnessing the capabilities and technologies of a single drone, it streamlines the coordination and control processes, reducing the need for multi-drone deployments, which can be operationally demanding and resource intensive. The efficiency and reliability exhibited by this solitary drone operation underscore its proficiency in providing a cost-effective and technically sound solution for journey execution, aligning with the overarching objective of enhancing performance and resource utilization.

Table 9 Distance Calculation

Drone	Distance (m)
1	321.22
2	321.22
3	321.22
4	320.29

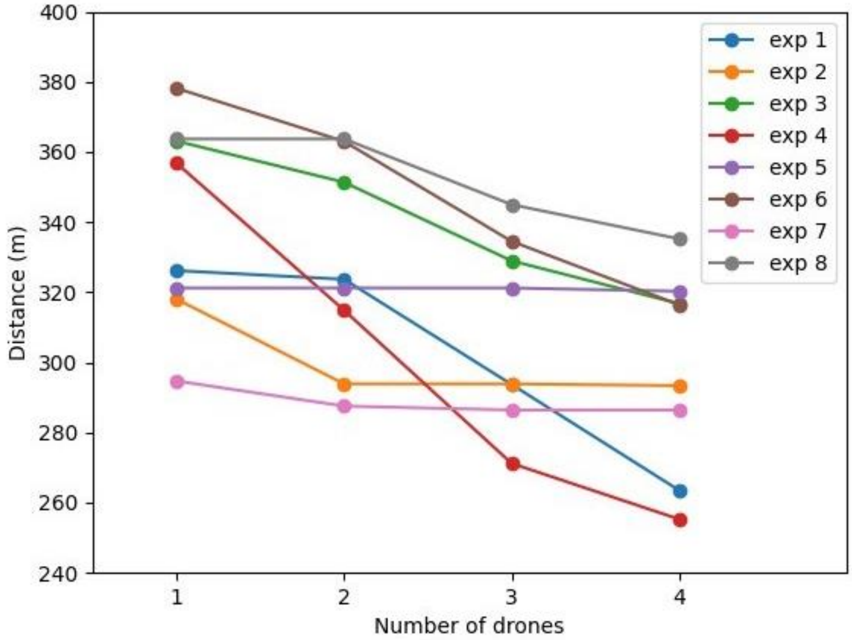


Figure 10 Results of all experiments in graphical form

The graph in question serves as a visual representation that elucidates the outcomes of individual experiments while highlighting distinctions among various predefined scenarios. Each line within this graphical representation effectively captures the way distances are attenuated or altered as a function of incrementally augmenting the count of deployed drones within the ambit of a twelve-task environment. As experiment no 1,4,6 shows more potential change than experiment no 2,3,8 but these are better than experiment no 5,7 which shows not much changes. In essence, the graph intricately dissects the intricate interplay between the number of drones and the dynamic changes in distances associated with the specific scenarios under examination.

Table 10 Time calculation Drone 1-4

Task	Drone 1	Drone 2	Drone 3	Drone 4
	Time (s)	Time (s)	Time (s)	Time (s)
10	1.04	0.82	0.72	0.70
12	4.06	2.18	1.56	1.32
14	2.59	2.45	2.18	1.53
16	2.43	2.06	3.88	2.82
20	126.94	14.18	81.71	11.29
22	75.23	11.52	26.21	15.24
24	50.08	9.49	10.60	6.37
25	31.41	6.35	51.18	24.36
27	75.63	15.21	51.81	20.3

Table 11 Time calculation Drone 5-10

Tasks	Drone 5	Drone 6	Drone 7	Drone 8	Drone 9	Drone 10
	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)	Time (s)
10	0.73	0.53	0.90	1.82	0.99	1.30
12	1.53	1.72	1.94	1.46	2.01	1.64
14	4.02	1.45	2.01	2.61	3.21	1.45
16	1.99	1.45	1.32	2.89	3.12	2.41
20	262.63	31.78	9.65	37.74	270.18	150.1
22	90.3	21.75	45.02	32.13	23.51	18.65
24	250.8	19.94	30.15	52.12	37.54	85.32
25	52.15	38.16	45.12	85.14	56.32	15.32
27	96.36	12.86	75.15	100.6	80.2	55.10

In the time calculation process, the Lenovo IdeaPad Slim 3, equipped with an Intel Core i3 10th generation processor, proves to be a reliable computational platform. The utilization of 8GB of RAM and a speedy 256GB SSD ensures efficient data access and manipulation. However, it is worth noting that some of the calculations, highlighted in red, have exhibited significant computational complexity, surpassing the one-hour or 60-minute threshold. This extended time frame is indicative of the inherent computational challenges posed by the Integer Linear Programming (ILP) constraints embedded within the problem. ILP, a mathematical optimization technique, often requires solving procedures that involve integer variables, leading to increased time complexity.

The beyond range sections from 27 could also suggest the necessity of algorithmic optimization, hardware upgrades, or parallel computing strategies to expedite the solution process. Furthermore, it's crucial to consider the impact of the problem size and the inherent limitations of the hardware, as larger instances of ILP problems can demand substantial computational resources and time. Additionally, increasing the RAM or using a more powerful CPU could potentially reduce the computational time, particularly for complex ILP problems.

Various combinations of number of drones and tasks are tested. For our available computer resources, the maximum number of tasks that the proposed ilp formulation can solve within reasonable time is 27, regardless of the number of drones being used.

CHAPTER 4

Conclusion

In summary, this research introduces a robust and highly effective solution based on Integer Linear Programming (ILP) to address the intricate challenges of drone routing and task assignment. The central focus of this endeavor is to minimize the cumulative distance traveled by drones, a pivotal factor in streamlining the cost-efficiency and overall operational effectiveness of drone logistics. The vivid visualizations generated by the code serve as tangible evidence of the approach's optimality, solidifying its status as a valuable tool for the optimization of drone-based task management. We have done experimentation on numerous numbers of tasks to check the optimality of the solution by also fluctuating the number of drones. This algorithm performs well with the 27 number of tasks, and we have checked it with the ten drones. The solution have our required result in the form of minimized distance, no subtours or loops and optimal path. After 27 tasks the algorithm takes more than sixty minutes to check for the solution. For more experimentation above the limit, it needs the advancement of machine instead of doing our experiments on.

Looking ahead, the path to further enhance this model lies in the integration of real-world data and the consideration of external factors such as weather conditions and airspace regulations. This pragmatic evolution will bolster the model's utility, enabling it to tackle the intricate challenges posed by actual operational environments. By broadening the model's capabilities in this manner, we are poised to ensure its continued role as a steadfast and indispensable tool for organizations and industries that strive to harness the full potential of drones in their logistics and operational strategies.

REFERENCES

[1] Z. Zhang, J. Wang, D. Xu and Y. Meng, "Task Allocation of Multi-AUVs Based on Innovative Auction Algorithm," *2017 10th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China, 2017, pp. 83-88, doi: 10.1109/ISCID.2017.231.

[2] Y. Fan, F. Deng and X. Shi, "Multi-robot Task Allocation and Path Planning System Design," 2020 39th Chinese Control Conference (CCC), Shenyang, China, 2020, pp. 4759-4764, doi: 10.23919/CCC50068.2020.9188762.

[3] Yang, W., Zhang, X., Zhang, H., Zhang, Q., & Ma, H. (2020). Task Assignment for Multi-UAV Surveillance Based on Improved ILP Algorithm. IEEE Access, 8, 146431-146442.

[4] Chen, X., Chen, X., Yang, X., & Zhang, L. (2018). An Integer Linear Programming Approach for UAV Task Allocation in Target Tracking. IEEE Transactions on Vehicular Technology, 67(8), 7302-7316.

[5] Guo, Y., Luo, H., Zhang, D., Xu, L., & Dong, M. (2020). UAV task assignment and path planning in precision agriculture based on improved ant colony optimization and ILP. Transactions of the Chinese Society of Agricultural Engineering, 36(16), 1-11.

[6] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li and L. Wang, "Joint Optimization of Multi-UAV Target Assignment and Path Planning Based on Multi-Agent Reinforcement Learning," in IEEE Access, vol. 7, pp. 146264-146272, 2019, doi: 10.1109/ACCESS.2019.2943253.

[7] A. Sathyan, N. D. Ernest, and K. Cohen, “An Efficient Genetic Fuzzy Approach to UAV Swarm Routing,” *https://doi.org/10.1142/S2301385016500011*, vol. 4, no. 2, pp. 117–127, May 2016, doi: 10.1142/S2301385016500011.

[8] S. Huang, R. S. H. Teo, and K. K. Tan, “Collision avoidance of multi unmanned aerial vehicles: A review,” *Annu. Rev. Control*, vol. 48, pp. 147–164, Jan. 2019, doi: 10.1016/J.ARCONTROL.2019.10.001.

[9] Z. Chen, J. Alonso-Mora, X. Bai, D. D. Harabor and P. J. Stuckey, "Integrated Task Assignment and Path Planning for Capacitated Multi-Agent Pickup and Delivery," in IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 5816-5823, July 2021, doi: 10.1109/LRA.2021.3074883.

[10] Y. Liu, Q. Wang, H. Hu, and Y. He, “A Novel Real-Time Moving Target Tracking and Path Planning System for a Quadrotor UAV in Unknown Unstructured Outdoor Scenes,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 11, pp. 2362–2372, Nov. 2019, doi: 10.1109/TSMC.2018.2808471.

[11] J. Wu, L. Zou, L. Zhao, A. Al-Dubai, L. Mackenzie, and G. Min, “A multi-UAV clustering strategy for reducing insecure communication range,” *Comput. Networks*, vol. 158, pp. 132–142, Jul. 2019, doi: 10.1016/J.COMNET.2019.04.028.

[12] C. Qu, W. Gai, J. Zhang, and M. Zhong, "A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning," *Knowledge-Based Syst.*, vol. 194, p. 105530, Apr. 2020, doi: 10.1016/J.KNOSYS.2020.105530.

[13] Y. Cai, Z. Wei, R. Li, D. W. K. Ng, and J. Yuan, "Joint Trajectory and Resource Allocation Design for Energy-Efficient Secure UAV Communication Systems," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4536–4553, Mar. 2020, doi: 10.1109/TCOMM.2020.2982152.

[14] X. Yu, C. Li, and J. F. Zhou, "A constrained differential evolution algorithm to solve UAV path planning in disaster scenarios," *Knowledge-Based Syst.*, vol. 204, p. 106209, Sep. 2020, doi: 10.1016/J.KNOSYS.2020.106209.

[15] M. D. Phung and Q. P. Ha, "Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization," *Appl. Soft Comput.*, vol. 107, p. 107376, Aug. 2021, doi: 10.1016/J.ASOC.2021.107376.

[16] W. Yao, N. Qi, N. Wan, and Y. Liu, "An iterative strategy for task assignment and path planning of distributed multiple unmanned aerial vehicles," *Aerosp. Sci. Technol.*, vol. 86, pp. 455–464, 2019, doi: 10.1016/j.ast.2019.01.061.

[17] H. Huang, D. Zhu, and F. Ding, "Dynamic task assignment and path planning for multi-auv system in variable ocean current environment," J. Intell. Robot. Syst. Theory Appl., vol. 74, no. 3–4, pp. 999–1012, Aug. 2014, doi: 10.1007/S10846-013-9870-2/METRICS

[18] H. L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," IEEE Trans. Robot., vol. 25, no. 4, pp. 912–926, 2009, doi: 10.1109/TRO.2009.2022423.

[19] G. Oh, Y. Kim, J. Ahn, and H. L. Choi, "Market-Based Task Assignment for Cooperative Timing Missions in Dynamic Environments," J. Intell. Robot. Syst. Theory Appl., vol. 87, no. 1, pp. 97–123, Jul. 2017, doi: 10.1007/S10846-017-0493-X/METRICS.

[20] A. Farinelli, L. Iocchi, and D. Nardi, "Distributed on-line dynamic task assignment for multi-robot patrolling," Auton. Robots, vol. 41, no. 6, pp. 1321–1345, Aug. 2017, doi: 10.1007/S10514-016-9579-8/FIGURES/12.

APPENDIX

APPENDIX A

```
import matplotlib.pyplot as plt
import random
import numpy as np
from pulp import *
import time  # Import the time module
# Create subplots for visualization
fig, axs = plt.subplots(2, 2)
seed_value = 42356897
### 35689742, 56897423, 68974235, 89742356, 97423568, 74235689,42356897
fig.suptitle('Seed value = ' + str(seed_value) )
mycolors = ['red','blue','green', 'cyan']
Drones = [(25, 30), (75, 90), (75, 30), (25, 90)]
nDrones = [1, 2, 3, 4]
nAxes = nDrones[-1]
nTasks = 12
drone_colors = ['C0', 'C1', 'C2', 'C3', 'C4']*5

# Loop through tasks
for ft in range(nAxes):
   print("\nft = %d" % ft)
   row = ft//2
   col = ft%2
   # Record the start time
   start_time = time.time()
   # Initialize drones and tasks lists
   AreaWidth = 100
   AreaHeight = 120
```

```
random.seed(seed_value)
Tasks = []
for k in range(nTasks):
    x = round(random.random() * AreaWidth, 2)
    y = round(random.random() * AreaHeight, 2)
    Tasks.append((x, y))
for k in range(nTasks):
    x = Tasks[k][0]
    y = Tasks[k][1]
    axs[row, col].scatter(x,y,s = 25, color ='k', alpha=0.8)
    axs[row, col].text(x+0.3,y+0.3,str(k+1))
for k in range(nDrones[ft]):
    x = Drones[k][0]
    y = Drones[k][1]
    axs[row, col].scatter(x,y,s = 60, color =drone_colors[k],
                    marker = 's', alpha=0.9)
    axs[row, col].text(x+0.3,y+0.3,str(k+1)
# Loop through drones
for drn in range(nDrones[ft]):
    # Record the start time for this drone's trip
    start_time = time.time()
    # Calculate and print the time it takes for the drone to reach each task
    for j, task in enumerate(Tasks):
        x1, y1 = Drones[drn]
        x2, y2 = task
        distance = ((x1 - x2) ** 2 + (y1 - y2) ** 2) ** 0.5
        elapsed_time = time.time() - start_time
        print(f"Drone {drn + 1} to Task {j + 1} time: {elapsed_time:.2f} seconds")
##In this loop the tasks and drones lists are plotted on subplots defined by
```

```
##'row' and 'col'. s is parameter size set and alpha controls the transparency of
##of point
   DistDT = {}
   for m in range(nDrones[ft]):
      for n in range(nTasks):
         dist = ((Drones[m][0]-Tasks[n][0])**2
               +(Drones[m][1]-Tasks[n][1])**2)**0.5
         DistDT[(m,n)] = dist
         print("DistDT",m,":",n+1,"= %.2f"%dist)
   DistTT = {}
   for m in range(nTasks):
      for n in range(nTasks):
         if m!=n:
            dist = ((Tasks[m][0]-Tasks[n][0])**2
                  +(Tasks[m][1]-Tasks[n][1])**2)**0.5
            DistTT[(m,n)] = dist
            print("DistTT",m+1,":",n+1,"= %.2f"%dist)

##DistDT is the distance from drone to task and DistTT is the distance between
##task to task. For the calculation of distance Euclidean formula is used.

   ## ILP formulation
   SRC = range(1,nTasks+1) #Source
   DST = range(1,nTasks+1) #Destination
   DRN = range(1,nDrones[ft]+1) #Drone
   Trips = [(i,j) for i in SRC for j in DST if i!=j ]
   print("Trips = ",Trips)
   prob = LpProblem("Drone Problem", LpMinimize)
   #Defining Variables
```

```
y1 = LpVariable.dicts("y1", (DRN, DST), cat ="Binary") #####
y2 = LpVariable.dicts("y2", (SRC, DRN), cat ="Binary") #####
x = LpVariable.dicts("x", (SRC, DST), cat="Binary")

u = LpVariable.dicts('u', (SRC), lowBound=1, upBound=len(Tasks), cat='Integer') # we need to keep track of the order in the tour to eliminate the possibility of subtours
#The objective function is added to 'prob' first
prob += (
    lpSum([y1[drn][j] *DistDT[(drn-1,j-1)] for j in DST for drn in DRN])
    + lpSum([x[i][j] * DistTT[(i-1,j-1)] for (i, j) in Trips])
    + lpSum([y2[i][drn] *DistDT[(drn-1,i-1)] for i in SRC for drn in DRN]),
   "Total Distance ")
#Constraint 1: Drone(s) starts from its origin and goes to its nearest task
prob += lpSum([y1[drn][j] for drn in DRN for j in DST]) <= nDrones
#Constraint 2: for each task, drone(s) must arrive exactly once
for j in DST:
   prob += lpSum([y1[drn][j] for drn in DRN] + [x[i][j] for i in SRC if i!=j]) == 1
#Constraint 3: for each task, the drone must leave exactly once
for i in SRC:
   prob += lpSum([y2[i][drn] for drn in DRN] + [[x[i][j]] for j in DST if i!=j]) == 1
#Constraint 4: The outgoing drones must return back to its origin
for drn in DRN:
   prob += lpSum(y1[drn][j] for j in DST) == lpSum(y2[i][drn] for i in SRC)
# #Constraint 5: Miller-Tucker-Zemlin (MTZ) Method for subtour elimination
for i in SRC:
   for j in DST:
      if i != j:
         prob += u[j] - u[i] >= 1 - (len(Tasks) * (1 - x[i] [j]))
prob.writeLP("myChain%d.lp"%ft)
prob.solve()
```

```
print("Status:", LpStatus[prob.status])
print("Total distance (ILP): %.2f"%pulp.value(prob.objective))
ListOfTasks = [[] for k in range(nDrones[ft]
                        )]
for drn in DRN:
    for j in DST:
        if value(y1[drn][j])==1:
            ListOfTasks[drn-1].append(j)
            break
    print('Drone ', drn, ':', ListOfTasks[drn-1])
    if len(ListOfTasks[drn-1])>0:   ### the drone is active
        i = ListOfTasks[drn-1][0]
        search = True
        while (search):
            if value(y2[i][drn])==1:
                search = False
            else:
                for j in DST:
                    if value(x[i][j])==1:
                        ListOfTasks[drn-1].append(j)
                        i = j
                        break
        print('Drone ', drn, ':', ListOfTasks[drn-1])
        axs[row, col].plot([Drones[drn-1][0],Tasks[ListOfTasks[drn-1][0]-1][0]],
                 [Drones[drn-1][1],Tasks[ListOfTasks[drn-1][0]-1][1]],
                 color=drone_colors[drn-1])
        t1 = ListOfTasks[drn-1][0]
        for t2 in ListOfTasks[drn-1][1:]:
            axs[row, col].plot([Tasks[t1-1][0],Tasks[t2-1][0]],
```

```
                    [Tasks[t1-1][1],Tasks[t2-1][1]],
                  color=drone_colors[drn-1])
            t1 = t2
          axs[row, col].plot([Tasks[ListOfTasks[drn-1][-1]-1][0],Drones[drn-1][0]],
                    [Tasks[ListOfTasks[drn-1][-1]-1][1],Drones[drn-1][1]],
              color=drone_colors[drn-1], linestyle='--')
    title_temp = np.round(value(prob.objective),2)
    axs[row, col].set_title("Total distance = " + str(title_temp) + ' m')
    # Record the end time
    end_time = time.time()
    # Calculate and print the elapsed time
    elapsed_time = end_time - start_time
    print("Elapsed time for ft = %d: %.2f seconds" % (ft, elapsed_time))
    # Show plots
    axs[row, col].set_xlim([0,AreaWidth])
    axs[row, col].set_ylim([0,AreaHeight])

    plt.show(block=False)
#Calculate the total time for the entire code
total_execution_time = time.time() - start_time
print("Total execution time: %.2f seconds" % total_execution_time).
```

AUTHOR BIOGRAPHY

NAME	**Imran Saeed Mirza**
DATE OF BIRTH	27/02/1996
PLACE OF BIRTH	Pakistan
INSTITUTIONS ATTENDED	Chulalongkorn University 2021-2023 The University of Faisalabad, Pakistan 2014-2018 Punjab College of Science, Pakistan 2012-2014 Dar-e-Arqam Schools, Pakistan 2010-2012
HOME ADDRESS	P22 Khayaban Colony no 2 Madina town, Faisalabad, Pakistan
PUBLICATION	**1.** Adaptive Modulation and Power Allocation in Green Cognitive Radio Networks (2022 19th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications, and Information Technology (ECTI-CON). Publisher IEEE Explorer **2.** Reconstruction of Digital Chemistry Lab by Incorporating Virtual Reality Platform (2022 37th International Technical Conference on Circuits/Systems, Computers and Communications. Publisher IEEE Explorer. **3.** Development of a Virtual Telecommunication System Research Laboratory (2022 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)). Publisher IEEE Explorer. **4.** A multiplayer and activities based approach to Chulalongkorn Virtual Interactive Memorial Hall 2023 9th

International Conference on Engineering, Applied Sciences, and Technology (ICEAST).

5. Task assignment and path planning of multiple unmanned aerial vehicles using Integer Linear Programming IEEE Region 10 Technical Conference 2023, Tencon-2023. Publisher IEEE Explorer.

AWARD RECEIVED Not yet