T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

EFFICIENT DEPLOYMENT OF UAVS FOR DISASTER MANAGEMENT

YUSUF ALPEREN DÖNMEZ

SUPERVISOR
PROF. DR. DIDEM GÖZÜPEK

GEBZE
2026

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

EFFICIENT DEPLOYMENT OF UAVS FOR DISASTER MANAGEMENT

YUSUF ALPEREN DÖNMEZ

SUPERVISOR
PROF. DR. DIDEM GÖZÜPEK

2026
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 15/01/2025 by the following jury.

JURY

Member
(Supervisor) : Prof. Dr. Didem Gözüpek

ABSTRACT

This study addresses the critical challenge of re-establishing communication networks in disaster-stricken areas where traditional infrastructure has been damaged. Unmanned Aerial Vehicles (UAVs) are deployed as aerial base stations to provide on-demand connectivity to victims.

Inspired by a multi-criterion optimization approach from the literature, this project formulates the deployment problem as an Integer Linear Programming (ILP) model. The objective is to maximize user coverage while minimizing deployment costs, the number of UAVs, and the distance between users and drones.

Two solution methods are implemented and compared: (1) An Optimal Solution using the Branch and Bound (B&B) algorithm via the PuLP library, and (2) A proposed low-complexity Greedy Heuristic for rapid deployment. A desktop-based Graphical User Interface (GUI) was developed to allow users to simulate various disaster scenarios (floods, earthquakes) on a grid-based map, adjust constraints such as budget and signal radius, and visualize the results.

Simulation results demonstrate that while the Optimal algorithm provides the most cost-efficient solution, the Heuristic approach achieves comparable connectivity with significantly lower computational time, making it suitable for real-time disaster response.

Project Resources: Zenodo Repository: `https://zenodo.org/records/15786174` GitHub Source Code: `https://github.com/YusufAlperenDonmez/GTU-Graduation-Project.git`

Keywords: Disaster Management, UAV Deployment, Integer Linear Programming, Heuristic Algorithms, Multi-Criterion Optimization.

ÖZET

Bu çalışma, geleneksel altyapının hasar gördüğü afet bölgelerinde iletişim ağlarının yeniden kurulması sorununu ele almaktadır. İnsansız Hava Araçları (İHA'lar), afetzedelere talep üzerine bağlantı sağlamak amacıyla havadan baz istasyonları olarak konuşlandırılmaktadır.

Literatürdeki çok kriterli bir optimizasyon yaklaşımından esinlenilen bu proje, konuşlandırma problemini bir Tamsayılı Doğrusal Programlama (ILP) modeli olarak formüle etmektedir. Amaç, dağıtım maliyetlerini, İHA sayısını ve kullanıcılar ile dronlar arasındaki mesafeyi en aza indirirken kullanıcı kapsama alanını en üst düzeye çıkarmaktır.

İki çözüm yöntemi uygulanmış ve karşılaştırılmıştır: (1) PuLP kütüphanesi aracılığıyla Dal ve Sınır (B&B) algoritması kullanılarak elde edilen Optimal Çözüm ve (2) Hızlı dağıtım için önerilen düşük karmaşıklıklı Açgözlü (Greedy) Sezgisel Algoritma. Kullanıcıların ızgara tabanlı bir harita üzerinde çeşitli afet senaryolarını (sel, deprem) simüle etmelerine, bütçe ve sinyal yarıçapı gibi kısıtlamaları ayarlamalarına ve sonuçları görselleştirmelerine olanak tanıyan masaüstü tabanlı bir Grafiksel Kullanıcı Arayüzü (GUI) geliştirilmiştir.

Simülasyon sonuçları, Optimal algoritmanın en uygun maliyetli çözümü sağlamasına rağmen, Sezgisel yaklaşımın benzer bağlantı oranlarına önemli ölçüde daha düşük hesaplama süresiyle ulaştığını ve bu durumun onu gerçek zamanlı afet müdahalesi için uygun hale getirdiğini göstermektedir.

Proje Kaynakları:
Zenodo Deposu: `https://zenodo.org/records/15786174`
GitHub Kaynak Kodu: `https://github.com/YusufAlperenDonmez/GTU-Graduation-Projec`
`git`

Anahtar Kelimeler: Afet Yönetimi, İHA Konuşlandırma, Tamsayılı Doğrusal Programlama, Sezgisel Algoritmalar, Çok Kriterli Optimizasyon.

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, **Prof. Dr. Didem Gözüpek**, for her invaluable guidance, continuous support, and insightful feedback throughout the course of this graduation project. Her mentorship was instrumental in defining the scope of this study and overcoming the technical challenges encountered during the implementation.

Yusuf Alperen Dönmez

LIST OF SYMBOLS AND ABBREVIATIONS

Abbreviations

Abbreviation	:	**Explanation**
UAV	:	Unmanned Aerial Vehicle
ILP	:	Integer Linear Programming
B&B	:	Branch and Bound Algorithm
GUI	:	Graphical User Interface
LoS	:	Line-of-Sight

Model Sets and Indices

N	:	Set of User (Victim) locations
M	:	Set of Candidate UAV locations
i	:	Index for users, $i \in N$
j	:	Index for UAV candidates, $j \in M$

Decision Variables

x_j	:	Binary variable: 1 if UAV is installed at site j; 0 otherwise
y_{ij}	:	Binary variable: 1 if user i is covered by UAV j; 0 otherwise

Parameters

C_{max}	:	Maximum total budget available
$Cost_{uav}$	:	Cost of deploying a single UAV
R	:	Maximum communication radius of a UAV
γ_{max}	:	Maximum user capacity of a single UAV
β	:	Minimum percentage of users that must be covered

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1. Background and Motivation

Natural disasters such as earthquakes, floods, and hurricanes often result in the catastrophic failure of terrestrial communication infrastructures. The immediate aftermath, known as the ”Golden Hour,” is critical for search and rescue operations. However, the lack of communication networks hampers coordination between rescue teams and victims [cite: 17-19].

Recent advancements in aerial technology have positioned Unmanned Aerial Vehicles (UAVs) as a viable solution for post-disaster communications. UAVs can be rapidly deployed as ”Mobile Helping Units” (MHUs) or aerial base stations to provide on-demand coverage [cite: 20-21]. Unlike fixed ground stations, UAVs offer flexible mobility, adaptive altitude, and a higher probability of establishing Line-of-Sight (LoS) connections[cite: 37].

However, the deployment of UAVs is a complex multi-objective optimization problem. It involves balancing limited budgets and battery constraints against the need to cover geographically dispersed victims with high signal quality [cite: 22-23].

1.2. Problem Statement

The core problem addressed in this study is the optimal placement of a limited number of UAVs in a disaster-stricken area. This is formulated as a multi-objective optimization problem where the system must simultaneously:

- Minimize the total deployment cost and number of UAVs[cite: 22].
- Maximize the number of connected victims (User Coverage)[cite: 23].
- Minimize the distance between users and UAVs to ensure QoS[cite: 22].

This problem is mathematically modeled as an Integer Linear Programming (ILP) problem[cite: 24].

1.3. Objectives and Scope

The primary objective of this project is to develop a comprehensive decision-support tool for UAV deployment. The specific goals are:

1. **Mathematical Modeling:** To formulate the UAV deployment problem as an Integer Linear Programming (ILP) model based on the framework proposed by Masroor et al.[cite: 126].

2. **Algorithm Implementation:**

   - To implement an **Optimal Solution** using the Branch and Bound (B&B) algorithm via the PuLP library to serve as a benchmark[cite: 25].
   - To develop a **Greedy Heuristic Algorithm** in Python to provide rapid, near-optimal solutions for time-critical scenarios[cite: 26].

3. **Simulation Tool Development:** To create a desktop-based **Graphical User Interface (GUI)** using Tkinter. This tool allows decision-makers to:

   - Visualize disaster grids and user distributions.
   - Adjust constraints such as Budget (C_{max}), Radius (R), and Min/Max Load (γ).
   - Run batch simulations to compare algorithmic performance.

1.4. Contributions

This study contributes to the field of disaster management and operations research by bridging the gap between theoretical optimization and practical application. The key contributions are:

- **Dual-Method Analysis:** A comparative performance analysis of Exact (ILP) vs. Heuristic methods, highlighting the trade-off between computational runtime and mathematical optimality[cite: 444].
- **Interactive Visualization:** Unlike static theoretical studies, this project provides a dynamic GUI that visualizes the "Golden Hour" response, showing exactly where UAVs should be placed to maximize coverage.
- **Performance Metrics:** The generation of detailed analytics, including Weighted Utility Scores and Connectivity Charts, to aid in strategic resource planning.

2. LITERATURE REVIEW

2.1. Introduction

The deployment of Unmanned Aerial Vehicles (UAVs) has garnered significant attention in recent years due to their agility, low cost, and ability to establish Line-of-Sight (LoS) communication links. This chapter reviews the existing literature on UAV-assisted networks, focusing specifically on their application in disaster management and the optimization techniques used for their deployment.

2.2. UAV Deployment Strategies

Early research in this domain focused primarily on the rapid deployment of UAVs to maximize general wireless coverage, regardless of the scenario.

- **Minimizing Delay:** Zhang et al. [1] investigated the fast deployment of heterogeneous UAVs. They formulated a min-max delay problem to optimize the time it takes for UAVs to cover a target area.
- **Load Balancing:** Wang et al. [2] proposed a greedy search algorithm to minimize the number of required UAVs while ensuring load balancing among them. Their work highlighted the importance of preventing any single aerial base station from becoming a bottleneck.
- **3D Placement:** Yuheng et al. [3] utilized Particle Swarm Optimization (PSO) to determine the optimal 3D placement of UAVs, demonstrating that altitude optimization is crucial for maximizing coverage range.

2.3. UAVs in Disaster Management

When applied to disaster scenarios, the focus shifts from general coverage to reliability, resilience, and emergency response.

- **Monitoring and Sensing:** Erdelj et al. [4] proposed a multi-UAV system specifically for natural disaster management, combining Wireless Sensor Networks (WSN) with aerial nodes to monitor affected areas.

- **Relay Networks:** Liu et al. [5] explored the use of UAVs as relay nodes to assist Machine-to-Machine (M2M) communications when ground infrastructure is damaged.
- **Post-Disaster Recovery:** Li et al. [6] applied an Artificial Bee Colony (ABC) algorithm to optimize the throughput of UAV base stations in post-disaster scenarios, comparing their results against Genetic Algorithms (GA).

2.4. Comparison and Research Gap

While many studies address UAV deployment or disaster management separately, few have tackled the **multi-objective** problem of simultaneously optimizing cost, coverage, and user-to-UAV distance.

Table 2.1 summarizes the key differences between this project (based on Masroor et al. [7]) and existing literature.

Table 2.1: Comparison of Related Work in UAV Deployment.

Reference	**UD**	**DM**	**Key Contribution & Limitation**
Zhang et al. [1]			Focused on delay minimization; did not consider deployment cost constraints.
Wang et al. [2]			Optimized for load balance using Greedy Search; lacked disaster-specific constraints.
Erdelj et al. [4]			Theoretical framework for disaster monitoring; lacked mathematical optimization of placement.
Liu et al. [5]			Resource allocation for M2M; focused on bandwidth rather than physical deployment.
Current Work [7]			**Joint optimization of UAV Count, Deployment Cost, and User Distance using ILP and Heuristics.**

UD: UAV Deployment, DM: Disaster Management

2.5. Summary

The literature reveals a gap in decision-support systems that can provide optimal deployment strategies under the strict budget and capacity constraints typical of the ”Golden Hour.” This thesis addresses this gap by implementing both an exact ILP model and a rapid heuristic algorithm to balance these competing objectives.

3. SYSTEM MODEL AND PROBLEM FORMULATION

3.1. System Overview and Intuition

The problem addressed in this study is the rapid deployment of an aerial communication network in a post-disaster environment. When terrestrial infrastructure fails, Unmanned Aerial Vehicles (UAVs) serve as ”floating base stations” to reconnect victims.

3.1.1. The Challenge

Deployment is a trade-off between three competing goals [cite: 302-305]:

1. **Coverage vs. Cost:** Connecting every single victim might require an unlimited budget. We must find a solution that maximizes coverage within a fixed financial limit (C_{max}).
2. **Quality of Service (QoS):** Merely covering a user is insufficient; the UAV must be physically close enough to ensure a strong signal (Line-of-Sight).
3. **Resource Constraints:** Each UAV has a limited bandwidth capacity (γ_{max}), meaning it cannot serve an infinite number of users.

The goal of our optimization model is to find the deployment configuration that satisfies these conflicting objectives simultaneously.

3.2. Air-to-Ground (AG) Channel Modeling

Before optimizing placement, we must model the communication link. We adopt the probabilistic Air-to-Ground path loss model described by Masroor et al. [cite: 270-280].

The probability of establishing a Line-of-Sight (LoS) connection depends on the elevation angle between the user and the UAV. The probability P_{LoS} is given by the sigmoid function:

$$P_{LoS} = \frac{1}{1 + \alpha \exp(-\beta(\theta - \alpha))} \tag{3.1}$$

Where α and β are environmental constants (representing urban, suburban, or rural clutter), and θ is the elevation angle calculated as:

$$\theta = \tan^{-1}\left(\frac{h}{d_{mn}}\right) \tag{3.2}$$

Here, h is the UAV altitude and d_{mn} is the horizontal distance between the UAV and the user. A higher elevation angle (meaning the UAV is more directly overhead) results in a higher probability of a clear signal [cite: 265-275].

3.3. Integer Linear Programming (ILP) Formulation

We formulate the problem as a Multi-Objective Integer Linear Programming (ILP) model.

3.3.1. Decision Variables

We define two binary variables to digitize the decision-making process [cite: 230-235]:

- **Deployment Variable (x_j):**

$$x_j = \begin{cases} 1 & \text{if a UAV is deployed at candidate location } j \\ 0 & \text{otherwise} \end{cases}$$

- **Connection Variable (y_{ij}):**

$$y_{ij} = \begin{cases} 1 & \text{if User } i \text{ is connected to UAV } j \\ 0 & \text{otherwise} \end{cases}$$

3.3.2. Objective Function

To optimize for cost, quantity, and quality simultaneously, we minimize a **Weighted Utility Function (F)**. This function penalizes high costs, excessive drone usage, and long connection distances [cite: 305-315]:

$$\text{Minimize } F = \underbrace{w_1 \sum_{j\in M} x_j}_{\text{Min UAV Count}} + \underbrace{w_2 \sum_{j\in M} K_{cost} x_j}_{\text{Min Cost}} + \underbrace{w_3 \sum_{i\in N}\sum_{j\in M} dist_{ij} y_{ij}}_{\text{Min User Distance}} \tag{3.3}$$

Where w_1, w_2, w_3 are user-defined weights. For example, increasing w_3 forces the solver to prioritize placing UAVs closer to users, potentially at a higher financial cost.

3.3.3. Model Constraints

The optimization is subject to strict operational rules [cite: 317-324]:

1. Coverage Requirement (The "Safety Net"): The system serves no purpose if it doesn't cover victims. We mandate that at least β percent of the total population ($|N|$) must be served:

$$\sum_{i \in N} \sum_{j \in M} y_{ij} \geq \beta \times |N| \tag{3.4}$$

2. Single Connection Policy: To prevent bandwidth waste, a user can connect to at most one UAV:

$$\sum_{j \in M} y_{ij} \leq 1, \quad \forall i \in N \tag{3.5}$$

3. Deployment Validity (Ghost Drone Prevention): A user cannot connect to location j unless a UAV is actually deployed there ($x_j = 1$):

$$y_{ij} \leq x_j, \quad \forall i \in N, \forall j \in M \tag{3.6}$$

4. Bandwidth Capacity: Each UAV has a limit (γ_{max}) on the number of simultaneous connections it can handle:

$$\sum_{i \in N} y_{ij} \leq \gamma_{max} x_j, \quad \forall j \in M \tag{3.7}$$

5. Budget Ceiling: The total cost of all deployed UAVs must not exceed the emergency fund C_{max}:

$$\sum_{j \in M} x_j \times K_{cost} \leq C_{max} \tag{3.8}$$

3.4. Summary of Parameters

Table 3.1 summarizes the key mathematical notations used in this model.

Table 3.1: Key Optimization Parameters.

Symbol	Description
N	Set of disaster victims (Users)
M	Set of potential UAV candidate locations
C_{max}	Maximum available budget
γ_{max}	Maximum connection capacity per UAV
β	Minimum required coverage ratio (e.g., 0.8 for 80%)
K_{cost}	Unit cost of deploying a single UAV

4. SOLUTION METHODS

4.1. Introduction

Two distinct approaches were implemented to solve the optimization problem defined in Chapter 3. The first is an exact method using Branch and Bound (B&B) to find the global optimum, serving as a benchmark. The second is a proposed Greedy Heuristic designed for rapid deployment in time-critical scenarios.

4.2. Method 1: Optimal Solution (Branch and Bound)

We utilized the **PuLP** library to interface with the CBC (Coin-OR Branch and Cut) solver. The solver employs the Branch and Bound (B&B) algorithm to explore the solution space systematically.

4.2.1. Theoretical Algorithm (Pseudocode)

The Branch and Bound algorithm works by partitioning the problem into smaller sub-problems (branching) and organizing them into a tree. Branches that cannot produce a solution better than the current best are discarded (pruning) to save computational resources[cite: 314].

4.2.2. Implementation vs. Theory

In the actual Python implementation, we do not manually manage the priority queue or the branching process shown in Algorithm 1. Instead, we define the decision variables and constraints using the PuLP framework, which abstracts these operations to the underlying CBC solver. This ensures high performance and stability.

Key Implementation Change:

- **Solver Abstraction:** The complex branching logic (lines 4-13 in pseudocode) is handled internally by the C++ solver, while the Python code focuses purely on model definition.

Algorithm 1 Branch and Bound (B&B) Logic

Require: Set of Users U, Candidate Locations M, Budget C_{max}
Ensure: Optimal UAV Deployment Set S^*
Initialize $Queue \leftarrow \{Original\_Problem\}$
$Best\_Objective \leftarrow \infty$
$S^* \leftarrow \emptyset$
while $Queue \neq \emptyset$ **do**
  Select sub-problem P from $Queue$
  Solve linear relaxation of P to get solution x_{relax} and objective Z
  if P is Infeasible **or** $Z \geq Best\_Objective$ **then**
    Prune (Discard P)
  else if x_{relax} is Integer Feasible **then**
    $Best\_Objective \leftarrow Z$
    $S^* \leftarrow x_{relax}$
  else
    Branch: Split P into sub-problems P_1, P_2 by constraining a fractional variable
    Add P_1, P_2 to $Queue$
  end if
end while
return S^*

4.3. Method 2: Proposed Greedy Heuristic

To address the exponential time complexity of exact methods, a custom Greedy Heuristic was implemented. This algorithm approximates the optimal solution by making locally optimal choices at each step.

4.3.1. Theoretical Algorithm (Pseudocode)

The heuristic follows a ”Best-First” strategy. In each iteration, it selects the candidate location that covers the maximum number of *currently uncovered* users[cite: 407].

4.3.2. Implementation Optimizations

While the pseudocode describes the theoretical logic, the Python implementation includes several optimizations for speed that differ from a direct translation of Algorithm 2:

- **Pre-computed Distance Matrix (Line 8 Optimization):** Instead of calculating Euclidean distances inside the loop, we pre-calculate a matrix D_{ij} for all user-candidate pairs. This reduces the scoring step to an $O(1)$ lookup.

Algorithm 2 Proposed Greedy Heuristic Algorithm

Require: Users U, Candidates M, Radius R, Budget B

Ensure: Deployed UAVs List D

$Uncovered \leftarrow U$
$D \leftarrow \emptyset$
$Current\_Budget \leftarrow B$
while $Current\_Budget > 0$ **and** $Uncovered \neq \emptyset$ **do**
    $Best\_Candidate \leftarrow$ Null
    $Max\_Covered \leftarrow 0$
    for $j \in M$ **do**
        if $j \notin D$ **then**
            $Score \leftarrow$ Count of users in $Uncovered$ within distance R of j
            if $Score > Max\_Covered$ **then**
                $Max\_Covered \leftarrow Score$
                $Best\_Candidate \leftarrow j$
            end if
        end if
    end for
    if $Best\_Candidate \neq$ Null **then**
        Add $Best\_Candidate$ to D
        Remove covered users from $Uncovered$
        $Current\_Budget \leftarrow Current\_Budget - Cost(j)$
    else
        Break
    end if
end while
return D

- **Set Operations (Line 16 Optimization):** The `uncovered_users` list is implemented as a Python `set()`. This allows for efficient set difference operations when removing covered users, avoiding $O(N)$ list removals.
- **Early Termination (Line 18 Optimization):** We added an explicit check: if the remaining budget is less than the minimum cost of any available UAV, the loop terminates immediately. This prevents unnecessary iterations that are not present in the theoretical model.

5. GRAPHICAL USER INTERFACE (GUI)

5.1. Overall Structure

The application is built using the **Tkinter** framework. It features a responsive two-panel layout designed for ease of use.

5.1.1. Control Panel (Left)

The left sidebar serves as the configuration hub. It contains:

- **Scenario Inputs:** Dropdowns for Grid Size (3×3 to 5×5) and Number of Users.
- **Constraint Sliders:** Inputs for Budget, Min/Max Load, and Signal Radius.
- **Progress Indicator:** A progress bar that provides real-time feedback during batch simulations (Beta Sweep).

5.1.2. Visualization Panel (Right)

The right panel utilizes **Matplotlib** embedded within Tkinter to render the disaster grid.

- **Blue Dots:** Represents the location of disaster victims.
- **Red Squares:** Represents the active, deployed UAVs.
- **Lines:** Visualizes the connection link between a user and their assigned UAV.

5.2. Result Visualization Features

Upon completion of the simulation, the application generates comparative charts.

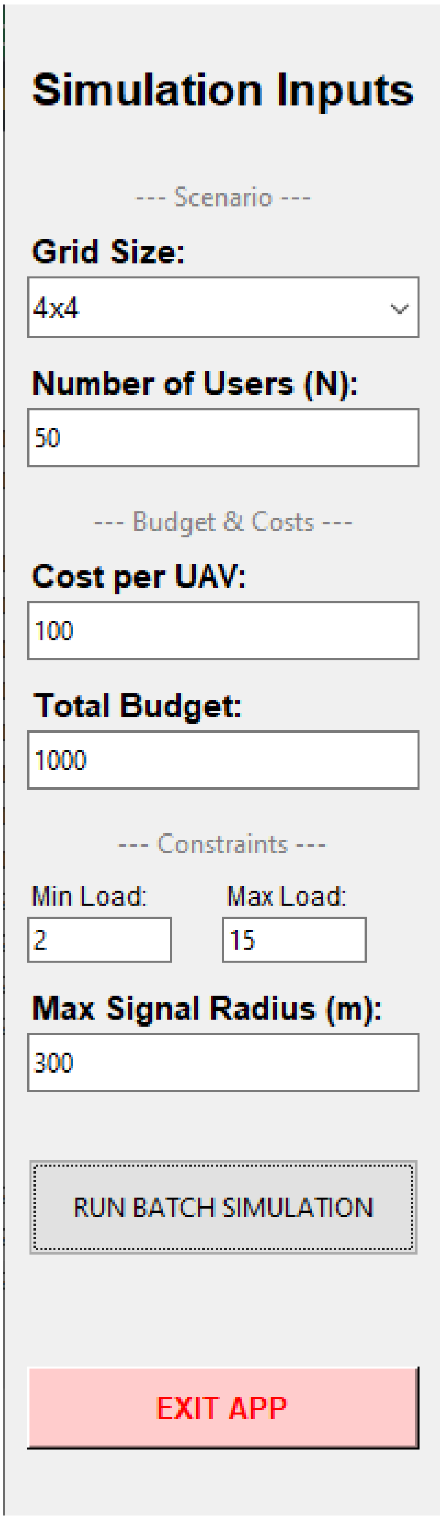

Figure 5.1: The Configuration Panel for setting simulation parameters.

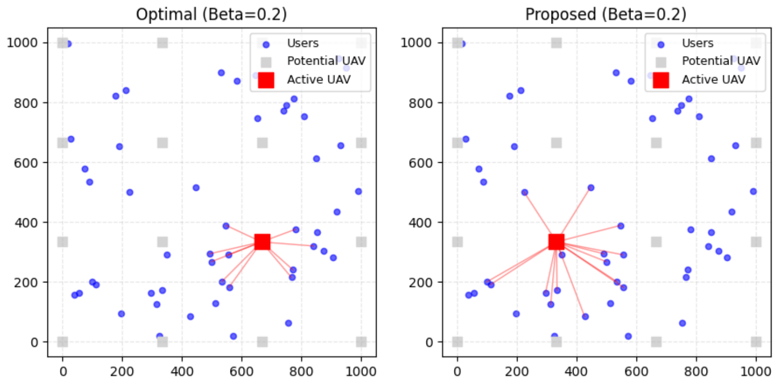

Figure 5.2: Interactive Map showing Optimal UAV placement and user connections.

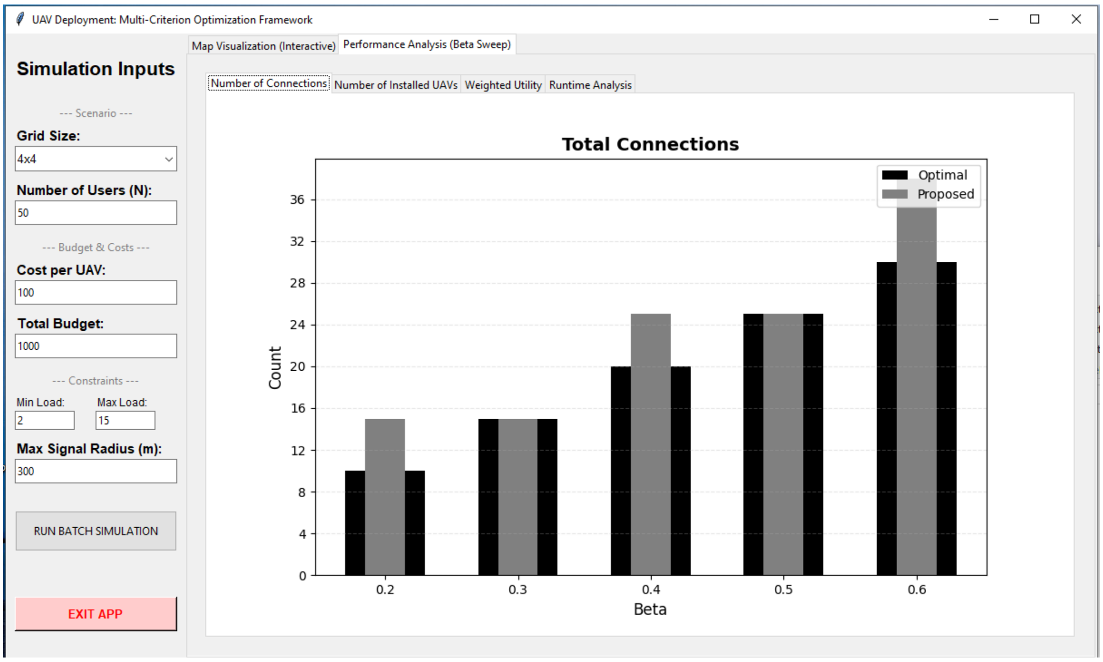

Figure 5.3: Comparative analysis of Optimal vs. Heuristic performance.

6. RESULTS AND PERFORMANCE ANALYSIS

6.1. Experimental Setup

To validate the performance of the proposed solution, we conducted extensive simulations comparing the *Optimal Branch & Bound (B&B)* algorithm against the *Proposed Greedy Heuristic*. The experiments were run on a synthetic disaster grid with the following parameters:

- **Grid Dimensions:** 4×4 (16 Candidate Locations)
- **User Distribution:** 50 Randomly distributed victims
- **UAV Constraints:** Capacity $\gamma_{max} = 10$, Radius $R = 300m$
- **Beta Sweep:** The minimum coverage requirement (β) was varied from 0.2 (20%) to 0.6 (60%).

6.2. Metric 1: Connectivity Analysis

The primary goal of the system is to maximize user coverage. Figure 6.1 illustrates the number of connected users.

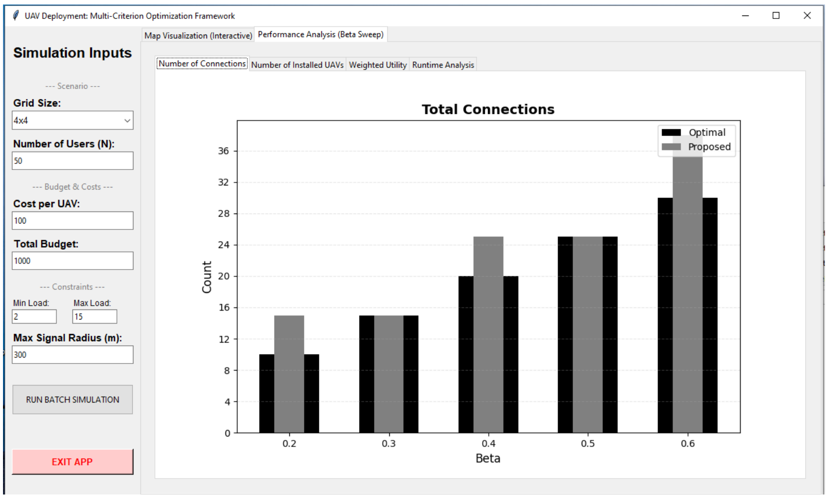

Figure 6.1: Comparison of Total Connected Users (Optimal vs. Heuristic).

6.3. Metric 2: UAV Deployment Efficiency

We analyze how many UAVs are required to achieve the target coverage.

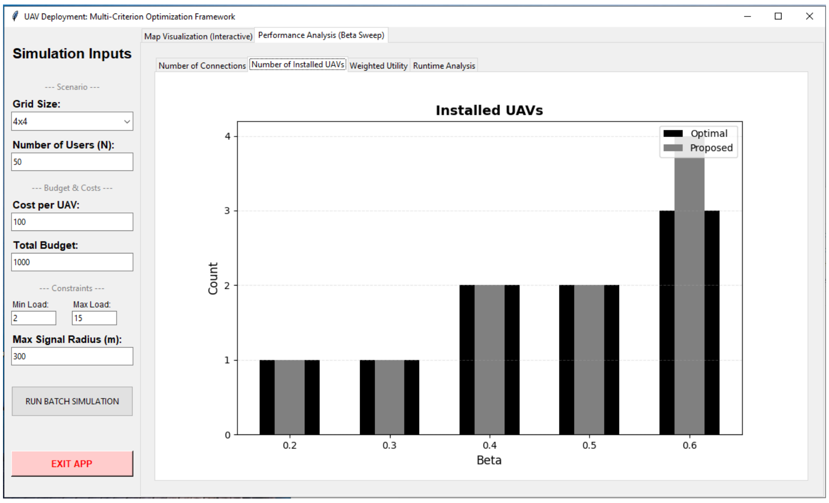

Figure 6.2: Number of Installed UAVs vs. Beta Parameter.

6.4. Metric 3: Weighted Utility Scores

The weighted utility function combines cost, distance, and penalty metrics into a single score. Lower scores are better.

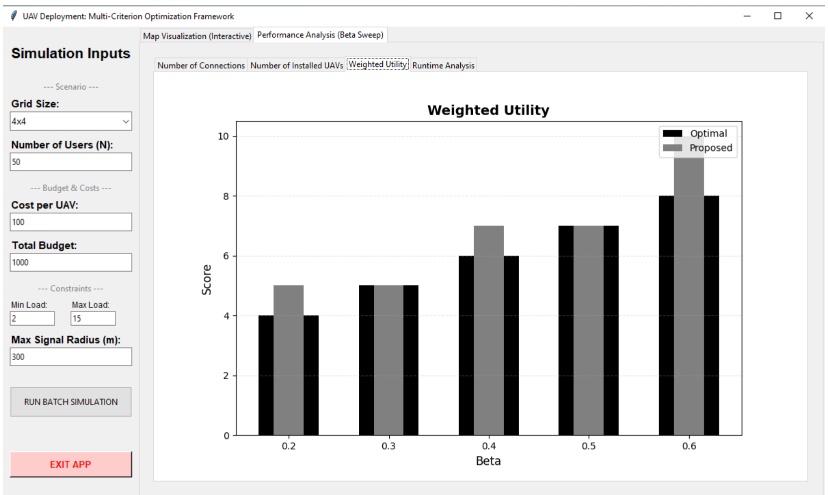

Figure 6.3: Weighted Utility Score Comparison.

6.5. Metric 4: Runtime and Complexity

The most critical distinction is computational speed. The Heuristic algorithm executes in near-constant time ($< 0.1s$) regardless of grid complexity, while the Optimal algorithm exhibits exponential growth.

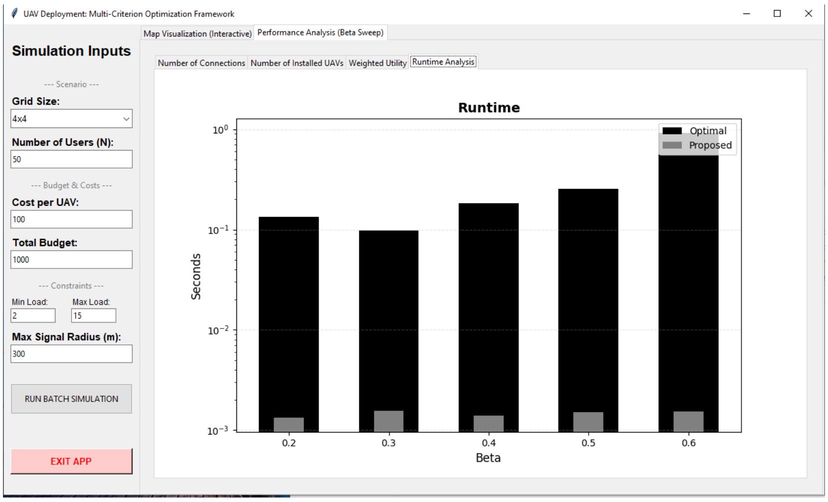

Figure 6.4: Runtime Analysis (Logarithmic Scale).

6.6. Summary of Findings

The comparative analysis conducted in this study validates the fundamental trade-off between mathematical optimality and computational feasibility in disaster management. The experimental results lead to the following key conclusions:

1. **Optimal Solution (ILP) as a Benchmark:** The Branch and Bound algorithm serves as the theoretical benchmark for the system. It consistently achieves the lowest possible weighted utility score by finding the most cost-efficient configuration of UAVs. However, its exponential runtime growth ($O(2^N)$) renders it impractical for large-scale, real-time scenarios[cite: 468]. It is best suited for:

    - **Pre-Disaster Planning:** Strategic allocation of resources when time is not a constraint.
    - **Benchmarking:** Evaluating the quality of heuristic solutions against a known perfect standard.
    - **Budget-Critical Scenarios:** Situations where financial constraints (C_{max}) are extremely tight and every unit of cost must be optimized.

2. **Heuristic Approach for Real-Time Response:** The proposed Greedy Heuristic demonstrated near-optimal coverage performance with negligible computation time (often $< 0.1s$). Although it occasionally utilizes more UAVs to achieve the same coverage ratio compared to the ILP, this slight inefficiency is a calculated cost for speed. It is the superior choice for:

   - **The "Golden Hour":** Immediate post-disaster response where delayed decision-making can lead to loss of life.
   - **Dynamic Re-planning:** Scenarios where user locations change rapidly, requiring the solution to be re-calculated every few seconds.
   - **Large-Scale Grids:** Situations involving hundreds of candidate locations where exact solvers would fail to converge within a reasonable time-frame[cite: 394].

In conclusion, this study establishes that while the ILP model provides the "perfect" solution, the Heuristic approach provides the "practical" solution necessary for the time-critical nature of disaster management.

7. CONCLUSION

This study addressed the critical challenge of re-establishing communication networks in disaster-stricken areas where traditional infrastructure has failed. By leveraging Unmanned Aerial Vehicles (UAVs) as mobile base stations, we developed a decision-support framework to optimize their deployment during the crucial "Golden Hour" of disaster response[cite: 19, 48].

The problem was formulated as a multi-objective Integer Linear Programming (ILP) model, balancing the competing goals of maximizing user coverage, minimizing deployment costs, and reducing user-to-UAV distance [cite: 23, 305-309]. To solve this, two distinct approaches were implemented and compared:

1. An **Optimal Branch and Bound Algorithm**, which provides the mathematically perfect solution but incurs high computational costs, making it suitable for static, budget-critical planning[cite: 133].

2. A **Greedy Heuristic Algorithm**, which delivers near-optimal coverage in negligible time ($< 0.1s$), proving its viability for real-time, dynamic emergency scenarios[cite: 26, 395].

Furthermore, the theoretical models were bridged to practical application through the development of an interactive **Graphical User Interface (GUI)**. This software tool allows decision-makers to visualize the deployment area, adjust constraints (e.g., budget, signal radius) on the fly, and immediately see the trade-offs between cost and coverage[cite: 1471]. The results confirm that while exact optimization is ideal for resource efficiency, heuristic methods are indispensable for the rapid responsiveness required in disaster management [cite: 553-554].

BIBLIOGRAPHY

[1] X. Zhang and L. Duan, “Fast deployment of uav networks for optimal wireless coverage,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 588–601, 2019.

[2] H. Wang, H. Zhao, W. Wu, J. Xiong, D. Ma, and J. Wei, “Deployment algorithms of flying base stations: 5g and beyond with uavs,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 009–10 027, 2019.

[3] Z. Yuheng, Z. Liyan, and L. Chumpeng, “3-d deployment optimization of uavs based on particle swarm algorithm,” in *IEEE 19th International Conference on Communication Technology (ICCT)*, 2019, pp. 954–957.

[4] M. Erdelj, M. Król, and E. Natalizio, “Wireless sensor networks and multi-uav systems for natural disaster management,” *Computer Networks*, vol. 124, pp. 72–86, 2017.

[5] X. Liu and N. Ansari, “Resource allocation in uav-assisted m2m communications for disaster rescue,” *IEEE Wireless Communications Letters*, vol. 8, no. 2, pp. 580–583, 2019.

[6] J. Li, D. Lu, G. Zhang, J. Tian, and Y. Pang, “Post-disaster unmanned aerial vehicle base station deployment method based on artificial bee colony algorithm,” *IEEE Access*, vol. 7, pp. 168 327–168 336, 2019.

[7] R. Masroor, M. Naeem, and W. Ejaz, “Efficient deployment of uavs for disaster management: A multi-criterion optimization approach,” *Computer Communications*, vol. 177, pp. 185–194, 2021.

APPENDICES

Appendix A: Software and Libraries

The following software tools and Python libraries were used to develop the simulation environment and implement the optimization algorithms:

- **Python 3.x**: The core programming language used for the entire application.
- **PuLP**: An open-source linear programming modeler used to define the decision variables, objective function, and constraints for the Optimal (ILP) solution.
- **Tkinter**: The standard Python GUI framework used to create the interactive dashboard, map visualization, and control panels.
- **Matplotlib**: Used for plotting the comparative bar charts (Connectivity, UAV Count, Runtime) and embedding the interactive grid map within the GUI.
- **NumPy**: Used for efficient numerical operations and handling the coordinate grid data.

Appendix B: Simulation Parameters

To validate the performance of the proposed algorithms, the following parameter ranges were used during the experimental simulations:

Table 7.1: Simulation Configuration Parameters.

Parameter	**Value / Range**
Grid Dimensions	4×4 (16 nodes) and 5×5 (25 nodes)
Number of Users (N)	50, 80, 100
UAV Unit Cost	100 units
Deployment Budget (C_{max})	500 – 1000 units
Max Signal Radius (R)	200m – 300m
Min Coverage Req. (β)	0.2 (20%) – 0.6 (60%)

CV

Personal Information

Name Surname	:	Yusuf Alperen Dönmez
Place and Date of Birth	:	Eskişehir, 11/24/1998
E-mail	:	yadonmez2018@gtu.edu.tr

Education

2018-2026 : B.Sc. in Computer Engineering, Gebze Technical University

Experience

- **Ektam Kıbrıs Ltd.** – Software Engineering Intern
  Description: Developed an in-house ERP Sales mobile application to optimize internal sales processes. Built the mobile interface using **Flutter** (MVVM architecture), the backend with **Node.js**, and the database with **MongoDB**. Designed user interfaces in Figma and implemented key features such as customer management, order creation, and product filtering. Gained end-to-end experience in the full software development lifecycle.

Technical Skills

- **Programming Languages:** Python, C++, Java, C, Dart
- **Tools & Libraries:** PuLP, Tkinter, Matplotlib, NumPy, Git, Node.js, Flutter