UNICO I+D Project
6G-INTEGRATION-4

E13-6G-INTEGRATION-4

Integration of 5G&6G comms into the navigation comms

Abstract

This deliverable includes the design and implementation of an optimization and control system for high-altitude platform swarms (HAPS) intended to provide 5G/6G communications coverage. Bio-inspired algorithms, namely Particle Swarm Optimization (PSO) and the Hungarian allocation algorithm (Munkres), are used to determine the optimal positions of HAPS and their dynamic reallocation in response to changes in user demand. In addition, the generation of transition trajectories is analyzed and improvements towards continuous, adaptive and scalable system integration in real environments are proposed.

Financiado por la Unión Europea NextGenerationEU

GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES

Plan de Recuperación, Transformación y Resiliencia

Document properties

Document number	E13-6G-INTEGRATION-4
Document title	Integration of 5G&6G comms into the navigation comms
Document responsible	Eduardo Cano Corral (Edair Technologies)
Document editor	Alfonso Torrelo Vaca (Edair Technologies)
Editorial team	Alfonso Torrelo Vaca (Edair Technologies)
Target dissemination level	Public
Status of the document	Edition
Version	1.0

Production properties

Reviewers	Eduardo Arca Ramirez (Edair Technologies) Eduardo Cano Corral (Edair Technologies) Daniel Segovia Vargas

Disclaimer

This document has been produced in the context of the 6G-INTEGRATION-4 Project. The research leading to these results has received funding from the Spanish Ministry of Economic Affairs and Digital Transformation and the European Union-Next Generation EU through the UNICO 5G I+D programme.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

Contents

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

List of Figures

List of Equations

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

List of Acronyms

APU – Auxiliary Power Unit

AC – Alternating Current

ANN – Artificial Neural Network

DC – Direct Current

EMS – Energy Management System

ESS – Energy Storage Systems

FLC – Fuzzy Logic Control

HESS – Hybrid Energy Storage Systems

MEA – More Electrical Aircraft

PEM – Proton Exchange Membrane

PEMFC – Proton Exchange Membrane Fuel Cell

RAT – Ram Air Turbine

RES – Renewable Energy Sources

SOC – State of Charge

MF – Membership Functions

Executive Summary

This paper describes an intelligent system for dynamic deployment of HAPS platforms aimed at improving the coverage of 5G and 6G networks. Using multi-objective optimization techniques, the system calculates in each scenario the ideal HAPS node configuration based on criteria such as coverage, efficiency and resource penalty. The PSO algorithm allows the efficient search for optimal configurations, while the Hungarian algorithm guarantees operational continuity through minimum reallocation between iterations.

In addition, a trajectory generation module is developed for the HAPS, evaluating the displacement between positions and minimizing energy and transition costs. The system is structured in a modular way, facilitating its adaptation to new environments and requirements, and allowing its extension with functionalities such as geographic exclusion, advance planning and continuous event-based operation.

This report includes a theoretical analysis of the algorithms used, a review of the source code developed in MATLAB, and a proposal of key improvements to take the system towards an operational implementation in smart grids.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

1 Introduction

The evolution towards fifth and sixth generation mobile networks poses new challenges in terms of coverage, mobility and infrastructure flexibility. In this framework, high altitude platforms (HAPS) are positioned as key elements to extend connectivity over large or difficult to access areas, acting as persistent and autonomous communication nodes.

This work presents the design and validation of a system for dynamic allocation and optimization of HAPS swarms with the objective of maximizing efficient coverage of mobile users. By applying optimization algorithms such as Particle Swarm Optimization (PSO) and combinatorial allocation techniques such as the Hungarian method, an adaptive planning of the number and position of nodes is achieved, ensuring operational continuity between scenarios.

The approach includes the penalization of suboptimal configurations due to excess nodes, lack of coverage, overlaps and unnecessary movements, as well as the generation of optimized trajectories. The platform has been simulated in MATLAB, and its modular structure allows the incorporation of future enhancements to address more realistic conditions, such as no-go zones, demand prediction, and real-time management. This report consolidates the conceptual, algorithmic, and technical basis of the proposed system, paving the way for operational applications of coordinated HAPS.

Financiado por la Unión Europea NextGenerationEU
GOBIERNO DE ESPAÑA MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL SECRETARÍA DE ESTADO DE TELECOMUNICACIONES E INFRAESTRUCTURAS DIGITALES
Plan de Recuperación, Transformación y Resiliencia
UNICO I+D

2 Study of swarm algorithms and collaborative strategies

Task allocation in UAV (Unmanned Aerial Vehicles) swarms is a central problem in applications such as surveillance, precision agriculture, search and rescue, logistics and military operations. Given the distributed, dynamic and often uncertain nature of the operating environment, designing efficient algorithms for this task is an active research challenge.

2.1 Classification of Partitioning Algorithms

The most relevant approaches are grouped into three broad categories:

2.1.1 Centralized algorithms

A central controller makes decisions for the entire swarm.

- **Advantages:** high efficiency and global optimality.
- **Disadvantages:** computational bottleneck, poor fault tolerance, not scalable.

Examples:

- **MILP (Mixed Integer Linear Programming):** used for tasks such as cooperative policing, coverage maximization or energy minimization.
- **Auction-Based with centralized auction:** auctions are used where a central entity assigns tasks according to bids.

2.1.2 Distributed Algorithms

Each UAV makes decisions autonomously, usually with partial information.

- **Advantages:** scalability, fault resilience, low communication requirements.
- **Disadvantages:** can be suboptimal, difficulty in convergence and coordination.

Examples:

- **Market-Based Distributed Auction:** UAVs negotiate with each other using protocols such as Contract Net Protocol.
- **Consensus Algorithms:** such as Vicsek or Boids rules, to synchronize decisions.
- **Token Passing / Gossip Algorithms:** to assign tasks in dynamic environments.

2.1.3 Bioinspired / Heuristic Algorithms

They are inspired by natural systems, ideal for uncertain and dynamic environments.

- **Advantages:** adaptability, scalability, low computational load.

- **Disadvantages:** lack of guaranteed optimality.

Examples:

- **Ant Colony Optimization (ACO):** used for shortest paths or sequential tasks.
- **Particle Swarm Optimization (PSO):** adapts well to multi-objective scenarios.
- **Genetic Algorithms (GA):** allow exploration of complex search spaces.
- **Bee Colony Algorithms, Firefly Algorithms:** for dynamic multi-task assignment.

2.1.4 Multi-Agent Reinforcement Learning (MARL)

- Use of algorithms such as **MADDPG**, **QMIX**, or **Multi-Agent PPO**.
- They allow task assignment in partially observable and non-stationary scenarios.
- High potential in swarms with functional heterogeneity.

Example: UAVs that learn cooperative patrolling policies under energy and coverage constraints.

2.1.5 Hybrid Algorithms

- Combination of classical approaches (MILP, heuristics) with AI (ML, RL).
- Hierarchical algorithms are used: central global planning + distributed local adaptation.

Example: task scheduling via MILP followed by dynamic reallocation with MARL.

2.1.6 Allocation under Uncertainty / Stochastic

- Dec-POMDP (Decentralized Partially Observable Markov Decision Processes) models for uncertain environments.
- Applications in search and rescue missions with incomplete information.

3 Integration of PSO algorithm with 5G&6G coverage

The growing demand for high-speed connectivity, especially with the advent of 5G and 6G networks, presents significant challenges in terms of coverage and efficiency. High Altitude Platform Stations (HAPS) emerge as a promising solution to extend coverage in hard-to-reach areas or to complement terrestrial infrastructure. However, determining the optimal location of these platforms and managing their movements efficiently is a complex optimization problem.

This project addresses that challenge by combining two powerful algorithms: Particle Swarm Optimization (PSO) and the Hungarian Algorithm (Munkres). The PSO is responsible for finding the ideal geographical positions for HAPS that maximize user coverage, while the Hungarian Algorithm is used to orchestrate the movements of existing HAPS and decide which ones should enter or leave service when conditions change, minimizing the ¸transition cost.

3.1 Operational needs assessment

The coordinated deployment of high-altitude aerial platforms (HAPS) as a solution for the provision of 5G connectivity in remote, dynamic or congested environments involves a number of **critical operational needs**. This system has been developed with the objective of providing an adaptive and optimized architecture to automate decision making regarding the location, quantity and trajectory of HAPS, considering geographical constraints, coverage demands and energy efficiency.

One of the main operational needs is that the **integration of HAPS in swarm format requires a control structure with a global purpose**, in which all units work towards a common goal of coverage, resource minimization and compliance with environmental constraints. In this sense, the system includes a multi-objective optimization model that allows to intelligently distribute the HAPS to maximize overall efficiency.

Another need is the **optimization of the user environment**, where the density, location and dynamics of end users are integrated in real time or through predicted scenarios. The system allows for a given distribution to determine the minimum number of HAPS needed and their optimal positions, using a combination of clustering (k-means) and optimization algorithms (PSO) with penalties for overlap and poor coverage.

In addition, the **reduction of overlap between coverage areas** of the different HAPS is contemplated, in order to avoid redundancies, interferences and unnecessary consumption of resources. This is managed by an explicit penalty within the cost function, which increases the value of the objective when the coverages of two or more HAPS intersect beyond what is necessary.

The system also addresses the need for the **area of effect of HAPS to cover as many users as possible**, seeking to maximize effective coverage and minimize the number of deployed platforms. This not only reduces operating costs but also improves the scalability and logistical feasibility of the system, especially in temporary deployments or emergency situations.

Another fundamental aspect is the **need for communication between HAPS and higher or lateral nodes**, such as satellites, ground stations or adjacent HAPS. This communication is essential for swarm coordination, updating optimal routes and managing the coverage network as a collaborative system.

While this module does not directly manage the physical communication layers, it does lay the foundation for nodes to make informed decisions together.

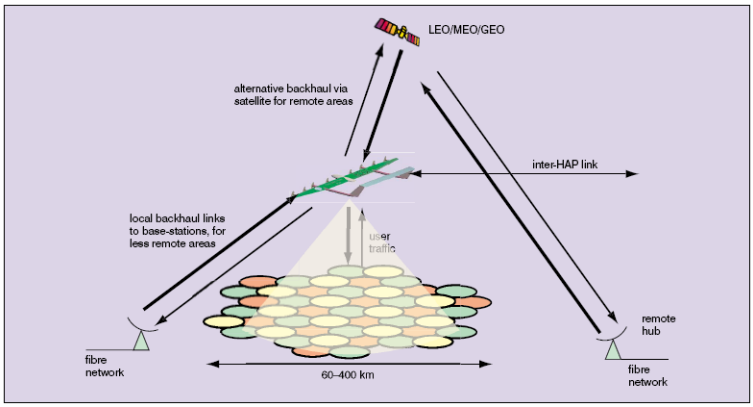

Figure 3.1 - Communication of HAPS swarm for coverage

Finally, for scenarios where users change location, **an additional HAPS trajectory and tracking layer is incorporated**, which calculates which nodes should be relocated, removed or activated. This includes an optimal allocation module using the Munkres algorithm and the generation of individualized trajectories for each platform, with persistent identification, which is essential to maintain traceability and control of the system over time.

3.2 Definition of Coverage and HAPS

In this system, a HAPS is defined as a pseudo-stationary aerial unit located in the stratosphere, whose function is to act as a coverage node for spatially distributed users on the ground. At the code level, a HAPS is represented as a circular coverage center with a fixed (or adaptive) radius and a position defined in coordinates (x, y), on a two-dimensional plane representing the geographical area of operation.

Key characteristics of a HAPS in the model:

1. **Position (x, y)**:
    - It represents the projection of the HAPS on the terrestrial plane.
    - In the optimization, these are the main variables that are modified.
    - Must be within the valid area of operation (subject to geographic restrictions if used)
2. **Constant Altitude** (implicit):
    - A fixed altitude is assumed (e.g. 20 km), so distances are calculated considering this altitude as constant when necessary (e.g., for path loss).
    - This simplifies the model by reducing it to two dimensions for decision making.
3. **Coverage Area**:
    - Each HAPS has a defined coverage radius (radio_cobertura), which delimits the circular region within which it can serve users.

- Users are considered covered if they are within this radius.
- Can be used to define penalties for overlap between HAPS.

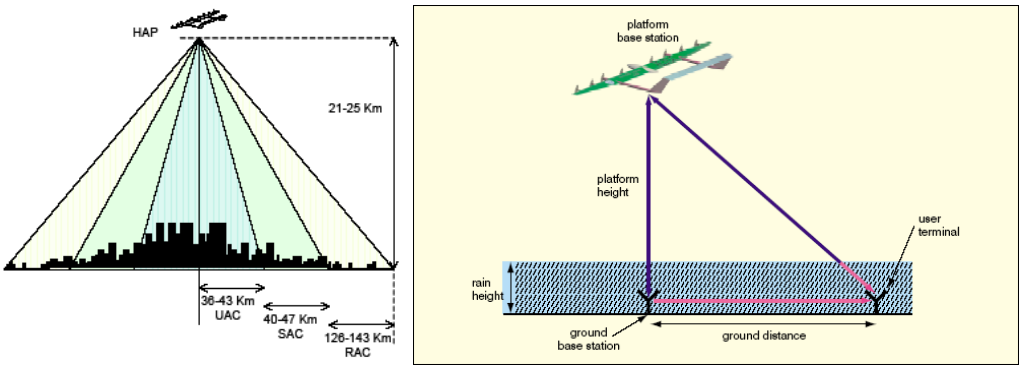

Figure 3.2 - Relation between coverage radius and height.

4. **Unique Identification**
   - Each HAPS must have an identifier (ID or index) in order to track its trajectory and to evaluate changes between successive optimal configurations.
5. **Motion Dynamics** (optional, in upper layer):
   - For adaptive systems, each HAPS can have a previous and a new position, allowing a trajectory to be generated.
   - This implies having states such as "active", "moving", or "withdrawn".
6. **Geographical Constraints (optional)**:
   - The position of a HAPS may be subject to restrictions defined by invalid zones (e.g., water, restricted airspace).
   - Penalties may be introduced if a HAPS is located within these zones or use direct masking of the search domain.

3.3 Development of PSO algorithm for optimal positioning

Particle Swarm Optimization (PSO) is a bio-inspired optimization metaheuristic that is inspired by the collective behaviour and intelligence of groups, such as flocks of birds foraging for food or schools of fish dodging predators. Developed by James Kennedy and Russell Eberhart in 1995, PSO is a global optimization technique that does not require the gradient of the function to be optimized, which makes it well suited for problems where the cost function is discontinuous, non-derivable or difficult to compute.

The algorithm developed will follow the logic proposed by the following flowchart:

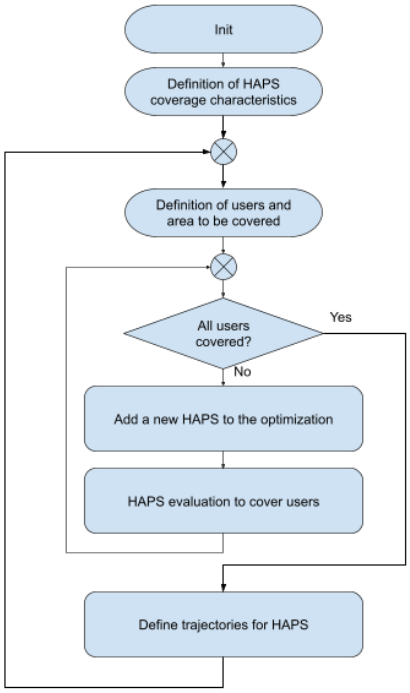

Figure 3.3 - Optimization and trajectory planning flux diagram.

3.3.1 Key Components of a Particle

In the context of PSO, each particle represents a candidate solution in the search space. A particle is a vector containing the coordinates (X, Y) of a specific number of HAPS. Each particle has:

- **Position (x_i):** The current coordinates of the candidate solution. For *k* HAPS, it would be a vector $[x_1, y_1, x_2, y_2, \dots, x_k, y_k]$.
- **Velocity (v_i):** The change vector of the particle's position. It indicates how fast and in which direction the particle will move in the next iteration.
- **Best Personal Position ($pBest_i$):** The best position (the one that resulted in the lowest objective function cost) that particle *i* has visited in its individual trajectory so far.
- **Best Global Position ($gBest$):** The best position (lowest cost) found by any particle in the entire swarm up to the current iteration. This is the collective goal that all particles seek to achieve.

3.3.2 Iterative Process and Fundamental Equations

The PSO algorithm proceeds in iterations. At each iteration *t*, the velocities and positions of all particles are updated using the following equations:

1. **Velocity Update (v_i^{t+1}):** The new velocity of particle i at iteration *t + 1* is calculated as a linear combination of three factors:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot \left(pBest_i - x_i^t\right) + c_2 \cdot r_2 \cdot (gBest - x_i^t)$$

Equation 3.1 - Velocity actualization for PSO Algorithm

Where:

- ω: Inertia Factor (controls exploration vs. exploitation).
- v_i^t: Current velocity of particle *i*.
- c_1, c_2: Acceleration coefficients (weights that determine the influence of the cognitive and social components, respectively).
- r_1, r_2: Random numbers uniformly distributed between [0,1] (introduce stochasticity).
- $pBest_i$: Best position the *i* particle has found so far.
- $gBest$: Best position found by any particle in the entire swarm so far.
- x_i^t: Current position of particle *i.*

2. **Position Update (x^{t_i+1}):** Once the new velocity is calculated, the position of the particle is updated by simply adding this velocity to its current position:

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

Equation 3.2 - Position actualization for PSO Algorithm.

The process is repeated until a convergence criterion is reached (e.g., a maximum number of iterations or a minimum improvement in the fitness function).

3.3.3 Application in the Code

In your code, the PSO is implemented through MATLAB's particleswarm function in Eval_HAPS.m. The cost function, which the PSO tries to minimize is cobertura_cost_penalizado.m. Parameters such as SwarmSize (number of particles), MaxIterations (number of swarm steps), and the bounds lb and ub (the geographic area) shape the behaviour of the PSO. The goal is to find the locations (X, Y) of the HAPS that result in the lowest possible penalty, i.e., the maximum coverage with the minimum overlap and the optimal number of HAPS.

The function cobertura_cost_penalizado.m calculates the total cost of a HAPS configuration based on multiple criteria:

Input:

- Matrix of HAPS locations and users.
- Parameters: coverage radius, penalties for number of HAPS, users not covered, overlap and displacement.

Criteria evaluated:

1. **Users covered:** using pdist2, users within the radius of at least one HAPS are identified.
2. **Number of HAPS:** penalizes configurations with many nodes.

3. **Users not covered:** high penalty to prioritize full coverage.
4. **Solape:** penalizes if a user is covered by multiple HAPS.
5. **Movement:** a term is included that evaluates the displacement with respect to the previous configuration.

Output:

Returns a scalar value representing the total cost associated with that HAPS configuration, which is used to compare alternatives in the Eval_HAPS.m function.

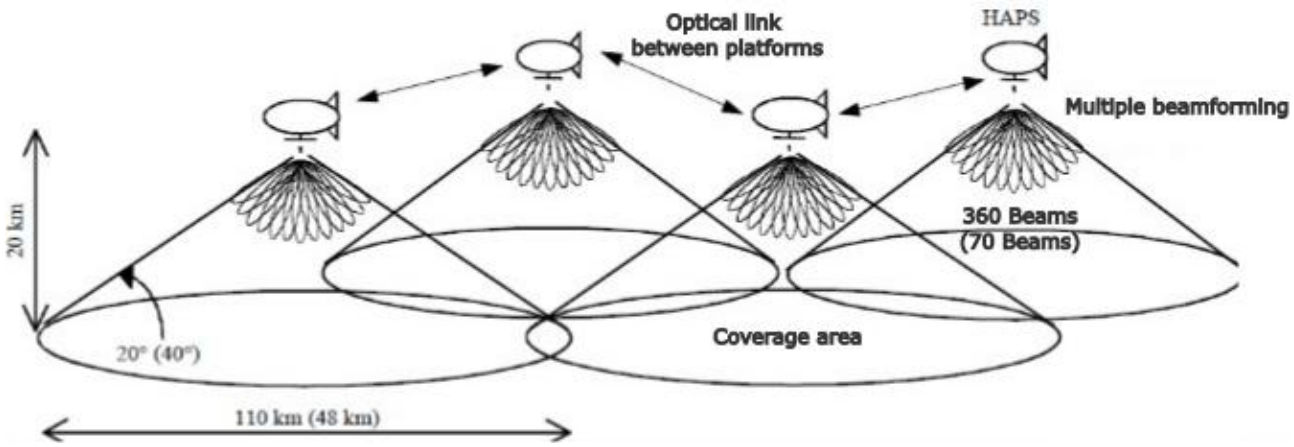

Figure 3.4 - Collaborative coverage HAPS configuration.

4 Trajectory generation for swarm positioning

Once the optimal HAPS positions for a given number of users and areas of operation are obtained from a PSO allocation, a Hungarian optimization algorithm will be used to define the outgoing and incoming HAPS in the allocation.

4.1 Implementation of Hungarian Algorithm (Munkres)

The Hungarian Algorithm, also known as the Munkres or KuhnMunkres algorithm, is a combinatorial optimization technique designed to solve the optimal assignment problem. This problem is formulated as the assignment of n agents to n tasks such that each agent performs exactly one task, each task is performed by exactly one agent, and the total cost of the assignment is minimized. It is an exact algorithm, which means that it always finds the optimal solution.

4.1.1 Intuition and Allocation Metaphor

Imagine you have a set of workers (HAPS_prev) and a set of tasks (HAPS new or new optimal locations). Each worker can perform each task, but at a different cost. The goal is to assign a worker to each task (and vice versa) so that the total cost of all assignments is as low as possible. The Hungarian algorithm seeks to identify zeros in a cost matrix (representing no-cost assignments) and transform them so that a complete assignment can be found using only those zeros.

4.1.2 Cost and Objective Matrix

The heart of the algorithm is the **cost matrix (*C*).** If we have n elements in set A and m elements in set B, the matrix C is of size n × m, where C_{ij} represents the cost of assigning element A_i to element B_j.

The objective of the Hungarian algorithm is to find a permutation of the assignments that minimizes the sum of the costs associated with those assignments.

4.1.3 Underlying Process

The Hungarian algorithm operates through a series of transformations on the cost matrix. Although the mathematical details of its steps are complex (involving operations such as reducing rows and columns, covering zeros with the fewest number of rows, and creating new zeros if a complete assignment cannot be made), the central idea is:

- **Reduce the Matrix:** values are subtracted from rows and columns to create as many zeros as possible in the matrix, without changing the optimal solution of the problem. These zeros represent ¸zero cost for a potential allocation.
- **Cover Zeros:** An attempt is made to cover all zeros in the matrix using the minimum number of rows (horizontal or vertical). If the number of rows is equal to the dimension of the matrix (e.g. n for a nxn matrix), an optimal assignment has been found.
- **Adjust the Matrix:** If all zeros cannot be covered with enough rows, an operation is performed to create more zeros or make existing zeros more accessible. This involves finding the

smallest uncovered value, subtracting it from all uncovered elements and adding it to the elements at the intersections of the lines. This step is repeated until a complete assignment can be made.

Finally, zeros are selected that represent an optimal assignment where each row and column has exactly one zero assigned to it.

4.1.4 Aplicación en el Código

The Hungarian Algorithm is used in the asignar_haps.m function.

- The cost matrix (cost_matrix) is constructed by calculating the Euclidean distances between each HAPS in the previous configuration (HAPS_prev) and each HAPS in the new optimal configuration (HAPS_new). The MATLAB function pdist2 facilitates this calculation.
- A distance threshold (max_dist) is introduced to penalize HAPS assignments that would involve unrealistically long moves with an infinite cost. This is crucial in a physics context where platform movements have limits or prohibitive costs.
- The munkres function takes this cost matrix and returns the **optimal assignment** (assignment).
- From this assignment, your code intelligently identifies which HAPS:
    - **Moving:** Those from HAPS_prev that have an assignment to a position in HAPS_new.
    - **They are "new":** Positions in HAPS_new that have not been assigned to any existing HAPS (they require a new HAPS to be deployed).
    - **They are "outgoing":** HAPS in HAPS_prev that have not been assigned to any position in HAPS_new (they can be retired or relocated).

This allocation management is critical for operational efficiency, minimizing the number of movements and the number of HAPS to be deployed or retired with each optimal configuration change.

5 Simulation Results

In this section we will present simulations performed for the proposed algorithms. These simulations have been conducted in the MATLAB environment, with random areas that have no relation with a real environment. The objective of these results is to validate that for a given number of users; the algorithms discretize and optimize the solution. To validate them, the following graphs will be presented:

- 1st Iteration: For a first configuration of users, the initial HAPS partitioning.
- Trajectories followed by the HAPS to adapt to the next iteration.
- 2nd Iteration: For the second configuration of users, the consequent distribution of HAPS is presented.

5.1 Simulation 1: HAPS leaving

In this example we present the results of starting with a configuration where more HAPS are needed than in the second one, therefore, in the generation of trajectories the outgoing HAPS will be asked to cover other positions far from the coverage area.

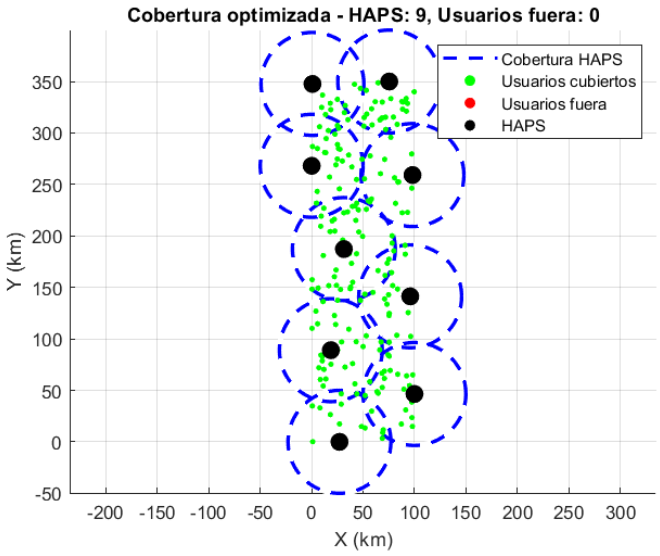

Figure 5.1 - Optimal starting position for the coverage area.

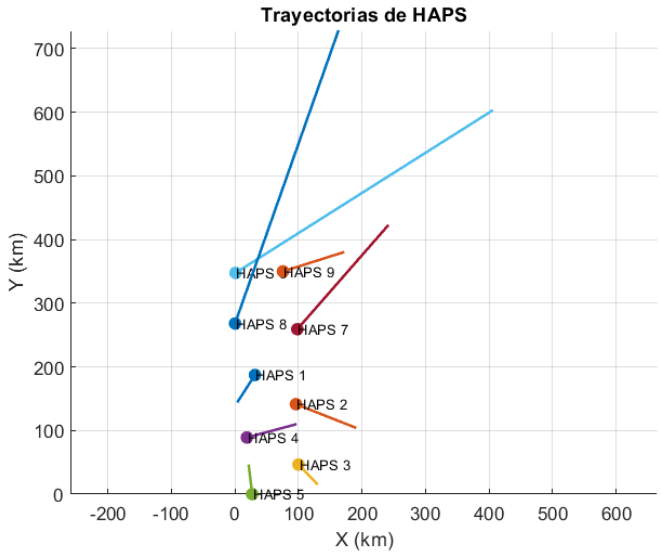

Figure 5.2 - Trajectories to optimally cover configuration 2.

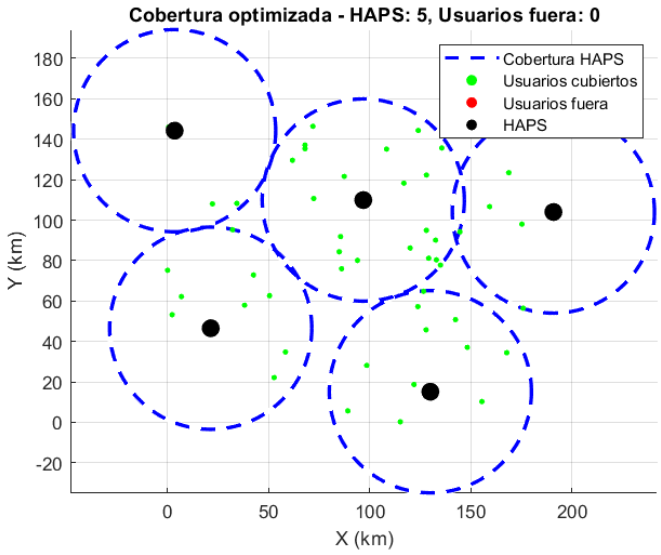

Figure 5.3 - Optimal final position for the new coverage area.

5.2 Simulation 2: HAPS entering

This example shows the results of starting with a configuration where less HAPS are needed than in the second one, therefore, in the generation of trajectories it will ask remote HAPS covering positions where they are not needed to go to cover the necessary positions within the coverage area.

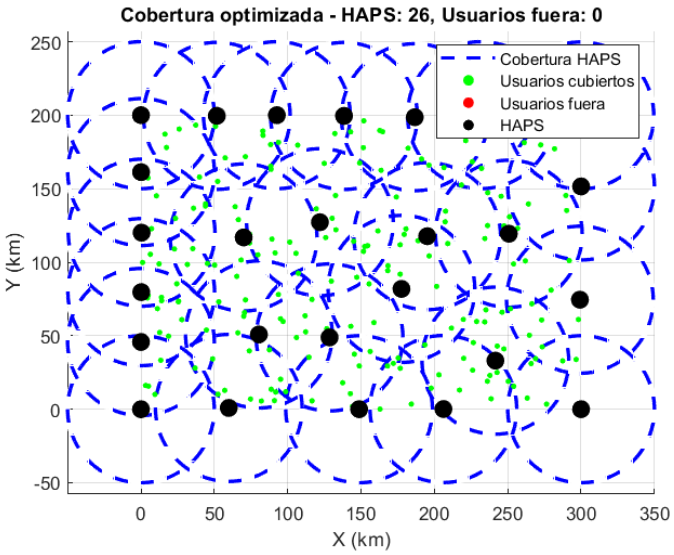

Figure 5.4 - Optimal starting position for the coverage area.

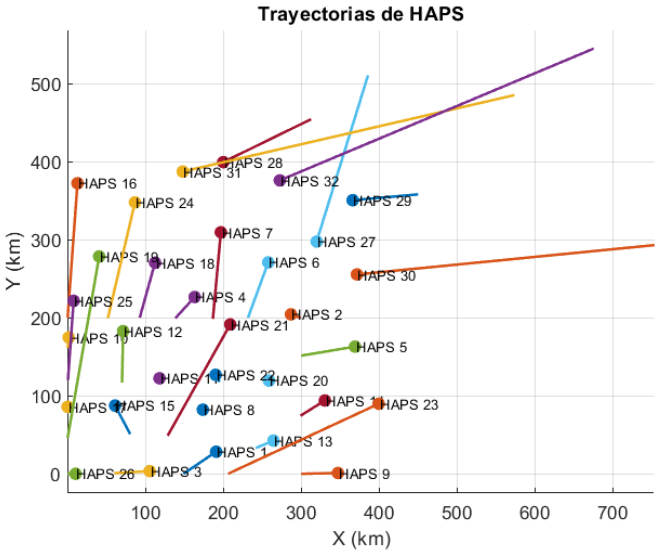

Figure 5.5 - Trajectories to optimally cover configuration 2.

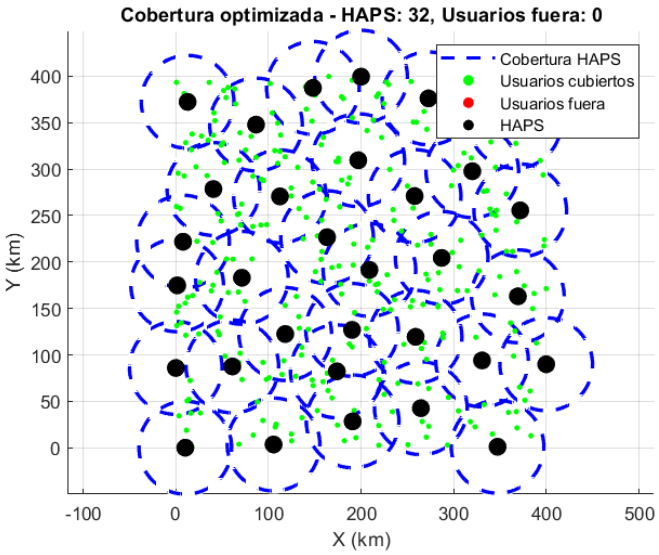

Figure 5.6 – Optimal final position for the new coverage area.

6 Implementation of algorithm improvements

The algorithms presented in this paper can be improved in order to be integrated over a swarm of HAPS platforms. Mainly improvements in optimization and realization of the following functionalities will be required:

1. **Continuous flow of optimization and sharing**
   - Restructure Integration_5G_SwarmControl_v6.m as a cyclic or event-driven process, where optimization runs continuously or on demand.
   - Allow periodic or event-driven re-optimization (e.g., new users or loss of connectivity).
   - Incorporate smooth transition mechanisms between consecutive configurations using adaptive learning.
2. **Continuous flow of trajectory generation**
   - Enhance generate_trajectories.m to support dynamic generation based on real platform kinematics.
   - Incorporate spline-based smoothing, velocity controllers and physical constraints.
   - Evaluate feasibility of trajectories based on response time and current state of HAPS.
3. **Inclusion of forbidden areas**
   - Add a binary map mask representing exclusion zones (geographical, political, or technical).
   - Adapt Eval_HAPS and generate_paths to avoid assignments or routes passing through restricted areas.
   - Evaluate heavy penalties or hard restrictions according to the type of zone.
4. **Inclusion of time requirements (planning horizon)**
   - Extend Eval_HAPS with prediction of user demand (per day/hour), using time series or historical patterns.
   - Evaluate the cost of maintaining or moving HAPS in anticipation of demand (opportunity cost vs. energy cost).
   - Introduce a planner that optimizes not only the current state, but also the future evolution of the system over a planning horizon.

These improvements will allow the system to move from a static step-by-step simulation to an operational and continuous platform capable of adapting.

7 Summary and conclusions

The developed system achieves an optimal and dynamic allocation of HAPS platforms for 5G/6G user coverage distributed in random scenarios. The simulations validate the effectiveness of the PSO+Munkres approach to generate efficient coverage with minimum number of platforms, ensuring operational continuity between iterations.

The implementation of smooth trajectories, identification of outgoing/ incoming nodes, and penalties for overlaps and non-coverage allow robust swarm management. In addition, the modular design of the code facilitates the integration of future enhancements.

Notable proposals for system enhancements include:

- Continuous event-based optimization and sharing.
- Generation of realistic trajectories with kinematic constraints.
- Inclusion of geographic exclusion zones.
- Future planning considering projected demand.

These improvements will turn the system into a tool ready for use in real operating environments, with the ability to scale and dynamically adapt to environmental and user traffic conditions.

8 References

[1] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*, New Orleans, LA, USA, 2007, pp. 1027–1035.

[2] R. W. Beard and T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton, NJ, USA: Princeton Univ. Press, 2012.

[3] C. A. Coello Coello, D. A. Van Veldhuizen, and G. B. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. Boston, MA, USA: Springer, 2007.

[4] J. Cortés, S. Martínez, T. Karatas, and F. Bullo, "Coverage control for mobile sensing networks," *IEEE Trans. Robot. Autom.*, vol. 20, no. 2, pp. 243–255, Apr. 2004.

[5] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar.–Apr. 2021.

[6] ITU-R F.747, "Technical and operational characteristics for high altitude platform stations providing IMT," Int. Telecommun. Union, Geneva, Switzerland, Rep., 2021.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, Perth, WA, Australia, 1995, vol. 4, pp. 1942–1948.

[8] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Q.*, vol. 2, no. 1–2, pp. 83–97, 1955.

[9] M. Rezaei and M. Sadeghi, "Trajectory planning for multiple UAVs using improved PSO algorithm," *J. Intell. Robot. Syst.*, vol. 91, pp. 421–432, Jan. 2018.

[10] M. Sabatini, M. Di Felice, and M. Chowdhury, "UAV swarms in 5G and beyond: A survey of applications and enabling technologies," *IEEE Access*, vol. 9, pp. 108161–108180, 2021.

[11] 3GPP TR 38.811, "Study on New Radio (NR) to support non-terrestrial networks (Release 15)," 3rd Generation Partnership Project (3GPP), Sophia Antipolis, France, Tech. Rep., 2020.

[12] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.