

# Inteligencia Artificial

## Informe Final: Nombre Proyecto

Benjamín Cisternas

22 de noviembre de 2025

### Evaluación

|                                   |       |
|-----------------------------------|-------|
| Mejoras 2da Entrega (10 %):       | _____ |
| Código Fuente (10 %):             | _____ |
| Representación (5 %):             | _____ |
| Descripción del algoritmo (15 %): | _____ |
| Experimentos (10 %):              | _____ |
| Resultados (40 %):                | _____ |
| Conclusiones (10 %):              | _____ |
| <b>Nota Final (100):</b>          | _____ |

### Resumen

Este trabajo aborda el problema de vigilancia persistente con vehículos aéreos no tripulados sobre un área discretizada en grilla, donde cada celda acumula urgencia mientras no es visitada. El objetivo principal es minimizar la urgencia total bajo restricciones de movimiento, colisión y limitaciones operacionales. Se presenta una revisión del estado del arte que recorre desde enfoques clásicos y heurísticos hasta métodos actuales basados en aprendizaje por refuerzo e híbridos. Se formalizan dos modelos matemáticos: un modelo base de minimización de urgencia y una variación que incorpora restricciones energéticas con estaciones de recarga. Finalmente, se discuten las similitudes y diferencias entre las técnicas, sus limitaciones y oportunidades de investigación futura, incluyendo optimización multiobjetivo, coordinación distribuida resiliente y validación con datos reales.

## 1. Introducción

Este documento presenta un análisis del estado del arte del problema de vigilancia persistente con vehículos aéreos no tripulados (UAVs), junto con una formalización matemática que permite comprender la complejidad computacional y las alternativas de solución propuestas anteriormente [3, 4].

El informe se organiza en cinco secciones principales. Se define formalmente el problema, sus variables, restricciones y objetivos. Luego, se revisa el estado del arte, destacando los enfoques metodológicos más relevantes desde los métodos clásicos hasta las tendencias actuales basadas en aprendizaje. Posteriormente, se formalizan dos modelos matemáticos: un modelo base de minimización de urgencia y una variación con restricciones energéticas. Finalmente, se presentan las conclusiones y líneas de trabajo futuro.

El problema de vigilancia persistente con UAVs consiste en planificar las rutas de vuelo de múltiples drones sobre una grilla bidimensional para mantener una cobertura continua del área durante una ventana temporal determinada [3]. Cada celda acumula urgencia cuando no

es visitada, y el objetivo es minimizar esta urgencia acumulada respetando restricciones de movimiento, evitación de colisiones, obstáculos y, opcionalmente, limitaciones energéticas [4, 5].

La vigilancia persistente con UAVs es un problema de relevancia práctica creciente debido a su aplicación en seguridad pública, monitoreo ambiental, protección de infraestructura crítica, búsqueda y rescate, y vigilancia de zonas de desastre [3]. Los drones ofrecen ventajas operacionales significativas: costos reducidos, menores tiempos de desplazamiento, flexibilidad y menor impacto ambiental comparado con vehículos tripulados o terrestres. Sin embargo, las limitaciones de autonomía energética, coordinación multi-agente y adaptación a entornos dinámicos plantean desafíos computacionales y de planificación que requieren modelos matemáticos precisos y algoritmos eficientes [4, 5].

## 2. Definición del Problema

El problema de vigilancia persistente con vehículos aéreos no tripulados (UAVs) es una extensión de los problemas clásicos de cobertura y patrullaje en robótica, adaptado al contexto de drones y optimización de rutas en entornos dinámicos. Este problema consiste en planificar las trayectorias de vuelo de un conjunto de drones sobre un área representada mediante una grilla bidimensional, con el objetivo de mantener la vigilancia continua de todas las celdas del espacio durante una ventana temporal determinada.

Cada celda de la grilla posee una métrica denominada urgencia, que representa la necesidad de ser monitoreada. Esta urgencia se acumula de manera continua en cada paso de tiempo  $t$  en que la celda no ha sido visitada por ningún dron. Cuando un dron sobrevuela una celda, la urgencia de dicha celda se reinicia a cero, registrándose la información correspondiente. El objetivo central del problema es minimizar la urgencia acumulada total de todas las celdas a lo largo de la ventana de tiempo  $T$ , garantizando así una cobertura eficiente y equitativa del área [3].

### 2.1. Variables del problema

Las principales variables que caracterizan el problema son las siguientes:

- $V$ : Conjunto de bases logísticas, que representan los puntos de lanzamiento y aterrizaje de los drones.
- $O$ : Conjunto de obstáculos y zonas prohibidas, correspondientes a celdas donde los drones no pueden sobrevolar.
- $W$ : Conjunto de pesos de acumulación de urgencia, que permiten diferenciar la prioridad de vigilancia entre distintas celdas.
- $k$ : Número de drones disponibles para realizar la vigilancia.
- $T$ : Ventana temporal de operación durante la cual los drones deben realizar la vigilancia.

### 2.2. Restricciones

El problema está sujeto a las siguientes restricciones operacionales [4, 5]:

- Todos los drones deben despegar desde alguna de las bases logísticas del conjunto  $V$ .
- Entre dos instantes consecutivos de tiempo  $t$  y  $t + 1$ , cada dron puede moverse a una de las ocho celdas vecinas adyacentes o permanecer estacionario en su celda actual.
- Dos drones no pueden ocupar la misma celda en el mismo instante de tiempo  $t$ , excepto en las bases logísticas.

- Los drones no pueden sobrevolar celdas que contengan obstáculos o zonas prohibidas del conjunto  $O$ .
- Para efectos de este problema base, se asume suficiente energía para completar la operación durante la ventana temporal  $T$  (sin recarga); variantes con restricciones energéticas requieren modelar recarga [5].

### 2.3. Complejidad del problema

El problema de vigilancia persistente con UAVs presenta una complejidad computacional significativa que lo clasifica como NP-hard. Los principales retos y aspectos que incrementan su complejidad son:

- **Espacio de estados exponencial:** El número de configuraciones válidas crece exponencialmente con el tamaño de la grilla, el número de UAVs y el horizonte temporal. Para una grilla de  $n \times n$  celdas,  $k$  UAVs y  $T$  pasos de tiempo, el espacio de búsqueda es del orden  $O((n^2)^{kT})$ , lo que hace inviable la exploración exhaustiva incluso para instancias pequeñas.
- **Coordinación multi-agente:** La necesidad de coordinar múltiples UAVs simultáneamente introduce dependencias complejas entre las decisiones de cada agente. Las restricciones de no colisión requieren sincronización espaciotemporal, multiplicando la complejidad de planificación y dificultando la descomposición del problema.
- **Naturaleza dinámica de la urgencia:** La urgencia acumulada en cada celda evoluciona dinámicamente en función de las visitas previas, creando un sistema de retroalimentación donde las decisiones pasadas afectan el valor futuro del objetivo. Esto impide aplicar estrategias greedy simples y requiere visión de largo plazo.
- **Restricciones energéticas:** La autonomía limitada de los UAVs introduce restricciones de alcance que acoplan decisiones locales (próximo movimiento) con planificación global (garantizar retorno a base). La planificación de recarga añade una capa adicional de complejidad al requerir balance entre vigilancia y mantenimiento operacional.
- **Incertidumbre y robustez:** En escenarios reales, el problema enfrenta incertidumbre en la localización, fallas de comunicación, cambios ambientales (viento, obstáculos emergentes) y fallas de equipos. Diseñar soluciones robustas que mantengan desempeño aceptable bajo estas perturbaciones amplifica significativamente la dificultad.
- **Objetivos conflictivos:** Frecuentemente se requiere optimizar múltiples objetivos simultáneos (minimizar urgencia, consumo energético, tiempo de misión, riesgo) que presentan trade-offs inherentes, transformando el problema en optimización multiobjetivo con frontera de Pareto compleja.

Estos factores convierten la vigilancia persistente con UAVs en un problema de optimización combinatoria altamente complejo que requiere el desarrollo de heurísticas especializadas, meta-heurísticas o métodos de aprendizaje para obtener soluciones de calidad aceptable en tiempos de cómputo razonables.

### 2.4. Problemas relacionados

El problema de vigilancia persistente está relacionado con otros problemas clásicos de robótica y optimización:

- Patrullaje multi-robot: minimizar el tiempo de inactividad (idleness) de puntos de interés mediante visitas repetidas [1].

- Cobertura de rutas (Coverage Path Planning): encontrar trayectorias que cubran completamente un área de interés, minimizando superposiciones y tiempos de vuelo (véase la revisión general en [3]).
- Patrullaje adversarial: prevención del acceso no detectado de agentes hostiles en el área de vigilancia [1].

## 2.5. Variantes conocidas

Existen diversas variantes del problema de vigilancia persistente con UAVs [4, 5]:

- Vigilancia con restricciones energéticas: autonomías limitadas y estaciones de recarga (o UGVs como cargadores móviles) [5].
- Tasas de urgencia variables: celdas con diferentes tasas de acumulación de urgencia en función del tiempo, priorización dinámica (véase [3]).
- Entornos dinámicos: cambios en la transitabilidad (nuevos obstáculos o zonas prohibidas) durante la operación [3].

El problema abordado en este documento corresponde a la formulación base de la vigilancia persistente con múltiples UAVs sobre una grilla 2D, con el objetivo de minimizar la urgencia acumulada, considerando restricciones de movimiento, colisión y zonas prohibidas.

## 3. Estado del Arte

### 3.1. Origen e historia del problema

El problema de vigilancia persistente con UAVs tiene sus raíces en los años 1990 y principios de 2000, cuando los avances en tecnología de vehículos no tripulados y la demanda de aplicaciones militares y de seguridad motivaron el desarrollo de sistemas autónomos de monitoreo. Originalmente, surge en el contexto militar estadounidense, donde la necesidad de mantener vigilancia continua sobre áreas estratégicas (fronteras, zonas de conflicto, bases militares) impulsó la investigación en planificación de rutas para flotillas de UAVs [3].

Durante la década de 2000, el problema se formalizó académicamente como una extensión de los problemas clásicos de cobertura de rutas (Coverage Path Planning) y patrullaje multi-robot, incorporando restricciones específicas de los UAVs: autonomía energética limitada, dinámicas de vuelo, comunicación intermitente y coordinación descentralizada. Los primeros trabajos se enfocaron en garantizar cobertura completa mediante rutas cíclicas predefinidas, asumiendo entornos estáticos y conocidos [3].

A partir de 2010, con el auge de los sistemas multi-agente y la democratización de plataformas UAV comerciales (drones de bajo costo), el problema se expandió hacia aplicaciones civiles: monitoreo ambiental, respuesta a desastres, vigilancia de infraestructura crítica, agricultura de precisión y búsqueda y rescate [4]. Este cambio de contexto introdujo nuevos desafíos: entornos dinámicos e inciertos, requisitos de operación en tiempo real, restricciones regulatorias y necesidad de soluciones escalables y robustas.

En la última década (2015-2025), el problema ha evolucionado hacia formulaciones más complejas que integran aprendizaje por refuerzo, optimización multiobjetivo, gestión energética con estaciones de recarga dinámicas, y coordinación resiliente ante fallas. La tendencia actual combina enfoques clásicos de optimización con técnicas de inteligencia artificial para lograr sistemas adaptativos que operen de manera autónoma en condiciones reales [6, 2].

### 3.2. Primeros enfoques

Inicialmente, se garantizaron coberturas continuas con algoritmos deterministas de recorridos repetitivos y políticas reactivas; sin embargo, presentaban baja adaptabilidad ante cambios del entorno o fallas de comunicación [1]. Los primeros métodos empleaban:

- **Rutas cíclicas predefinidas:** Partición del área en regiones y asignación de ciclos de patrullaje fijos a cada UAV. Aunque simples y predecibles, carecían de capacidad de adaptación ante eventos inesperados.
- **Diagramas de Voronoi:** Partición del espacio en regiones de responsabilidad mediante celdas de Voronoi, donde cada UAV patrulla su región asignada. Este enfoque facilita la descentralización pero requiere recomputación ante cambios topológicos.
- **Campos potenciales artificiales:** Modelado de la urgencia como un campo potencial atractivo y los obstáculos/UAVs como campos repulsivos. Los UAVs se mueven siguiendo el gradiente del campo resultante. Presenta problemas de mínimos locales y requiere ajuste cuidadoso de parámetros.

Con el uso de múltiples UAVs, aparecieron métodos de coordinación distribuida para asignar dinámicamente zonas de patrullaje y mejorar la escalabilidad bajo restricciones de energía y comunicación [4]. Estos enfoques pioneros establecieron las bases conceptuales del problema, aunque sus limitaciones motivaron el desarrollo de técnicas más sofisticadas.

### 3.3. Métodos clásicos y heurísticos

Los enfoques clásicos se centran en planificación determinista con algoritmos de búsqueda en grafos y heurísticas constructivas:

- **A\* y variantes:** Búsqueda de caminos óptimos considerando urgencia acumulada como costo. Eficiente para planificación de trayectorias individuales, pero escalabilidad limitada con múltiples UAVs y horizontes largos [3].
- **RRT (Rapidly-exploring Random Trees):** Exploración probabilística del espacio de configuraciones para generar rutas factibles evitando obstáculos. Útil en entornos complejos, pero no optimiza directamente la minimización de urgencia.
- **Programación lineal entera mixta (MILP):** Formulación exacta del problema como programa matemático. Garantiza optimalidad en instancias pequeñas, pero tiempo de cómputo prohibitivo en casos realistas [4].
- **Heurísticas greedy:** Selección miope del próximo movimiento maximizando reducción de urgencia local. Rápidas y simples, pero con desempeño subóptimo en el largo plazo.

Estos métodos presentan buenos tiempos de cómputo en entornos conocidos, pero sensibilidad a dinámicas no modeladas, incertidumbre y cambios del entorno [3]. Su principal ventaja es la garantía de convergencia y la interpretabilidad de las soluciones generadas.

### 3.4. Metaheurísticas y algoritmos evolutivos

Las metaheurísticas abordan la complejidad del espacio de búsqueda mediante exploración guiada estocástica:

- **Algoritmos genéticos (GA):** Codificación de rutas como cromosomas, evolución mediante selección, cruce y mutación. Permiten optimización multiobjetivo (urgencia, energía, tiempo) y exploración amplia del espacio de soluciones [5].

- **Optimización por enjambre de partículas (PSO):** Inspirado en comportamiento social de aves, ajusta soluciones candidatas mediante influencia del mejor global y local. Convergencia rápida en espacios continuos, requiere discretización cuidadosa.
- **Colonia de hormigas (ACO):** Construcción probabilística de rutas guiada por feromonas artificiales. Efectivo para problemas de tipo TSP y cobertura, con capacidad de balance entre exploración y explotación.
- **Búsqueda tabú y recocido simulado:** Métodos de búsqueda local con mecanismos de escape de óptimos locales. Útiles para refinamiento de soluciones iniciales generadas por heurísticas constructivas.

Las metaheurísticas permiten manejar restricciones complejas y múltiples objetivos simultáneos (cobertura, colisiones, energía) en espacios de búsqueda amplios [5]. Su limitación principal es la falta de garantías de optimalidad y la necesidad de ajuste extensivo de parámetros.

### 3.5. Aprendizaje por refuerzo y métodos híbridos

Los enfoques de aprendizaje representan la tendencia más reciente, permitiendo adaptación online y manejo de incertidumbre:

- **Q-learning y SARSA:** Aprendizaje de políticas de navegación mediante prueba y error. Aplicables a espacios de estados discretos pequeños, escalabilidad limitada.
- **Deep Reinforcement Learning (DRL):** Aproximación de funciones de valor mediante redes neuronales profundas (DQN, A3C, PPO). Se ha demostrado que DRL puede sostener cobertura persistente en regiones con dinámica incierta, aprendiendo políticas de coordinación implícitas [6].
- **Multi-Agent Reinforcement Learning (MARL):** Extensión de DRL a sistemas multi-agente con aprendizaje cooperativo o competitivo. Permite emergencia de comportamientos coordinados complejos sin planificación centralizada explícita.
- **Métodos híbridos:** Integración de planificación clásica con aprendizaje. Por ejemplo, uso de A\* para planificación de rutas seguras combinado con RL para selección adaptativa de objetivos. Estos enfoques combinan garantías de seguridad de métodos clásicos con adaptabilidad del aprendizaje [2].

El aprendizaje por refuerzo profundo aporta adaptabilidad en tiempo real frente a incertidumbre y cambios del entorno, pero demanda grandes volúmenes de datos de entrenamiento (simulación extensiva o experiencia real), presenta desafíos de convergencia y generalización, y requiere validación cuidadosa antes de despliegue operacional [6].

### 3.6. Tendencias actuales

Las líneas recientes de investigación convergen en:

- **Coordinación multi-agente escalable:** Desarrollo de algoritmos MARL/DRL para asignación distribuida de tareas y rutas robustas en enjambres de decenas o cientos de UAVs. Énfasis en protocolos de comunicación eficientes y tolerancia a pérdida de conectividad [6].
- **Gestión energética inteligente:** Planificación explícita de recargas con estaciones móviles (UGVs cargadores), puntos de intercambio de baterías, o recarga inalámbrica. Modelos predictivos de consumo considerando condiciones atmosféricas y perfiles de misión garantizan persistencia indefinida [2, 5].

- **Resiliencia y seguridad:** Implementación de consenso distribuido, detección de fallas, reconfiguración automática ante pérdida de UAVs, y protección contra ataques adversariales (spoofing GPS, jamming de comunicaciones) [1].
- **Métricas operativas avanzadas:** Incorporación de Age of Information (AoI) para cuantificar frescura de datos de vigilancia, optimización de trade-off entre cobertura espacial y frecuencia temporal de visitas, y priorización dinámica basada en eventos detectados.
- **Validación con datos reales:** Transición de simulación a experimentos de campo con flotillas reales, incorporando modelos de viento, obstáculos dinámicos (pájaros, otros drones), restricciones regulatorias (zonas de exclusión aérea) y condiciones meteorológicas adversas.

En conjunto, la tendencia es pasar de rutas estáticas planificadas offline hacia *sistemas cognitivos* que aprenden, se reconfiguran y optimizan continuamente para mantener la vigilancia con eficiencia energética, robustez ante fallas y adaptabilidad a condiciones operacionales cambiantes [6, 2].

La evolución histórica ha sido desde coberturas deterministas y heurísticas hacia coordinación distribuida con aprendizaje y conciencia energética, integrando múltiples paradigmas (optimización, heurísticas, aprendizaje) para enfrentar incertidumbre, restricciones de comunicación, limitaciones de batería y requisitos de operación autónoma en tiempo real.

## 4. Modelo Matemático

Esta sección presenta dos modelos matemáticos para el problema de vigilancia persistente con UAVs: un modelo base de minimización de urgencia acumulada [3] y una variación que incorpora restricciones energéticas con estaciones de recarga [4].

### 4.1. Modelo Base: Minimización de Urgencia Acumulada

El modelo base aborda el problema de planificación de rutas para múltiples UAVs sobre una grilla bidimensional con el objetivo de minimizar la urgencia acumulada total, considerando restricciones de movimiento, colisión y zonas prohibidas [3].

#### 4.1.1. Conjuntos y Parámetros

##### Conjuntos:

- $G = \{1, 2, \dots, n_x\} \times \{1, 2, \dots, n_y\}$ : Conjunto de celdas de la grilla 2D
- $U = \{1, 2, \dots, k\}$ : Conjunto de UAVs disponibles
- $T = \{0, 1, 2, \dots, T_{max}\}$ : Horizonte temporal de la misión
- $V \subseteq G$ : Conjunto de bases logísticas
- $O \subseteq G$ : Conjunto de obstáculos y zonas prohibidas
- $S = G \setminus O$ : Conjunto de celdas transitables (espacio de estados)
- $N(i, j)$ : Vecindario de 8-conectividad de la celda  $(i, j)$ , incluyendo la celda misma

##### Parámetros:

- $w_{ij} \in \mathbb{R}^+$ : Peso de acumulación de urgencia de la celda  $(i, j)$
- $d((i, j), (i', j'))$ : Distancia entre celdas  $(i, j)$  y  $(i', j')$
- $T_{max}$ : Ventana temporal máxima de operación

#### 4.1.2. Variables de Decisión

- $x_{ijt}^u \in \{0, 1\}$ : Variable binaria que vale 1 si el UAV  $u$  está en la celda  $(i, j)$  en el tiempo  $t$ , 0 en otro caso
- $a_{ijt} \in \mathbb{Z}^+$ : Urgencia acumulada de la celda  $(i, j)$  en el tiempo  $t$
- $\tau_{ij} \in \mathbb{Z}^+$ : Tiempo transcurrido desde la última visita a la celda  $(i, j)$

#### 4.1.3. Función Objetivo

La función objetivo busca **minimizar** la urgencia acumulada total sobre todas las celdas y todos los instantes de tiempo:

$$\min Z = \sum_{t=0}^{T_{max}} \sum_{(i,j) \in S} w_{ij} \cdot a_{ijt} \quad (1)$$

#### 4.1.4. Restricciones

- 1. Inicialización de UAVs en bases:** Cada UAV debe iniciar en alguna base logística [4]:

$$\sum_{(i,j) \in V} x_{ij0}^u = 1, \quad \forall u \in U \quad (2)$$

- 2. Conservación de flujo:** Cada UAV ocupa exactamente una celda en cada instante [4]:

$$\sum_{(i,j) \in S} x_{ijt}^u = 1, \quad \forall u \in U, \forall t \in T \quad (3)$$

- 3. Movimiento válido:** Los UAVs solo pueden moverse a celdas vecinas (8-conectividad) o permanecer estacionarios:

$$x_{i'j't}^u \leq \sum_{(i,j) \in N(i',j') \cap S} x_{ijt-1}^u, \quad \forall u \in U, \forall (i',j') \in S, \forall t \in T \setminus \{0\} \quad (4)$$

- 4. Evitación de colisiones:** Dos UAVs no pueden ocupar la misma celda simultáneamente, excepto en bases:

$$\sum_{u \in U} x_{ijt}^u \leq 1, \quad \forall (i,j) \in S \setminus V, \forall t \in T \quad (5)$$

- 5. Prohibición de obstáculos:** Los UAVs no pueden sobrevolar obstáculos:

$$x_{ijt}^u = 0, \quad \forall u \in U, \forall (i,j) \in O, \forall t \in T \quad (6)$$

- 6. Dinámica de urgencia:** La urgencia acumula según el peso de la celda cuando no es visitada y se reinicia al ser visitada:

$$a_{ijt} = \begin{cases} 0, & \text{si } \sum_{u \in U} x_{ijt}^u \geq 1 \\ a_{ij,t-1} + w_{ij}, & \text{en otro caso} \end{cases} \quad \forall (i,j) \in S, \forall t \in T \setminus \{0\} \quad (7)$$

Con condición inicial:

$$a_{ij0} = 0, \quad \forall (i,j) \in S \quad (8)$$

- 7. Naturaleza de las variables:**

$$x_{ijt}^u \in \{0, 1\}, \quad a_{ijt} \in \mathbb{Z}^+ \cup \{0\} \quad (9)$$

#### 4.1.5. Espacio de Búsqueda

El espacio de búsqueda del modelo está determinado por todas las asignaciones válidas de posiciones a UAVs a lo largo del horizonte temporal. Cada solución  $\mathbf{x} = (x_{ijt}^u)_{u \in U, (i,j) \in S, t \in T}$  representa una trayectoria completa para todos los UAVs, y el espacio está acotado por:

$$|\Omega| \leq |S|^{k \cdot (T_{max} + 1)} \quad (10)$$

donde cada UAV puede ocupar cualquiera de las  $|S|$  celdas transitables en cada uno de los  $T_{max} + 1$  instantes de tiempo. Las restricciones de movimiento, colisiones y obstáculos reducen significativamente este espacio factible.

### 4.2. Variación: Modelo con Restricciones Energéticas

Esta variación incorpora las restricciones de energía limitada de los UAVs y la capacidad de recarga en bases logísticas, siguiendo el enfoque propuesto por Scherer y Rinner [4]. El modelo extiende el modelo base añadiendo gestión explícita de la batería y garantías de persistencia mediante caminos seguros.

#### 4.2.1. Parámetros Adicionales

- $E_u \in \mathbb{Z}^+$ : Capacidad energética máxima del UAV  $u$  (medida en pasos de tiempo)
- $c_{mov} \in \mathbb{R}^+$ : Consumo energético por movimiento o permanencia (asumido constante  $c_{mov} = 1$ )
- $r \in \mathbb{R}^+$ : Tasa de recarga en bases logísticas

#### 4.2.2. Variables Adicionales

- $e_t^u \in \mathbb{R}^+$ : Energía remanente del UAV  $u$  en el tiempo  $t$
- $y_t^u \in \{0, 1\}$ : Variable binaria que vale 1 si el UAV  $u$  está recargando en el tiempo  $t$

#### 4.2.3. Función Objetivo Extendida

La función objetivo permanece como la minimización de urgencia, pero ahora sujeta a viabilidad energética:

$$\min Z = \sum_{t=0}^{T_{max}} \sum_{(i,j) \in S} w_{ij} \cdot a_{ijt} \quad (11)$$

#### 4.2.4. Restricciones Adicionales

**8. Dinámica energética:** La energía disminuye con el movimiento y aumenta durante la recarga:

$$e_t^u = e_{t-1}^u - c_{mov} + r \cdot y_t^u, \quad \forall u \in U, \forall t \in T \setminus \{0\} \quad (12)$$

**9. Límites de energía:**

$$0 \leq e_t^u \leq E_u, \quad \forall u \in U, \forall t \in T \quad (13)$$

**10. Recarga solo en bases:** Un UAV solo puede recargar cuando está en una base logística:

$$y_t^u \leq \sum_{(i,j) \in V} x_{ijt}^u, \quad \forall u \in U, \forall t \in T \quad (14)$$

**11. Garantía de camino seguro:** En todo momento, cada UAV debe tener suficiente energía para alcanzar alguna base logística:

$$e_t^u \geq \min_{(i,j) \in V} d(\text{pos}(u, t), (i, j)), \quad \forall u \in U, \forall t \in T \quad (15)$$

donde  $\text{pos}(u, t) = (i, j)$  si  $x_{ijt}^u = 1$ .

**12. Condición inicial energética:**

$$e_0^u = E_u, \quad \forall u \in U \quad (16)$$

#### 4.2.5. Espacio de Búsqueda Extendido

El espacio de búsqueda de esta variación es sustancialmente más complejo debido a la dimensión energética adicional. Cada estado del sistema ahora incluye tanto la posición como la energía remanente de cada UAV:

$$\mathcal{S} = \{(\mathbf{p}, \mathbf{e}) : \mathbf{p} \in S^k, \mathbf{e} \in [0, E_u]^k\} \quad (17)$$

donde  $\mathbf{p}$  representa las posiciones de los  $k$  UAVs y  $\mathbf{e}$  sus energías remanentes. Las restricciones de caminos seguros reducen dramáticamente el espacio factible al prohibir estados desde los cuales un UAV no puede alcanzar una base antes de agotar su energía. Este modelo es NP-hard y requiere heurísticas o metaheurísticas para su resolución en instancias de tamaño realista.

## 5. Representación

La representación de soluciones es fundamental para el éxito del algoritmo evolutivo implementado. Esta sección describe las estructuras de datos utilizadas, su justificación y su correspondencia con el modelo matemático formalizado.

### 5.1. Representación del Cromosoma

Cada individuo de la población (cromosoma) codifica una solución completa al problema de vigilancia persistente. La representación se compone de dos componentes principales:

#### 5.1.1. Asignación de Bases Iniciales

Se utiliza un vector de enteros  $\mathbf{b} = (b_1, b_2, \dots, b_k)$  donde cada elemento  $b_u \in \{0, 1, \dots, |V|-1\}$  representa el índice de la base logística desde la cual despegue el UAV  $u$ . Esta representación:

- **Mapea directamente** con la restricción de inicialización del modelo matemático (ecuación 2): cada UAV debe iniciar en exactamente una base  $v \in V$ .
- **Es compacta:** requiere solo  $k$  enteros, reduciendo el espacio de memoria.
- **Facilita el elitismo:** permite preservar asignaciones exitosas entre generaciones.

### 5.1.2. Plan de Acciones Temporal

Se utiliza una matriz de acciones  $\mathbf{A} \in \mathbb{Z}^{k \times T}$  donde cada elemento  $a_{u,t} \in \{0, 1, 2, \dots, 8\}$  codifica la acción del UAV  $u$  en el instante  $t$ . La codificación de acciones es:

- $a = 0$ : Permanecer estacionario
- $a = 1$ : Mover Norte (N)
- $a = 2$ : Mover Noreste (NE)
- $a = 3$ : Mover Este (E)
- $a = 4$ : Mover Sureste (SE)
- $a = 5$ : Mover Sur (S)
- $a = 6$ : Mover Suroeste (SW)
- $a = 7$ : Mover Oeste (W)
- $a = 8$ : Mover Noroeste (NW)

Esta codificación se eligió porque:

- **Corresponde al vecindario de 8-conectividad** definido en el modelo matemático  $N(i, j)$ , permitiendo movimientos a las ocho celdas adyacentes más la opción de permanecer.
- **Es eficiente computacionalmente**: aplicar una acción requiere solo una operación switch/case con aritmética de coordenadas.
- **Simplifica operadores genéticos**: el cruce y mutación operan directamente sobre enteros en un rango pequeño, sin necesidad de decodificación compleja.
- **Evita restricciones de representación**: cualquier secuencia de acciones es sintácticamente válida, delegando la verificación de restricciones (obstáculos, colisiones, límites de grilla) a la función de fitness.

## 5.2. Estructuras de Datos Auxiliares

### 5.2.1. Estado de Urgencias

Durante la evaluación de fitness, se mantiene un mapa asociativo (tabla hash)  $\mathcal{U} : \text{Coordenada} \rightarrow \mathbb{R}^+$  que almacena la urgencia acumulada actual de cada celda con tasa de urgencia no nula. Esta estructura:

- **Implementa directamente** la dinámica de urgencia del modelo (ecuación 7): acumula  $w_{ij}$  cuando la celda no es visitada y se reinicia a cero cuando es visitada.
- **Usa solo memoria necesaria**: solo almacena celdas con  $w_{ij} > 0$ , evitando matrices dispersas.
- **Permite acceso eficiente**: operaciones de consulta y actualización en tiempo  $O(1)$  promedio.

### 5.2.2. Posiciones de UAVs

Se utiliza un vector  $\mathbf{p} = (p_1, p_2, \dots, p_k)$  donde  $p_u = (i, j)$  representa la posición actual del UAV  $u$  en la grilla. Este vector:

- **Corresponde a las variables de decisión**  $x_{ijt}^u$  del modelo: en cada instante  $t$ , exactamente una coordenada  $(i, j)$  tiene  $x_{ijt}^u = 1$  para cada UAV  $u$ .
- **Facilita la verificación de colisiones**: mediante un conjunto (set) temporal se detectan posiciones duplicadas en tiempo  $O(k \log k)$ .
- **Es actualizado iterativamente**: aplicando sucesivamente las acciones codificadas en el cromosoma.

### 5.3. Justificación de la Representación Elegida

La representación adoptada presenta las siguientes ventajas para el problema de vigilancia persistente con UAVs:

1. **Completitud**: Cualquier solución factible del modelo matemático puede ser codificada. La representación cubre todo el espacio de búsqueda definido por  $\Omega$ .
2. **Flexibilidad para restricciones**: Las violaciones de restricciones (obstáculos, colisiones, límites de grilla) se manejan mediante penalización gradual en la función de fitness, evitando complejidad en los operadores genéticos.
3. **Escalabilidad**: La representación escala linealmente con el número de UAVs ( $k$ ) y el horizonte temporal ( $T$ ). Para  $k = 5$  y  $T = 50$ , el cromosoma requiere solo 255 enteros (5 bases + 250 acciones).
4. **Operabilidad genética**: Los operadores de cruce y mutación operan directamente sobre secuencias de enteros, sin necesidad de reparación o decodificación compleja. El cruce temporal (un punto en el eje  $t$ ) preserva coherencia local en las trayectorias.
5. **Eficiencia de evaluación**: La simulación tick-por-tick permite calcular fitness en tiempo  $O(T \cdot k \cdot \log k)$ , donde el factor logarítmico proviene de la detección de colisiones mediante conjuntos ordenados.

### 5.4. Correspondencia con el Modelo Matemático

La siguiente tabla resume la correspondencia directa entre el modelo matemático formalizado y las estructuras de datos implementadas:

| Modelo Matemático   | Estructura de Datos   |
|---|---|
| Variables $x_{ijt}^u \in \{0, 1\}$ (posición del UAV)           | Vector $\mathbf{p}$ de coordenadas actuales, actualizado según acciones codificadas |
| Urgencia acumulada $a_{ijt} \in \mathbb{Z}^+$                   | Mapa $\mathcal{U} : \text{Coordenada} \rightarrow \text{double}$                    |
| Conjunto de bases $V$   | Vector de coordenadas <b>bases</b>  |
| Asignación inicial a bases (ecuación 2)                         | Vector de índices $\mathbf{b} = (b_1, \dots, b_k)$                                  |
| Secuencia de movimientos (restricción 3)                        | Matriz de acciones $\mathbf{A} \in \{0, \dots, 8\}^{k \times T}$                    |
| Dinámica de urgencia (ecuación 7)                               | Actualización condicional en $\mathcal{U}$ : acumular o resetear según visitas      |
| Función objetivo $Z = \sum_t \sum_{(i,j)} w_{ij} \cdot a_{ijt}$ | Variable <b>urgencia_acumulada_total</b> sumada en cada tick                        |

Esta correspondencia directa garantiza que la implementación es una instanciación fiel del modelo teórico, permitiendo que las soluciones generadas por el algoritmo evolutivo sean válidas para el problema original de minimización de urgencia acumulada.

## **6. Descripción del algoritmo**

Cómo fue implementando, interesa la implementación más que el algoritmo genérico, es decir, si se tiene que implementar SA, lo que se espera es que se explique en pseudo código la estructura general y en párrafo explicativo cada parte como fue implementada para su caso particular, si se utilizan operadores se debe explicar por qué se utilizó ese operador, si fuera el caso de una técnica completa, si se utiliza recursión o no, etc. Use diagramas para mostrar la estructura general del algoritmo, diagramas de flujo de movimientos, esquemas, etc. En este punto no se espera que se incluya código, eso va aparte en la entrega del código fuente.

## **7. Experimentos**

Se necesita saber cómo se hicieron los experimentos para testear los resultados del algoritmo (metodología usada, entorno de experimentación, etc.), cuáles son, cómo se definen y cómo se obtienen parámetros del algoritmo, como los fueron modificando, describir las instancias que se usaron (complejidad, estructura, etc), criterio de término (si aplica). Debe comparar su algoritmo con el estado del arte, además de comparar ejecuciones con distintas especificaciones de su mismo algoritmo (Ejm. el valor del parámetro  $x$  siendo 0.1 vs 0.5 vs 0.9). Describir cantidad de ejecuciones usando semillas distintas para generar estadísticas.

## **8. Resultados**

Que fue lo que se logró con la experimentación, incluir tablas y parámetros, gráficos (por ejm boxplot), lo más explicativo posible. En los resultados se espera que concluya cuál fue el rendimiento del algoritmo con los experimentos detallados en la sección anterior, y compare las diferencias entre configuraciones distintas de los experimentos. Analizar los resultados obtenidos y concluir acerca de aspectos del algoritmo y/o de la complejidad de las instancias, o acerca de características relacionadas con su implementación.

## **9. Conclusiones**

Las técnicas revisadas para el problema de vigilancia persistente con UAVs resuelven variantes relacionadas pero no idénticas del problema. Los enfoques clásicos y heurísticos se centran en planificación rápida de rutas en entornos conocidos y estáticos. Las metaheurísticas exploran espacios de búsqueda más complejos y manejan múltiples objetivos simultáneamente, aunque requieren mayor costo computacional. Los métodos de aprendizaje por refuerzo y enfoques híbridos aportan adaptabilidad en tiempo real frente a incertidumbre y cambios del entorno, pero demandan grandes volúmenes de datos de entrenamiento y presentan desafíos de convergencia y generalización.

Todas las técnicas comparten el objetivo fundamental de minimizar la urgencia acumulada (o maximizar la cobertura), pero difieren significativamente en sus supuestos operacionales. Los métodos clásicos asumen mapas completamente conocidos y entornos estáticos; las metaheurísticas permiten optimización multiobjetivo pero operan offline; y los enfoques de aprendizaje se diseñan para adaptación online pero requieren modelos de simulación precisos para entrenamiento [3, 4].

Las principales limitaciones incluyen escalabilidad en horizontes temporales largos y flotas grandes, sensibilidad a incertidumbre (ruido de sensores, fallas de comunicación, obstáculos inesperados), y supuestos idealizados como sincronización perfecta y conocimiento completo del entorno.

Las estrategias más prometedoras integran coordinación distribuida con aprendizaje online y modelos explícitos de energía y comunicación [4]. Los enfoques híbridos que combinan planificación clásica con aprendizaje por refuerzo muestran mejores resultados en entornos dinámicos. La incorporación de estaciones de recarga móviles o puntos de intercambio de baterías permite operación persistente real, superando la limitación crítica de autonomía energética [5].

Se proponen las siguientes líneas de investigación futura:

- **Optimización multiobjetivo robusta:** Extender el modelo base para balancear urgencia, consumo energético y riesgo operacional, utilizando técnicas de optimización robusta [5].
- **Coordinación distribuida resiliente:** Desarrollar protocolos de coordinación entre varios UAVs considerando una pérdida intermitente de comunicación y fallas individuales.
- **Modelado realista de consumo energético:** Incorporar modelos de consumo que consideren condiciones atmosféricas (viento, temperatura), maniobras y cargas, junto con estrategias de recarga.

Estas direcciones fortalecen la aplicabilidad práctica del problema y contribuyen a cerrar la brecha entre los modelos teóricos y las demandas operacionales reales de sistemas de vigilancia con UAVs.

## Referencias

- [1] Noa Agmon, Gal A. Kaminka, and Sarit Kraus. Multi-robot adversarial patrolling: Facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42:887–916, 2011.
- [2] Xiaoshan Lin, Yasin Yazicioglu, and Derya Aksaray. Robust planning for persistent surveillance with energy-constrained uavs and mobile charging stations. *IEEE Robotics and Automation Letters*, 7:1–1, 04 2022.
- [3] Nikhil Nigam. The multiple unmanned air vehicle persistent surveillance problem: A review. *Machines*, 2(1):13–72, 2014.
- [4] Jürgen Scherer and Bernhard Rinner. Persistent multi-uav surveillance with energy and communication constraints. In *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, page 1225–1230. IEEE Press, 2016.
- [5] Sepehr Seyed, Yasin Yazıcıoğlu, and Derya Aksaray. Persistent surveillance with energy-constrained uavs and mobile charging stations. *IFAC-PapersOnLine*, 52(20):193–198, 2019. 8th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2019.
- [6] Zhaomei Sun, Nan Wang, Hong Lin, and Xiaojun Zhou. Persistent coverage of uavs based on deep reinforcement learning with wonderful life utility. *Neurocomputing*, 521, 12 2022.