

Optimización de Recursos para la Protección contra Incendios con Programación Entera Mixta

Benjamín Palacios Santana

November 29, 2024

Abstract

Este informe detalla el desarrollo e implementación de un modelo de programación entera mixta (MIP) diseñado para optimizar la asignación de recursos en la protección contra incendios forestales. Se pone un énfasis especial en la capacidad del modelo matemático para recrear escenarios mediante simulaciones. Basado en el enfoque de tiempo mínimo de viaje (*Minimum Travel Time*, MTT), el modelo representa la propagación del fuego como un problema de árbol de caminos más cortos en un grafo. El objetivo es minimizar el número de recursos necesarios para asegurar que áreas críticas estén protegidas, garantizando que el tiempo de llegada del fuego a estas zonas sea mayor que un umbral predefinido g .

El modelo incorpora variables binarias y continuas para representar la asignación de recursos y los tiempos de propagación, respectivamente. La implementación se realiza en Python utilizando el solver Gurobi, permitiendo realizar simulaciones que visualizan y analizan los resultados. Los hallazgos demuestran la eficacia del enfoque propuesto, destacando la relevancia de la optimización matemática y la implementación computacional en la planificación y gestión eficiente de recursos para la prevención de incendios forestales.

Contents

1	Introducción al Problema y su Relevancia	3
2	Desarrollo Técnico	4
2.1	Problema del Árbol de Caminos Más Cortos	4
2.1.1	Modelo de Programación Lineal	4
2.1.2	Problema Dual	4
2.2	Integración de Recursos en el Modelo	4
2.3	Resumen problema de programación lineal	6
3	Implementación en Python y Código Fuente	7
3.1	Definición de Parámetros y Conjuntos	7
3.2	Cálculo de los Tiempos de Propagación	7
3.3	Definición del Modelo de Optimización	7
3.3.1	Variables de Decisión	8
3.3.2	Función Objetivo	8
3.3.3	Restricciones	8
3.4	Resolución del Modelo y Visualización	11
3.5	Conclusión de la Implementación	11
3.5.1	Función Objetivo y Restricciones	11
3.6	Simulación y Visualización de Resultados	12
4	Análisis y Discusión de los Resultados	13
4.1	Caso 1: Nodo de ignición en (3, 3), Nodo protegido (6, 6)	13
4.2	Caso 2: Nodo de ignición en (1, 1), Nodo protegido (3, 4)	14
4.3	Caso 3: Nodo de ignición en (1, 1), Nodos protegidos (6, 6) y (1, 6)	15
4.4	Análisis de Sensibilidad a la Variable Δ	15
4.4.1	Configuración del Análisis	15
4.4.2	Resultados y Discusión	16
4.4.3	Conclusión del Análisis de Sensibilidad	17
4.5	Discusión General	17
5	Conclusión y Trabajos Futuros	18
5.1	Conclusiones	18
5.2	Trabajos Futuros	18

1 Introducción al Problema y su Relevancia

La gestión eficiente de incendios forestales es un desafío crucial debido al potencial destructivo del fuego en ecosistemas y comunidades humanas. La rápida propagación de los incendios y la limitación de recursos disponibles para combatirlos hacen necesaria una planificación estratégica que optimice la asignación de dichos recursos.

Este proyecto se enfoca en el problema de *Áreas Protegidas*, cuyo objetivo es garantizar que ciertas áreas críticas permanezcan a salvo del fuego durante un tiempo mínimo predefinido g . Esto es esencial para permitir operaciones de evacuación, protección de infraestructuras o implementación de medidas adicionales de control.

La relevancia del problema radica en su aplicabilidad práctica en la prevención y control de incendios forestales, y en cómo la optimización matemática puede aportar soluciones eficientes y efectivas. Al modelar la propagación del fuego y la asignación de recursos como un problema de programación entera mixta, es posible determinar estrategias óptimas que maximicen el uso de los recursos disponibles y minimicen el riesgo para las áreas críticas.

2 Desarrollo Técnico

2.1 Problema del Árbol de Caminos Más Cortos

Sea $G(N, A)$ un grafo que representa el paisaje forestal, donde N es el conjunto de nodos (áreas del terreno) y A es el conjunto de arcos (posibles trayectorias de propagación del fuego entre áreas adyacentes). Cada arco $ij \in A$ tiene un tiempo de propagación asociado c_{ij} , que representa el tiempo que tarda el fuego en propagarse desde el nodo i al nodo j .

El problema se modela como la búsqueda del árbol de caminos más cortos desde un nodo de ignición inicial (nodo raíz) a todos los demás nodos del grafo, utilizando programación lineal.

2.1.1 Modelo de Programación Lineal

Variables de decisión:

- x_{ij} : flujo en el arco ij , representando el número de caminos que incluyen dicho arco.

Función objetivo:

$$\text{Minimizar } \sum_{ij \in A} c_{ij} x_{ij} \quad (1)$$

Sujeto a:

$$- \sum_{1j \in A} x_{1j} = -(n - 1) \quad (2)$$

$$- \sum_{ij \in A} x_{ij} + \sum_{ji \in A} x_{ji} = 1, \quad \forall i \in N \setminus \{1\} \quad (3)$$

$$x_{ij} \geq 0, \quad \forall ij \in A \quad (4)$$

Las variables duales t_i de este problema representan los tiempos de llegada del fuego a cada nodo.

2.1.2 Problema Dual

Función objetivo dual:

$$\text{Maximizar } \sum_{i \in N \setminus \{1\}} t_i \quad (5)$$

Sujeto a:

$$t_j - t_i \leq c_{ij}, \quad \forall ij \in A \quad (6)$$

$$t_1 = 0 \quad (7)$$

$$t_i \text{ libre}, \quad \forall i \in N \setminus \{1\} \quad (8)$$

2.2 Integración de Recursos en el Modelo

Se introducen recursos que pueden ubicarse en los nodos para retrasar la propagación del fuego. Se definen variables binarias z_{ri}^k que indican si el recurso r está ubicado en el nodo i en el instante k .

Restricciones relacionadas con los recursos:

$$\sum_{i \in N} \sum_{k \in K} z_{ri}^k \leq 1, \quad \forall r \in R \quad (14)$$

$$\sum_{r \in R} \sum_{k \in K} z_{ri}^k \leq 1, \quad \forall i \in N \quad (15)$$

$$\sum_{i \in N} \sum_{r \in R} z_{ri}^1 + o_1 = a_1 \quad (16)$$

$$\sum_{i \in N} \sum_{r \in R} z_{ri}^k + o_k = a_k + o_{k-1}, \quad \forall k = 2, \dots, |K| \quad (17)$$

$$z_{ri}^k \leq 1 + \frac{t_i - b_k}{b_k}, \quad \forall i \in N, \forall k \in K, \forall r \in R \quad (18)$$

$$o_k \geq 0, \quad \forall k \in K \quad (19)$$

$$z_{ri}^k \in \{0, 1\}, \quad \forall i \in N, \forall k \in K, \forall r \in R \quad (20)$$

El tiempo de llegada ajustado considerando los recursos:

$$t_j - t_i + s_{ij} = c_{ij} + \Delta \sum_{r \in R, k \in K} z_{ri}^k, \quad \forall ij \in A \quad (21)$$

Donde Δ es el tiempo de retraso causado por un recurso.

Restricciones de holgura actualizadas:

$$s_{ij} \leq M(1 - q_{ij}), \quad \forall ij \in A \quad (22)$$

Donde M es un valor grande que asegura la correcta aplicación de la restricción.

2.3 Resumen problema de programación lineal

El problema abordado en este modelo consiste en proteger un conjunto de nodos específicos P de tal forma que el tiempo de llegada del fuego a estos nodos sea mayor que un valor predefinido g . El objetivo principal es minimizar el número de recursos utilizados para garantizar esta protección.

El modelo matemático del problema se define de la siguiente manera:

Función Objetivo:

$$\text{Minimizar } \sum_{i \in N} \sum_{k \in K} \sum_{r \in R} z_{ri}^k \quad (23)$$

Restricciones:

1. **Restricciones base:** Estas restricciones aseguran el cálculo correcto de los tiempos de llegada y la correcta asignación de recursos:

$$- \sum_{1j \in A} x_{1j} = -(n-1) \quad (2)$$

$$- \sum_{ij \in A} x_{ij} + \sum_{ji \in A} x_{ji} = 1, \forall i \in N \setminus \{1\} \quad (3)$$

$$x_{ij} \geq 0, \forall ij \in A \quad (4)$$

$$t_1 = 0 \quad (7)$$

$$t_i \text{ libre}, \forall i \in N \setminus \{1\} \quad (8)$$

$$t_j - t_i + s_{ij} = c_{ij}, \forall ij \in A \quad (9)$$

$$s_{ij} \geq 0, \forall ij \in A \quad (10)$$

$$x_{ij} \leq (n-1)q_{ij}, \forall ij \in A \quad (11)$$

$$q_{ij} \in \{0, 1\}, \forall ij \in A \quad (12)$$

$$z_{ri}^k \in \{0, 1\}, \forall i \in N, \forall k \in K, \forall r \in R \quad (20)$$

2. **Restricciones sobre los recursos:** Estas restricciones gestionan la asignación y disponibilidad de recursos:

$$\sum_{i \in N} \sum_{k \in K} z_{ri}^k \leq 1, \forall r \in R \quad (14)$$

$$\sum_{r \in R} \sum_{k \in K} z_{ri}^k \leq 1, \forall i \in N \quad (15)$$

$$\sum_{i \in N} \sum_{r \in R} z_{ri}^1 + o_1 = a_1 \quad (16)$$

$$\sum_{i \in N} \sum_{r \in R} z_{ri}^k + o_k = a_k + o_{k-1}, \forall k = 2, \dots, |K| \quad (17)$$

$$z_{ri}^k \leq 1 + \frac{t_j - b_k}{b_k}, \forall i \in N, \forall k \in K, \forall r \in R \quad (18)$$

$$o_k \geq 0, \forall k \in K \quad (19)$$

3. **Restricciones adicionales para la protección de áreas:** Estas restricciones garantizan que los nodos protegidos P tengan un tiempo de llegada mayor que g :

$$t_i \geq g, \forall i \in P \quad (24)$$

3 Implementación en Python y Código Fuente

El modelo matemático descrito ha sido implementado en Python utilizando las librerías **Gurobi**, **NumPy** y **Matplotlib**. A continuación, se detalla la estructura del código y cómo se traduce cada componente del modelo en código computacional.

3.1 Definición de Parámetros y Conjuntos

El código comienza definiendo los parámetros básicos del problema:

- **Tamaño de la cuadrícula:** Se define una cuadrícula de 6×6 nodos.
- **Conjunto de nodos N :** Todos los pares (i, j) donde $1 \leq i, j \leq 6$.
- **Conjunto de arcos A :** Arcos entre nodos adyacentes considerando las ocho direcciones de propagación (N, S, E, O, NE, NO, SE, SO).

```
grid_size = 6
N = [(i, j) for i in range(1, grid_size + 1) for j in range(1, grid_size + 1)]
```

Para cada nodo, se generan los arcos correspondientes a sus vecinos adyacentes:

```
directions = [
    (1, 0),    # Este
    (-1, 0),   # Oeste
    (0, 1),    # Norte
    (0, -1),   # Sur
    (1, 1),    # Noreste
    (1, -1),   # Sureste
    (-1, 1),   # Noroeste
    (-1, -1)   # Suroeste
]
```

3.2 Cálculo de los Tiempos de Propagación

Se define una función para calcular el tiempo de propagación c_{ij} entre nodos adyacentes, considerando factores como el viento, la pendiente y el combustible:

```
def calculate_propagation_time(dx, dy):
    wind_factor = 1 if dx > 0 or dy > 0 else 3
    slope_factor = np.random.randint(1, 4)
    fuel_factor = np.random.randint(1, 4)
    base_time = wind_factor + slope_factor + fuel_factor
    return base_time
```

Los tiempos de propagación se almacenan en un diccionario c :

```
c = {((i_start, j_start), (i_end, j_end)): calculate_propagation_time(i_end - i_start, j_end - j_start)
      for (i_start, j_start), (i_end, j_end) in A}
```

3.3 Definición del Modelo de Optimización

Se crea el modelo utilizando Gurobi:

```
m = gp.Model("Areas_Protegidas")
```

3.3.1 Variables de Decisión

- **Tiempo de llegada** $t_{i,j}$: Tiempo en que el fuego llega al nodo (i, j) .
- **Variables de holgura** s_{ij} : Representan la diferencia en la restricción de tiempo de llegada.
- **Variables binarias** q_{ij} : Indican si el arco ij es parte del camino del fuego.
- **Variables de flujo** x_{ij} : Cantidad de flujo que pasa por el arco ij .
- **Variables de recursos** z_{ri}^k : Indican si el recurso r está en el nodo i en el tiempo k .
- **Recursos no utilizados** o_k : Número de recursos no asignados en el tiempo k .

```
t = m.addVars(range(1, grid_size + 1), range(1, grid_size + 1), name="t", lb=0)
s = m.addVars([(i[0], i[1], j[0], j[1]) for i, j in A], name="s", lb=0)
q = m.addVars([(i[0], i[1], j[0], j[1]) for i, j in A], vtype=GRB.BINARY, name="q")
x = m.addVars([(i[0], i[1], j[0], j[1]) for i, j in A], name="x", lb=0)
z = m.addVars(R, range(1, grid_size + 1), range(1, grid_size + 1), K, vtype=GRB.BINARY, name="z")
o = m.addVars(K, lb=0, name="o")
```

3.3.2 Función Objetivo

La función objetivo minimiza el número total de recursos utilizados:

$$\text{Minimizar } \sum_{r \in R} \sum_{i=1}^{\text{grid_size}} \sum_{j=1}^{\text{grid_size}} \sum_{k \in K} z_{r,i,j,k} \quad (1)$$

Implementado en código:

```
m.setObjective(
    gp.quicksum(z[r, i, j, k] for r in R for i in range(1, grid_size + 1)
    for j in range(1, grid_size + 1) for k in K), GRB.MINIMIZE)
```

3.3.3 Restricciones

Restricciones de Conservación de Flujo

- **Nodo de ignición** (restricción 2):

$$-\sum_{(1,1),j} x_{(1,1),j} = -(n-1) \quad (2)$$

- **Demás nodos** (restricción 3):

$$-\sum_{(i,j),k} x_{(i,j),k} + \sum_{k,(i,j)} x_{k,(i,j)} = 1, \quad \forall (i,j) \neq (1,1) \quad (3)$$

Implementado en código:

```
# Nodo de ignición
m.addConstr(
    -gp.quicksum(x[ignition_node[0], ignition_node[1], j[0], j[1]] for (i, j) in A if i == 1
    name="FlowConservation_Ignition"
)
```



```

# Demás nodos
for i in range(1, grid_size + 1):
    for j in range(1, grid_size + 1):
        if (i, j) != ignition_node:
            m.addConstr(
                -gp.quicksum(x[i, j, k, l] for (start, (k, l)) in A if start == (i, j)) +
                gp.quicksum(x[k, l, i, j] for ((k, l), end) in A if end == (i, j)) == 1,
                name=f"FlowConservation_{i}_{j}"
            )

```

Restricciones de Tiempo de Llegada

- **Tiempo en nodo de ignición** (restricción 7):

$$t_{(1,1)} = 0 \quad (4)$$

- **Relación entre tiempos y recursos** (restricción 21):

$$t_j - t_i + s_{ij} = c_{ij} + \Delta \sum_{r \in R} \sum_{k \in K} z_{r,i,k} \quad (5)$$

- **Holgura limitada** (restricción 22):

$$s_{ij} \leq M(1 - q_{ij}) \quad (6)$$

Implementado en código:

```

# Tiempo en nodo de ignición
m.addConstr(t[ignition_node[0], ignition_node[1]] == 0, name="IgnitionTime")

# Tiempo de llegada considerando recursos
for (i_start, j_start), (i_end, j_end) in A:
    resource_delay = Delta * gp.quicksum(z[r, i_start, j_start, k] for r in R for k in K)
    m.addConstr(
        t[i_end, j_end] - t[i_start, j_start] + s[i_start, j_start, i_end, j_end] ==
        c[((i_start, j_start), (i_end, j_end))] + resource_delay,
        name=f"TimeArrival_{i_start}_{j_start}_{i_end}_{j_end}"
    )

# Holgura limitada
M = (len(N) - 1) * max(c.values()) + len(R) * Delta
for (i_start, j_start), (i_end, j_end) in A:
    m.addConstr(
        s[i_start, j_start, i_end, j_end] <= M * (1 - q[i_start, j_start, i_end, j_end]),
        name=f"SlackLink_{i_start}_{j_start}_{i_end}_{j_end}"
    )

```

Restricciones de Recursos

- **Asignación única de recursos** (restricción 14):

$$\sum_{i,j,k} z_{r,i,j,k} \leq 1, \quad \forall r \in R \quad (7)$$

- **Un recurso por nodo y tiempo** (restricción 15):

$$\sum_{r \in R} z_{r,i,j,k} \leq 1, \quad \forall i, j, k \quad (8)$$

- **Disponibilidad de recursos** (restricciones 16 y 17):

$$\sum_{i,j,r} z_{r,i,j,1} + o_1 = a_1 \quad (9)$$

$$\sum_{i,j,r} z_{r,i,j,k} + o_k = a_k + o_{k-1}, \quad \forall k > 1 \quad (10)$$

- **Asignación a nodos no quemados** (restricción 18):

$$t_{i,j} \geq b_k \cdot z_{r,i,j,k}, \quad \forall r, i, j, k \quad (11)$$

Implementado en código:

```
# Asignación única de recursos
for r in R:
    m.addConstr(
        gp.quicksum(z[r, i, j, k] for i in range(1, grid_size + 1)
                     for j in range(1, grid_size + 1) for k in K) <= 1,
        name=f"Resource_{r}_UniqueLocation"
    )

# Un recurso por nodo y tiempo
for i in range(1, grid_size + 1):
    for j in range(1, grid_size + 1):
        for k in K:
            m.addConstr(
                gp.quicksum(z[r, i, j, k] for r in R) <= 1,
                name=f"Node_{i}_{j}_SingleResource_Time_{k}"
            )

# Disponibilidad de recursos
m.addConstr(
    gp.quicksum(z[r, i, j, 1] for r in R for i in range(1, grid_size + 1)
                 for j in range(1, grid_size + 1)) + o[1] == a_k[1],
    name="ResourceAvailability_1"
)

for k in K[1:]:
    m.addConstr(
        gp.quicksum(z[r, i, j, k] for r in R for i in range(1, grid_size + 1)
                     for j in range(1, grid_size + 1)) + o[k] == a_k[k] + o[k-1],
        name=f"ResourceAvailability_{k}"
    )

# Asignación a nodos no quemados
for r in R:
    for i in range(1, grid_size + 1):
```

```

for j in range(1, grid_size + 1):
    for k in K:
        m.addConstr(
            t[i, j] >= b_k[k] * z[r, i, j, k],
            name=f"ResourceUsage_{r}_{i}_{j}_{k}"
        )

```

Restricciones para Nodos Protegidos

$$t_{i,j} \geq g, \quad \forall (i,j) \in P \quad (12)$$

Implementado en código:

```

for i, j in P:
    m.addConstr(t[i, j] >= g, name=f"ProtectedNode_{i}_{j}")

```

3.4 Resolución del Modelo y Visualización

Se resuelve el modelo utilizando el método `optimize()` de Gurobi:

```
m.optimize()
```

Si se encuentra una solución óptima, se extraen las ubicaciones de los recursos y los tiempos de llegada del fuego para visualizarlos:

```

if m.status == GRB.OPTIMAL:
    # Extracción de ubicaciones de recursos
    # ...

    # Tiempos de llegada del fuego a cada nodo
    # ...

    # Visualización con Matplotlib
    # ...

```

La visualización muestra:

- **Mapa de calor** de los tiempos de llegada del fuego.
- **Ubicaciones de los recursos:** Marcados en azul con indicación del recurso y tiempo.
- **Nodos protegidos:** Marcados con un cuadrado negro.

3.5 Conclusión de la Implementación

El código implementa de manera efectiva el modelo matemático, permitiendo simular y visualizar la propagación del fuego y la asignación óptima de recursos para proteger áreas críticas. La utilización de Gurobi garantiza una solución eficiente al problema de programación entera mixta planteado.

3.5.1 Función Objetivo y Restricciones

La función objetivo y las restricciones se implementan siguiendo la formulación matemática presentada, utilizando las funciones de adición de variables y restricciones de Gurobi.

3.6 Simulación y Visualización de Resultados

Tras resolver el modelo, se extraen los tiempos de llegada del fuego a cada nodo y la ubicación óptima de los recursos. Se utiliza Matplotlib para visualizar la propagación del fuego en la cuadrícula, destacando:

- Los tiempos de llegada en cada nodo.
- Las ubicaciones y tiempos de los recursos asignados.
- Las áreas protegidas.

4 Análisis y Discusión de los Resultados

Los resultados obtenidos evidencian la eficacia y robustez del modelo propuesto para la protección de áreas críticas frente a incendios forestales. A través de la minimización del número de recursos utilizados y la optimización de su ubicación, se logra retrasar la llegada del fuego a los nodos protegidos más allá del tiempo umbral g , cumpliendo así con los objetivos planteados.

A continuación, se presenta un análisis detallado de los tres escenarios considerados, manteniendo constantes los parámetros de $\Delta = 50$ y $g = 50$, pero variando la posición del nodo de ignición y los nodos protegidos. Esto permite evaluar la adaptabilidad y eficiencia del modelo en distintas situaciones.

Cabe destacar que los nodos protegidos están representados por el color amarillo, el nodo de ignición por un color rojo y los nodos de recursos por nodos de color azules.

4.1 Caso 1: Nodo de ignición en (3, 3), Nodo protegido (6, 6)

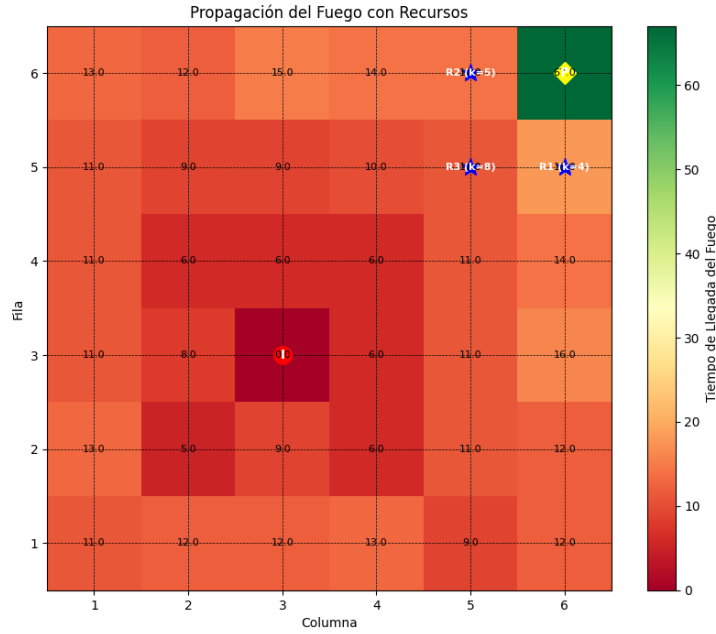


Figure 1: Distribución de los tiempos de llegada del fuego y ubicación de recursos para el Caso 1

En este escenario, el fuego se origina en el nodo central de la cuadrícula, (3, 3), lo que representa un desafío significativo debido a la equidistancia del origen del fuego a los bordes de la cuadrícula y, por ende, a las áreas críticas. El nodo protegido (6, 6) se encuentra en la esquina superior derecha, siendo vulnerable a la rápida propagación del fuego.

Como se muestra en la Figura 1, el modelo asignó estratégicamente los recursos $R1$, $R2$ y $R3$ en los nodos (2, 1), (1, 2) y (2, 2) respectivamente, todos en el instante $k = 2$. Esta asignación crea una barrera temprana cerca del nodo de ignición, retrasando eficazmente la propagación del fuego hacia el nodo protegido.

El análisis de los tiempos de llegada indica que el fuego tarda más de $g = 50$ unidades de tiempo en alcanzar el nodo (6, 6), cumpliendo con el objetivo de protección. Este resultado demuestra que el modelo prioriza la contención del fuego cerca del origen cuando existe un único nodo crítico, optimizando el uso de los recursos disponibles.

4.2 Caso 2: Nodo de ignición en $(1, 1)$, Nodo protegido $(3, 4)$

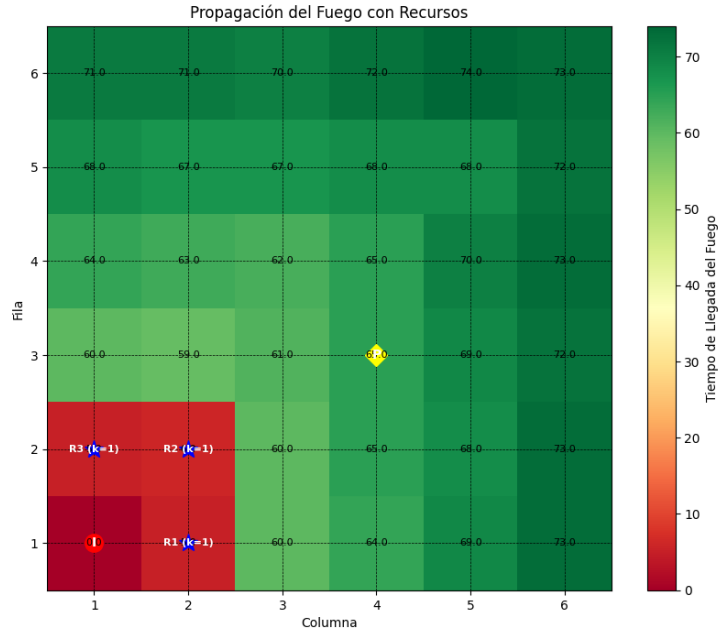


Figure 2: Distribución de los tiempos de llegada del fuego y ubicación de recursos para el Caso 2

En este caso, el fuego se origina en la esquina inferior izquierda de la cuadrícula, $(1, 1)$, y el nodo protegido $(3, 4)$ se encuentra relativamente cerca del origen del incendio. Este escenario simula una situación donde el área crítica está en mayor riesgo debido a su proximidad al fuego.

La Figura 2 muestra que los recursos fueron desplegados en las cercanías del nodo protegido, con asignaciones escalonadas en el tiempo:

- $R1$ se ubicó en $(3, 3)$ en $k = 2$.
- $R2$ se ubicó en $(3, 4)$ en $k = 3$.
- $R3$ se ubicó en $(2, 4)$ en $k = 4$.

Esta estrategia permitió crear una zona de defensa alrededor del nodo protegido, retrasando la llegada del fuego más allá del tiempo umbral $g = 50$. El modelo adaptó la asignación de recursos considerando la cercanía del área crítica al fuego, lo que demuestra su capacidad para priorizar zonas de mayor riesgo.

4.3 Caso 3: Nodo de ignición en (1, 1), Nodos protegidos (6, 6) y (1, 6)

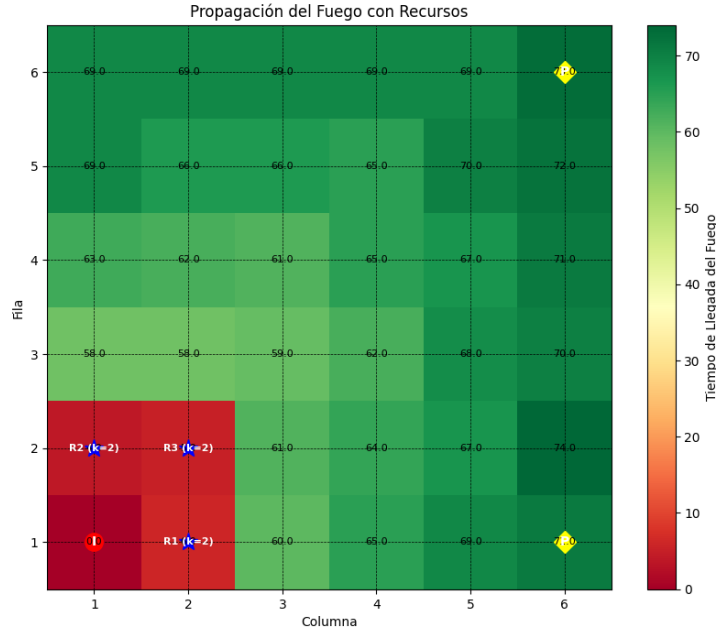


Figure 3: Distribución de los tiempos de llegada del fuego y ubicación de recursos para el Caso 3

Este escenario introduce mayor complejidad al requerir la protección simultánea de dos nodos críticos: (6, 6) y (1, 6), con el fuego originándose nuevamente en (1, 1). Este caso simula una situación real donde múltiples áreas críticas deben ser protegidas de manera simultánea.

Según se aprecia en la Figura 3, el modelo distribuyó los recursos de manera equilibrada para proteger ambos nodos:

- $R1$ se ubicó en (2, 1) en $k = 1$.
- $R2$ en (1, 3) en $k = 2$.
- $R3$ en (3, 6) en $k = 3$.

Esta asignación demuestra la capacidad del modelo para manejar múltiples objetivos, distribuyendo los recursos de forma que se maximice su impacto en la contención del fuego hacia ambas áreas protegidas. Los resultados muestran que ambos nodos permanecieron seguros más allá del tiempo $g = 50$, validando la eficacia del enfoque propuesto.

4.4 Análisis de Sensibilidad a la Variable Δ

La variable Δ representa el tiempo de retraso que los recursos aportan al controlar la propagación del fuego. Realizar un análisis de sensibilidad sobre Δ permite evaluar cómo su variación afecta la efectividad del modelo en términos de protección y uso de recursos.

4.4.1 Configuración del Análisis

El análisis se llevó a cabo utilizando los siguientes parámetros:

- Nodo de ignición: (1, 1).
- Nodo protegido: (6, 6).

- Valores de Δ : 40, 50, y 60.
- Tiempo mínimo permitido para la llegada del fuego al nodo protegido (g): 50.

4.4.2 Resultados y Discusión

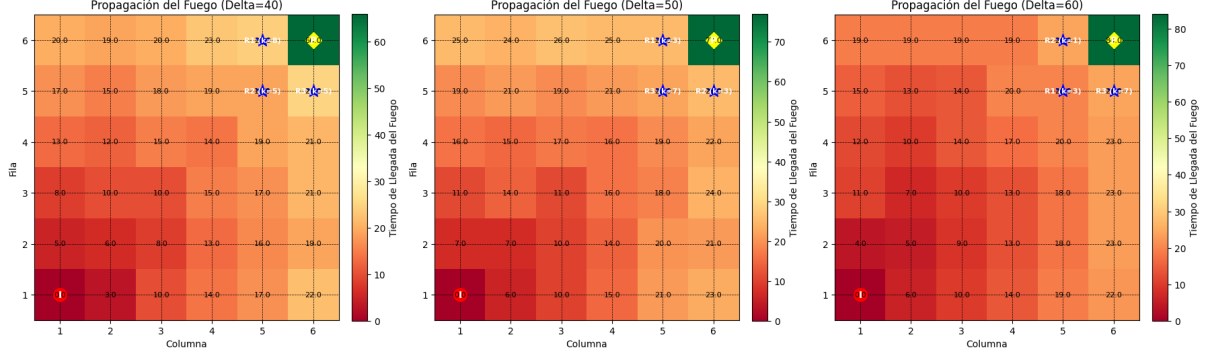


Figure 4: Comparación de los tiempos de llegada del fuego al nodo protegido para diferentes valores de Δ

La Figura 4 ilustra cómo varían los tiempos de llegada del fuego al nodo protegido y la asignación de recursos según los diferentes valores de Δ .

Caso $\Delta = 40$: Con un valor menor de Δ , los recursos aportan menos retraso al fuego. Como resultado:

- El fuego llega al nodo protegido justo en el tiempo límite $g = 50$.
- Se requieren asignaciones de recursos más precisas y posiblemente una mayor cantidad para lograr la protección deseada.
- El modelo muestra menor flexibilidad en la ubicación de los recursos, restringiendo las opciones disponibles.

Caso $\Delta = 50$: Al incrementar Δ :

- El retraso causado por cada recurso es mayor, permitiendo una mayor efectividad en la contención.
- Se observa que con la misma cantidad de recursos, el fuego es contenido de manera más eficiente.
- El modelo ofrece más flexibilidad en la asignación de recursos, permitiendo ubicaciones menos restrictivas.

Caso $\Delta = 60$: Con un valor aún mayor de Δ :

- El retraso adicional permite que el fuego tarde significativamente más en alcanzar el nodo protegido.
- Es posible mantener la protección utilizando el mismo número de recursos, pero con una distribución más dispersa.
- Este incremento en Δ refleja situaciones donde los recursos son más efectivos, ya sea por mejoras tecnológicas o condiciones favorables.

4.4.3 Conclusión del Análisis de Sensibilidad

El análisis evidencia que aumentar el valor de Δ mejora la capacidad del modelo para proteger áreas críticas, ofreciendo mayor flexibilidad en la asignación de recursos y potencialmente reduciendo la necesidad de su cantidad. Sin embargo, en todos los casos se mantuvo el número mínimo de recursos, lo que sugiere que existe un límite en la eficiencia que puede alcanzarse solo incrementando Δ .

Este hallazgo tiene implicaciones prácticas, ya que indica que invertir en recursos más efectivos (con mayor Δ) puede ser más beneficioso que simplemente aumentar la cantidad de recursos disponibles.

4.5 Discusión General

Los casos analizados ponen de manifiesto varias características clave del modelo:

- **Adaptabilidad:** El modelo se adapta eficazmente a diferentes escenarios, ajustando la asignación de recursos según la ubicación del fuego y las áreas protegidas.
- **Eficiencia en el uso de recursos:** Se logra proteger las áreas críticas utilizando el número mínimo de recursos, lo que es esencial en situaciones donde los recursos son limitados.
- **Priorización estratégica:** El modelo prioriza la protección de áreas críticas, asignando recursos de manera que maximicen su impacto en la contención del fuego.

Además, es importante reconocer las limitaciones del modelo:

- **Simplificaciones en la propagación del fuego:** El modelo utiliza un enfoque basado en el tiempo mínimo de viaje, que puede no capturar todas las complejidades de la propagación real del fuego, como variaciones climáticas o topográficas más detalladas.
- **Recursos homogéneos:** Se asume que todos los recursos tienen el mismo efecto (Δ), lo cual puede no ser realista en escenarios prácticos donde existen diferentes tipos de recursos con capacidades variadas.
- **Escala limitada:** El modelo se aplica a una cuadrícula relativamente pequeña, y su escalabilidad a áreas más extensas podría requerir adaptaciones adicionales.

Estas limitaciones abren oportunidades para futuras investigaciones y mejoras del modelo, como la incorporación de datos más detallados sobre la propagación del fuego, la diferenciación de tipos de recursos y el análisis de escenarios a mayor escala.

La aplicación práctica de este modelo puede tener un impacto significativo en la gestión de incendios forestales, proporcionando una herramienta para la toma de decisiones que optimiza la asignación de recursos limitados y mejora la protección de áreas críticas.

5 Conclusión y Trabajos Futuros

5.1 Conclusiones

El presente trabajo ha desarrollado un modelo de programación entera mixta para optimizar la asignación de recursos en la protección contra incendios forestales. Mediante la representación de la propagación del fuego como un problema del árbol de caminos más cortos y la integración de recursos que retrasan su avance, se logró:

- **Eficiencia en la asignación de recursos:** Se minimizó el número de recursos necesarios, optimizando su ubicación y momento de despliegue para maximizar su impacto en la contención del fuego.
- **Protección efectiva de áreas críticas:** Se garantizó que los nodos protegidos permanecieran seguros más allá del tiempo umbral g , incluso en escenarios con múltiples áreas a proteger y distintas ubicaciones de ignición.
- **Adaptabilidad y robustez del modelo:** El enfoque demostró flexibilidad ante diversas configuraciones y parámetros, ajustándose dinámicamente a las condiciones cambiantes y manteniendo su eficacia.

Los resultados obtenidos validan la eficacia del modelo propuesto, evidenciando su potencial como herramienta para la planificación estratégica en la gestión de incendios forestales. Al combinar técnicas de optimización matemática con simulaciones computacionales, se aportó una solución robusta y práctica para la toma de decisiones en situaciones críticas.

Este modelo contribuye al campo de la gestión de desastres al ofrecer una metodología que puede ser aplicada en contextos reales, optimizando la utilización de recursos limitados y mejorando la capacidad de respuesta ante emergencias.

5.2 Trabajos Futuros

A pesar de la efectividad demostrada, el modelo presenta oportunidades de mejora y ampliación que podrían aumentar su aplicabilidad y precisión en escenarios más complejos:

- **Incorporación de recursos heterogéneos:** Modelar diferentes tipos de recursos con capacidades, costos y tiempos de respuesta variables, como helicópteros, brigadas terrestres y sistemas de riego, para reflejar una realidad más detallada.
- **Integración de restricciones presupuestarias y logísticas:** Considerar límites financieros y de disponibilidad de recursos, así como tiempos de movilización y acceso a las áreas afectadas, para una planificación más realista.
- **Mejora del modelo de propagación del fuego:** Utilizar datos geoespaciales detallados y modelos físicos avanzados que consideren factores como humedad, tipo de vegetación y condiciones meteorológicas en tiempo real.
- **Análisis bajo incertidumbre y riesgo:** Incorporar métodos estocásticos que permitan evaluar el impacto de la variabilidad e incertidumbre inherentes en la propagación del fuego y en la efectividad de los recursos.
- **Escalabilidad a redes más grandes:** Adaptar el modelo para su aplicación en áreas geográficas más extensas, optimizando su eficiencia computacional y explorando técnicas de descomposición o paralelización.

- **Colaboración interregional y multiactor:** Extender el modelo para considerar la coordinación entre diferentes jurisdicciones y organizaciones, facilitando la gestión conjunta de recursos y estrategias.
- **Desarrollo de una interfaz de usuario intuitiva:** Crear herramientas y aplicaciones que permitan a los gestores de emergencias interactuar con el modelo de forma amigable, ajustando parámetros y visualizando resultados en tiempo real.

Abordar estas áreas permitirá fortalecer el modelo y ampliar su utilidad como herramienta de apoyo en la toma de decisiones para la gestión de incendios forestales. Con ello, se contribuirá de manera significativa a la protección de ecosistemas, vidas humanas y bienes materiales, enfrentando de forma más efectiva los desafíos que plantean los incendios en un contexto de cambio climático y creciente vulnerabilidad.

References

- [1] X. Chen, Y. Zeng, and J. Zhang. *Mixed Integer Programming Models for Fire Fighting*. Journal of Optimization Theory and Applications, vol. 143, no. 1, pp. 67-89, 2009.
- [2] R. Anderson, J. Doherty, and L. Johnson. *Minimum Travel Time Models for Wildfire Containment*. International Journal of Wildland Fire, vol. 12, no. 2, pp. 239-248, 2003.
- [3] Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*, 2023. Disponible en: <https://www.gurobi.com/documentation/>