



Departamento de
Ingeniería Industrial
Universidad de Concepción

MODELOS Y ALGORITMOS PARA EL MOVING FIREFIGHTER PROBLEM

Por: Claudio Alfonso Medina González

Tesis presentada a la Facultad de Ingeniería de la Universidad de Concepción
para optar al grado académico de Magíster en Ingeniería Industrial

Septiembre 2025
Concepción, Chile

Profesor Guía: Carlos Emilio Contreras Bolton

© 2025, Claudio Alfonso Medina González

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento

AGRADECIMIENTOS

Esta tesis fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (CCSS210001). Se recibió financiamiento adicional de ANID mediante el FONDECYT INICIACIÓN 11241132.

Agradecer llegar a esta instancia, a la curiosidad que me lleva a emprender nuevos desafíos, aunque muchas veces haya sido con bastante esfuerzo mental. Agradecer a los que han estado presentes a la distancia a lo largo de estos años, mi familia, mis padres, especialmente mi hermana por demostrarme que la valentía no es la ausencia de miedo, sino la capacidad de seguir adelante a pesar de él como diría Mandela. A aquellos que ya no están presentes, mis abuelos Luis Segundo, cuyos recuerdos muchas veces son valores en mi vida y José Manuel, quien transmitía cariño aunque hablaba poco, ellos viven en mis recuerdos. A ambas abuelas, ambas mujeres fuertes, Maria Elisa y Ema del Rosario. A los amigos que han estado ahí para hacer más amenas las jornadas de estudio, y muchos que integran parte de mi familia escogida creando nuevos recuerdos. Aquellos que he conocido durante la realización de esta tesis. Profesores del programa, que han hecho agradables las clases y brindando humanidad en cada clase. A Hannu y Markku, por enseñarme la calidez de la gente de un país tan frío, experiencias que amplían el mundo que conozco. Y especialmente al profesor guía Carlos Contreras por su gran apoyo, dedicación, paciencia, carisma y ayuda brindada durante la realización de esta tesis. Agradecer a la gente que de alguna u otra manera me alegra cada día, que quizás no incluyo para no seguir extendiendo esto.

Resumen

Esta tesis aborda el Moving Firefighter Problem (MFP), una extensión del clásico Firefighter Problem (FP), que modela la contención de fenómenos que se propagan en redes. El MFP introduce una representación más realista del problema al incorporar el movimiento de los bomberos en la red, cuya capacidad de respuesta depende de las distancias euclidianas entre nodos. Esta consideración hace que el tiempo de protección ya no sea uniforme y aumenta significativamente la complejidad del problema, lo que incrementa notablemente los tiempos de resolución. Actualmente, solo existe un modelo de programación cuadrática en la literatura, que resuelve eficientemente tamaños pequeños del problema. Para enfrentar este desafío, se proponen tres enfoques: un modelo de programación lineal entera, un modelo de programación con restricciones, y una metaheurística basada en el Iterated Local Search y Simulated Annealing (ILS-SA). Este último combina perturbaciones controladas con una búsqueda local con dos tipos de intensidades, junto con un mecanismo de aceptación probabilística dependiente de una temperatura que desciende a medida que avanzan las iteraciones. Esta estrategia permite equilibrar exploración e intensificación, mejorando la calidad de las soluciones y evitando quedar atrapado en óptimos locales. Los modelos y algoritmo propuestos son validados en un conjunto de 160 instancias de la literatura. Los resultados computacionales evidenciaron que el modelo de programación con restricciones logró encontrar mejores soluciones o cercanas al óptimo en tiempos de cómputo considerablemente más bajos que los modelos de programación cuadrática y lineal entera. Por su parte, el ILS-SA obtuvo soluciones competitivas con una notable reducción del tiempo de cómputo, destacando su utilidad para instancias de mayor tamaño o escenarios donde se requiere una respuesta rápida.

Keywords: *Moving Firefighter Problem, Programación Lineal Entera, Programación por Restricciones, Simulated Annealing, Iterated Local Search.*

Abstract

This thesis addresses the Moving Firefighter Problem (MFP), an extension of the classical Firefighter Problem (FP), which models the containment of spreading phenomena in networks. The MFP introduces a more realistic representation of the problem by incorporating the movement of firefighters within the network, where their response capacity depends on the Euclidean distances between nodes. This consideration makes the protection time non-uniform and significantly increases the problem's complexity, which in turn greatly extends solution times. Currently, only one quadratic programming model exists in the literature, capable of efficiently solving small-sized instances of the problem. To tackle this challenge, three approaches are proposed: an integer linear programming model, a constraint programming model, and a metaheuristic based on Iterated Local Search and Simulated Annealing (ILS-SA). The latter combines controlled perturbations with a local search featuring two levels of intensity, along with a probabilistic acceptance mechanism that depends on a temperature value decreasing as iterations progress. This strategy allows for a balance between exploration and intensification, improving solution quality and helping avoid entrapment in local optima. The proposed models and algorithm are validated on a benchmark set of 160 instances from the literature. Computational results showed that the constraint programming model was able to find better or near-optimal solutions in significantly lower computation times compared to the quadratic and integer linear programming models. Meanwhile, the ILS-SA produced competitive solutions with a notable reduction in computational time, highlighting its usefulness for larger instances or scenarios requiring rapid responses.

Keywords: *Moving Firefighter Problem, Integer Linear Programming, Constraint Programming, Simulated Annealing, Iterated Local Search.*

Índice General

AGRADECIMIENTOS	I
Resumen	II
Abstract	III
1 Introducción	1
1.1 Motivación	1
1.2 Hipótesis de investigación	3
1.3 Objetivos	3
1.3.1 Objetivo general	3
1.3.2 Objetivos específicos	4
1.4 Estructura del documento	4
2 The moving firefighter problem	5
2.1 The moving firefighter problem definición	5
2.2 Modelo de programación cuadrática entera mixta	7
2.3 Trabajos relacionados	10
3 Modelos de programación matemática propuestos	14
3.1 Modelo de MILP	14
3.2 Modelo de programación con restricciones	16
3.3 Obtención de los parámetros B y D	17
4 Metaheurística propuesta	19
4.1 Algoritmo general	19
4.2 Representación y evaluación de la solución	21
4.3 Solución inicial	23
4.4 Perturbaciones	24
4.4.1 Búsqueda local	25
5 Resultados	26
5.1 Instancias	26
5.2 Configuración de los experimentos	27
5.3 Parámetros de ILS-SA	27
5.4 Resultados de los modelos	28
5.5 Resultados de la metaheurística	30
5.6 Análisis de los componentes de ILS-SA	32

Índice General	v
6 Conclusión	37
Referencias	39
Anexo	42

Índice de Tablas

1	Resumen de la notación.	8
2	Parámetros utilizados en el algoritmo y sus valores calibrados.	28
3	Comparación entre los modelos de MIQCP, MILP y CP.	29
4	Comparación entre los métodos MIQCP, MILP, MCP e ILS-SA.	31
5	Resultados comparativos entre algoritmos propuestos e ILS-SA.	34

Índice de Figuras

1	Ejemplo de una instancia factible del MFP.	7
2	Representación de la solución.	21
3	Gráficos de caja de los tres modelos e ILS-SA.	32
4	Gráficos de convergencia de algoritmos propuestos para la instancia 135.	36
5	Propagación del fuego caso en que bombero no interviene parte a.	43
6	Propagación del fuego caso en que bombero no interviene parte b.	44
7	Propagación del fuego con solución presentada.	45

Lista de Algoritmos

1	Pseudocódigo para ILS-SA	20
2	Función de evaluación de la solución	22
3	Función de propagación del fuego	23

Capítulo 1

Introducción

Este capítulo presenta a modo introductorio una motivación para el trabajo, hipótesis, objetivos y la organización de la estructura del documento.

1.1. Motivación

Los incendios forestales representan una problemática crítica tanto a nivel nacional como global, debido a sus graves impactos en el medio ambiente, la economía y la seguridad de las personas. En particular, Chile ha enfrentado temporadas de incendios cada vez más intensas y destructivas, afectando especialmente la zona centro-sur del país. La región del Biobío concentra la mayor superficie de plantaciones forestales de Chile con más de 630 mil hectáreas (INFOR, 2023). Por tanto, es especialmente vulnerable a la propagación del fuego, lo que puede tener consecuencias devastadoras si no se implementan adecuadas estrategias de prevención, manejo y combate del fuego.

En los últimos años, los incendios han alcanzado dimensiones sin precedentes. Durante la temporada 2022–2023, se consumieron más de 430 mil hectáreas, mientras que la temporada 2016–2017 fue la más destructiva de la historia reciente, con más de 570 mil hectáreas afectadas (CONAF, 2024). Estos antecedentes resaltan la urgencia de desarrollar herramientas que mejoren la planificación y respuesta ante emergencias, en particular, en la contención de incendios en redes. Existe investigación que busca contribuir a una gestión eficiente de los recursos de protección y mitigación en estos escenarios de alta complejidad. Un caso es el Firefighter Problem (FP), que modela cómo contener la propagación de eventos adversos en redes, como incendios o

epidemias. La optimización de la asignación de recursos limitados en estos escenarios es crucial para mejorar la respuesta ante emergencias y minimizar daños en situaciones críticas.

El FP es modelado en un grafo compuesto por nodos y aristas, de manera determinista y en tiempo discreto (Hartnell & Rall, 1995). En un instante inicial (tiempo 0), un nodo comienza con el proceso de propagación del fuego. Además, un bombero defiende un nodo del grafo en cada periodo de tiempo, a la vez que el fuego se propaga hacia todos los nodos adyacentes no defendidos en igual periodo de tiempo. Una vez que el nodo es quemado o defendido, permanece en ese estado. Este proceso termina cuando el fuego no puede continuar propagándose. El FP considera que el fuego no puede propagarse por sobre los nodos protegidos, actuando estos a modo de barrera de defensa. En la forma generalizada de este problema, más de un nodo puede iniciar la propagación del fuego y más de un bombero podría iniciar la defensa de los nodos.

En la literatura, hay variantes del FP con diferentes objetivos como, minimizar el número de nodos quemados, salvar el máximo número de nodos, eliminar el fuego lo más pronto posible (corresponde a minimizar las unidades de tiempo para que el proceso finalice), proteger un conjunto seleccionado de nodos y determinar el mínimo número de bomberos que puede ser usado para contener el fuego de ciertos nodos, o minimiza el tiempo de propagación del fuego. Incluso, algunas veces estos objetivos entran en conflicto entre sí (MacGillivray, 2009).

Recientemente, se propuso el Moving Firefighter Problem (MFP) (Gutiérrez-De-La-Paz et al., 2022) que es una generalización del FP. El MFP considera que el tiempo que toma un bombero en moverse desde un nodo origen a un nodo destino está dado por una función de la distancia euclidiana. Por tanto, el tiempo de avance del fuego es diferente al tiempo protección de nodos. Los autores propusieron un modelo de programación cuadrática entera mixta (MIQCP) ¹ para resolver el MFP. Este modelo es computacionalmente costoso, ya que toma más de dos horas resolver de manera óptima 40 nodos y más de 24 horas, 50 nodos. El tiempo de procesamiento suele disminuir cuando hay más nodos encendidos inicialmente o cuando aumenta el factor de escala de distancias, lo que facilita defender más nodos.

El modelo de MIQCP actual de la literatura presenta limitaciones importantes en cuanto a su escalabilidad y aplicabilidad en contextos donde se requiere una toma de decisiones ágil. Esto evidencia un vacío en la literatura respecto al desarrollo

¹MIQCP, por sus siglas en inglés: Mixed Integer Quadratically Constrained Programming.

de enfoques más eficientes para resolver el MFP en tiempos razonables. En este contexto, la presente tesis propone tres enfoques que buscan reducir significativamente los tiempos de cómputo sin sacrificar la calidad de las soluciones: un modelo de programación lineal entera mixta (MILP)², un modelo de programación con restricciones (CP)³ y una metaheurística basada en Iterated Local Search (ILS) y Simulated Annealing (SA).

El algoritmo propuesto, ILS-SA, emplea perturbaciones controladas y una búsqueda local con dos tipos de intensidades, junto con un mecanismo de aceptación probabilística dependiente de una temperatura que desciende a medida que avanzan las iteraciones. Este enfoque permite equilibrar la exploración de nuevas soluciones y la intensificación en áreas prometedoras del espacio de soluciones, evitando quedar atrapado en óptimos locales. La probabilidad de aceptar soluciones peores disminuye gradualmente a medida que la temperatura baja, favoreciendo la búsqueda local a medida que el proceso avanza. Los tres enfoques propuestos son validados a través de 160 instancias de la literatura, que consideran instancias de 10 a 40 nodos. El modelo de CP encuentra soluciones de alta calidad en tiempos de cómputo más rápidos que los modelos de MIQCP y MILP. Mientras, el ILS-SA proporciona soluciones competitivas en instancias grandes, siendo especialmente útil cuando se requieren respuestas rápidas.

1.2. Hipótesis de investigación

Es posible proponer nuevos enfoques que resuelvan eficientemente el MFP en términos de tiempo de cómputo y calidad de solución.

1.3. Objetivos

Para lograr la hipótesis planteada previamente, se proponen los siguientes objetivos.

1.3.1. Objetivo general

Implementar dos modelos y un algoritmo para abordar el MFP de manera que sea eficiente en términos de tiempo de cómputo y en calidad de solución respecto a la literatura.

²MILP, por sus siglas en inglés: Mixed Integer Linear Programming.

³CP, por sus siglas en inglés: Constraint Programming.

1.3.2. Objetivos específicos

- Revisar la literatura del MFP y otros problemas similares.
- Formular un modelo de MILP para el MFP.
- Formular un modelo de CP para abordar el MFP.
- Diseñar una metaheurística eficiente para resolver el MFP.
- Implementar los modelos matemáticos y el algoritmo diseñados.
- Realizar experimentos computacionales para validar las soluciones propuestas.
- Evaluar y comparar el desempeño de los distintos enfoques con el estado del arte del MFP.

1.4. Estructura del documento

El resto del documento se estructura de la siguiente manera. El Capítulo 2 presenta la definición formal y el modelado del MFP para un bombero y trabajos relacionados a este. El Capítulo 3 presenta un modelo de MILP y un modelo de CP. El Capítulo 4 presenta la metaheurística propuesta con sus principales componentes y una explicación detallada. El Capítulo 5 presenta los resultados de la comparación entre los dos modelos propuestos, modelo de la literatura y la metaheurística propuesta. Finalmente, el Capítulo 6 presenta las conclusiones y trabajos futuros.

Capítulo 2

The moving firefighter problem

Este capítulo introduce la definición formal del MFP, estableciendo su formulación matemática y las restricciones que determinan la validez de una solución. Se revisan aproximaciones existentes y variantes del problema.

2.1. The moving firefighter problem definición

El MFP ([Gutiérrez-De-La-Paz et al., 2022](#)) se describe formalmente como un 6-tuplo (G, F, a, τ, f, T) , donde $G = (V, E)$ representa un grafo no dirigido de nodos V y aristas E ; $F \subseteq V$ es el conjunto de nodos en los cuales el fuego está encendido; a es el nodo inicial en que los bomberos están inicialmente ubicados (conocido también como depósito); $\bar{V} = V \cup \{a\}$ es el conjunto de nodos que incluye el depósito; $\tau : \bar{V} \times V \rightarrow \mathbb{R}_0^+$ es una función que determina el tiempo que tarda un bombero (o equipo de bomberos) en moverse desde el depósito a o desde un nodo $u \in V$ hacia otro nodo $v \in V$ para defenderlo; f es la cantidad de bomberos disponibles; y $T > 0$ es una constante que define el tiempo que tarda el fuego en propagarse desde un nodo quemado $u \in V$ a sus vecinos no protegidos en el grafo G .

Dado el supuesto que el fuego se propaga de forma uniforme a una tasa constante, el tiempo se discretiza en intervalos de duración T , cada uno denominado ronda de quema. Dentro de cada ronda de quema, el bombero tiene la oportunidad de defender nodos específicos en una serie de instantes, llamados rondas de defensa. Una vez que las rondas de defensa han concluido, el fuego se propaga desde los nodos incendiados a sus vecinos que no fueron defendidos. Una vez que un nodo es quemado o defendido, su estado se mantiene inalterable durante todo el proceso. El proceso finaliza cuando

el fuego ya no puede propagarse a nuevos nodos. La propagación del fuego comienza en la ronda 0, cuando únicamente los nodos del conjunto inicial F se encuentran en llamas. Se considera que un nodo está protegido si no se encuentra ni quemado ni defendido al término del proceso.

En este trabajo, al igual que [Gutiérrez-De-La-Paz et al. \(2022\)](#), se estudia el caso de un único bombero. Así una solución válida para el MFP es una secuencia de nodos que el bombero debe defender, $S = \{u_0, u_1, \dots, u_l\}$, donde l representa la posición del último nodo defendido y u_0 es a . Esta secuencia S , debe cumplir con la Ecuación (1) para cada $i = \{1, 2, \dots, l\}$. Donde $\beta(S_i)$ representa el tiempo en el que el nodo u_i se quemaría si la secuencia de defensa $S = \{u_0, u_1, \dots, u_l\}$ se truncara a $S_i = \{u_0, u_1, \dots, u_{i-1}\}$. El objetivo de la Ecuación (1) es restringir el conjunto de soluciones factibles, de modo que el bombero tenga el tiempo necesario para defender cada nodo de la secuencia antes de que el fuego los alcance.

$$\sum_{k=1}^{i-1} \tau(u_k, u_{k+1}) \leq \beta(S_i) \quad \forall i \geq 1 \quad (1)$$

La Figura 1 muestra una instancia factible del MFP, donde grafo G está compuesto por $V = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ y $E = \{(0, 6), (6, 5), (5, 4), (4, 1), (1, 3), (3, 2), (5, 8), (8, 7), (4, 8), (3, 7), (7, 9), (7, 10), (8, 11), (11, 10), (11, 12), (5, 12)\}$. El fuego comienza en los nodos $F = \{0, 1\}$, un solo bombero ($f = 1$) inicialmente estacionado en el depósito a , y $T = 1$. El depósito (a) y los nodos en V están en el plano euclidiano. Los números en violeta sobre algunos nodos indican su distancia más corta a su nodo más cercano en F , que en este ejemplo equivale al menor tiempo en el que arden. La figura también muestra una solución factible $S = \{a, 6, 5, 8, 7\}$ en que el bombero se mueve desde el nodo a para defender el nodo 5, luego se mueve a defender el nodo 6, acabando la primera ronda de quema, posteriormente al 8 y finalmente, se mueve para defender el nodo 7. Para esta secuencia de defensa S , el valor de $\beta(S_1)$ es igual a 1 porque el nodo 6 arde en el tiempo 1 cuando se aplica la secuencia de protección truncada $S_1 = \{a\}$, $\beta(S_2)$ es igual a 2 ya que el nodo 5 arde en el tiempo 2, cuando se aplica la secuencia de protección truncada $S_2 = \{a, 6\}$. En consecuencia, $\beta(S_3)$ es igual a 2, cuando se aplica la secuencia de protección truncada $S_3 = \{a, 6, 5\}$, y de manera similar, $\beta(S_4)$ es igual a 2. S es una secuencia válida porque $(a, 6) = 0,50 \leq \beta(S_1) = 1$, $(a, 6) + (6, 5) = 0,90 \leq \beta(S_2) = 2$, $(a, 6) + (6, 5) + (5, 8) = 1,29 \leq \beta(S_3) = 2$, $(a, 6) + (6, 5) + (5, 8) + (8, 7) = 1,59 \leq \beta(S_4) = 2$. El proceso termina después de dos rondas de quema (en el turno dos) dado que en ese momento el fuego ya no se puede

propagar. El valor de evaluación de esta solución es cinco porque los nodos 0, 1, 2, 3 y 4 están quemados al final del proceso.

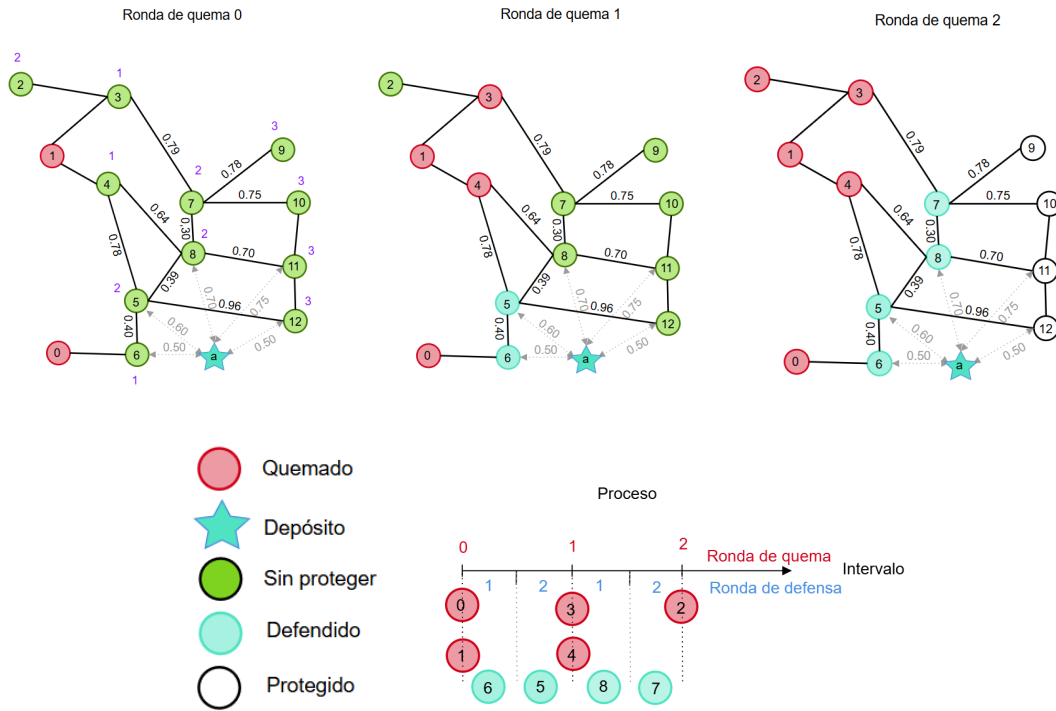


Figura 1: Ejemplo de una instancia factible del MFP.

2.2. Modelo de programación cuadrática entera mixta

El MIQCP para el MFP es propuesto por [Gutiérrez-De-La-Paz et al. \(2022\)](#). La formulación usa las siguiente variables. Dos variables de decisión, b_{ij} y d_{ij} , $i \in V$, $j \in R^B$. Estas variables binarias reportan el estado de cada nodo $i \in V$ en la ronda $j \in R^B$ de quema, es decir, $b_{ij} = 1$ si se ha quemado el nodo i en la ronda $j \in R^B$ de quema, caso contrario 0. De igual manera, para $d_{ij} = 1$ si se ha defendido el nodo $i \in V$ en la ronda $j \in R^B$, caso contrario 0. Adicionalmente, se define la variable \bar{d}_{jik} para reportar la secuencia de nodos $i \in V$ defendidos en cada ronda de defensa $k \in R^D$ de la ronda de quema $j \in R^B$. Por tanto, $\bar{d}_{jik} = 1$ si el nodo i está defendido en la ronda k de defensa y en la ronda j de quema, caso contrario 0. Además, se define la variable p_{jik} para reportar el último nodo $i \in V$ defendido en cada ronda de defensa $k \in R^D$ de la ronda de quema $j \in R^B$. Donde $p_{jik} = 1$ si el nodo i fue el último nodo defendido en la ronda k de defensa, en la ronda j de quema, caso contrario 0. Para facilitar la presentación, la Tabla 1 resume los conjuntos, parámetros y variables de

decisión.

Tabla 1: Resumen de la notación.

Conjuntos	Descripción
V	Conjunto de nodos, $\{1, 2, \dots, n\}$.
\bar{V}	Conjunto de nodos incluyendo el depósito o punto de anclaje de los bomberos, $\{a, 1, 2, \dots, n\}$.
E	Conjunto de aristas, $\{(i, j) : i, j \in V, i \neq j\}$.
F	Conjunto de nodos donde comienza el incendio, $F \subseteq V$.
S	Secuencia de defensa, $\{a, u_1, \dots, u_l\}$.
R^B	Conjunto de rondas de quema, $\{1, 2, \dots, B\}$.
R^D	Conjunto rondas de defensa, $\{1, 2, \dots, D\}$.
Parámetros	Descripción
a	Depósito de los bomberos.
B	Número de rondas de quema.
D	Número de rondas de defensa.
T	Duración de los intervalos de tiempo.
f	Número de bomberos.
$N(i)$	Vecindad del nodo, $i \in V$.
Variables de decisión	Descripción
b_{ij}	Variable binaria que indica si $i \in \bar{V}$ está quemado en la j -ésima ronda de quema, $j \in R^B$.
d_{ij}	Variable binaria que indica si $i \in \bar{V}$ es defendido en la j -ésima ronda de quema, $j \in R^B$.
\bar{d}_{jik}	Variable binaria que indica si $i \in \bar{V}$ es defendido en la k -ésima defensa de la j -ésima ronda de quema, $k \in R^D, j \in R^B$.
p_{jik}	Variable binaria que indica si $i \in \bar{V}$ fue el último nodo defendido en la k -ésima defensa de la j -ésima ronda de quema, $k \in R^D, j \in R^B$.
t_j	Variable continua que indica el tiempo necesario para ejecutar una secuencia de defensa en la j -ésima ronda de quema, $j \in R^B$.

El modelo de MIQCP se presenta a continuación:

$$\text{minimizar } \sum_{i \in V} b_{iB} \quad (2)$$

sujeto a:

$$b_{i0} = 1 \quad \forall i \in \bar{V} : i \in F \quad (3)$$

$$b_{i0} = 0 \quad \forall i \in \bar{V} : i \notin F \quad (4)$$

$$b_{ij} \geq b_{ij-1} \quad \forall i \in \bar{V}, j \in R^B \quad (5)$$

$$d_{a0} = 1 \quad (6)$$

$$d_{i0} = 0 \quad \forall i \in V \quad (7)$$

$$d_{ij} \geq d_{ij-1} \quad \forall i \in \bar{V}, j \in R^B \quad (8)$$

$$b_{ij} + d_{ij} \leq 1 \quad \forall i \in \bar{V}, j \in R^B \quad (9)$$

$$b_{ij} + d_{ij} \geq b_{mj-1} \quad \forall i \in \bar{V}, j \in R^B, m \in N(i) \quad (10)$$

$$\bar{d}_{0ik} = d_{i0} \quad \forall i \in \bar{V}, k \in R^D \quad (11)$$

$$\bar{d}_{ji1} \geq d_{ij-1} \quad \forall i \in \bar{V}, j \in R^B \quad (12)$$

$$\bar{d}_{jiD} = d_{ij} \quad \forall i \in \bar{V}, j \in R^B \quad (13)$$

$$\bar{d}_{jik} \geq \bar{d}_{jik-1} \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \setminus \{1\} \quad (14)$$

$$p_{0ik} = d_{i0} \quad \forall i \in \bar{V}, k \in R^D \quad (15)$$

$$p_{jik} \geq \bar{d}_{jik} - \bar{d}_{jik-1} \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \setminus \{1\} \quad (16)$$

$$p_{ji1} \geq \bar{d}_{ji1} - d_{ij-1} \quad \forall i \in \bar{V}, j \in R^B \quad (17)$$

$$\sum_{i \in \bar{V}} p_{jik} = 1 \quad \forall k \in R^D, j \in R^B \quad (18)$$

$$p_{jik} \geq p_{jik-1} \left(1 - \sum_{l \in \bar{V}} (\bar{d}_{jlk} - \bar{d}_{jlk-1}) \right) \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \setminus \{1\} \quad (19)$$

$$p_{ji1} \geq p_{j-1iD} \left(1 - \sum_{l \in \bar{V}} (\bar{d}_{jl1} - d_{lj-1}) \right) \quad \forall i \in \bar{V}, j \in R^B \quad (20)$$

$$t_0 = 0 \quad (21)$$

$$t_j = t_{j-1} + \sum_{l \in \bar{V}} \sum_{i \in \bar{V}} p_{jil} \tau_i p_{j-1lD} + \sum_{k \in R^D \setminus \{1\}} \sum_{l \in \bar{V}} \sum_{i \in \bar{V}} p_{jik} \tau_i p_{jlk-1} \quad \forall j \in R^B \quad (22)$$

$$t_j \leq j \quad \forall j \in R^B \quad (23)$$

$$b_{ij}, d_{ij} \in \{0, 1\} \quad i \in \bar{V}, j \in R^B \quad (24)$$

$$p_{jik}, \bar{d}_{jik} \in \{0, 1\} \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \quad (25)$$

$$t_j \in \mathbb{R}^+ \quad \forall j \in R^B \quad (26)$$

Las restricciones (3), (4) y (5) consideran que si un nodo $i \in \bar{V}$ es quemado en la ronda $j \in R^B$, permanece en ese estado en la siguiente ronda. La restricción (6) indica que el nodo de inicio a debe ser defendido al comienzo del proceso, de manera similar, las restricciones (7) indican que solo el nodo a se encuentra defendido en

el instante 0. Las restricciones (8) consideran que si un nodo $i \in \bar{V}$ es defendido en la ronda $j \in R^B$, permanecerá en ese estado hasta el final del proceso de quemado. Las restricciones (9) indican que el nodo está quemado, defendido o ninguna de las anteriores, no hay ocurrencia de ambas a la vez. Las restricciones (10) indican que el vecino directo de un nodo quemado en el turno $j \in R^B$ anterior, es quemado o es defendido en el siguiente turno.

Las restricciones (11) y (12) corresponden a todos los nodos defendidos en la $j - 1$ ($j \in R^B$) ronda de quema ya que deben permanecer defendidos al comienzo de la primera ronda de defensa en la siguiente ronda de quema. Las restricciones (13) son los estados de los nodos defendidos en el fin de la ronda $j \in R^B$ de quema. Para las restricciones (14), resulta similar a (8), donde si un nodo es protegido, permanece en ese estado. Las restricciones (15), (16), (17) y (18) establecen la relación entre las variables binarias p_{jik} y d_{jik} . Es importante destacar que $p_{jik} = 1$ puede interpretarse de dos maneras:

1. El nodo $i \in \bar{V}$ fue defendido exactamente en la $k \in R^D$ ronda de defensa dentro de la ronda de propagación $j \in R^B$, de acuerdo con las restricciones (16) y (17).
2. El nodo $i \in \bar{V}$ fue el último en ser defendido en una ronda anterior.

Las restricciones (18) aseguran que en cada ronda de defensa haya un único nodo marcado como el último defendido. Las restricciones (19) y (20) garantizan que p_{jik} sea 1 si hubo un aumento en la cantidad de nodos defendidos en la ronda $k \in R^D$ de defensa, de la ronda $j \in R^B$ de quema, en relación a la ronda de defensa anterior. Las restricciones (22) y (23) aseguran que el tiempo t_j requerido para defender una secuencia de nodos durante la $j \in R^B$ ronda de quema no exceda la cantidad de rondas de quema disponibles. Finalmente, las restricciones (24), (25) y (26) definen el dominio de las variables de decisión.

2.3. Trabajos relacionados

El FP ha ganado relevancia por sus propiedades teóricas, pero también debido a que puede ser útil en proveer de modelos simples para fenómenos más complejos como propagación de virus (Bonato et al., 2014), propagación de fuego en estaciones de terminales de combustible (Khakzad, 2018) y del fuego en edificios (Wang et al., 2022), etc. Por otra parte, problemas similares se han modelado mediante grafos,

como la propagación de campañas de marketing esparcimiento de ideas e innovaciones en redes sociales por boca a boca ([Kempe et al., 2005](#)).

Un acercamiento mediante el modelado para terminales de combustible puede encontrarse en [Khakzad \(2018\)](#). Este trabajo tiene como objetivo realizar una aproximación mediante grafos para impedir el efecto dominó en grandes incendios en terminales de combustible. En este caso, los bomberos pueden adoptar tres medidas, defensiva, ofensiva y mixta. La defensiva es cuando las unidades expuestas al fuego son enfriadas con agua. Ofensiva es cuando las unidades que se están quemando son atacadas. Finalmente, en un enfoque mixto considera las dos medidas anteriores, que corresponde a la mayoría de acciones durante incendios de terminales de combustible.

El FP puede ser planteado en grafos infinitos por [Chen et al. \(2017\)](#), donde hay un número finito de bomberos disponibles para ser asignados a algunos nodos del grafo en el tiempo inicial, y cada bombero solo puede moverse desde su ubicación actual a uno de sus vecinos, o permanecer en el mismo lugar, en cada paso de tiempo. Un nodo es considerado defendido si un bombero lo alcanza a más tardar al mismo tiempo que el fuego. En este caso, el problema se centró más bien en estudiar la contención del fuego, es decir, analizar si hay forma de evitar que el fuego siga propagándose.

De manera similar, la propagación del fuego ha sido estudiada a través del Graph Burning Problem (GBP), introducido en el contexto del contagio social por [Bonato et al. \(2014\)](#). El GBP es un problema NP-hard que permite cuantificar la vulnerabilidad de un grafo frente al proceso de quema o contagio. Este problema consiste en identificar la secuencia mínima de nodos que provocan la quema del grafo, siguiendo un proceso en el cual, en cada paso de tiempo discreto, un nodo quemado permanece en ese estado y quema a sus vecinos. Al mismo tiempo, en cada paso de tiempo se activa otro nodo para continuar la propagación, hasta que el proceso concluye cuando todo el grafo ha sido completamente quemado.

[García-Díaz et al. \(2022b\)](#) proponen un modelo de programación lineal entero y dos problemas de satisfacción de restricciones para resolver de manera óptima el GBP. Además, propusieron instancias de tamaño desde 30 nodos a más de 5200 nodos. Luego, [García-Díaz et al. \(2022a\)](#) presentaron dos algoritmos de aproximación para la resolución del GBP. Ambos algoritmos requieren de un proceso de inicialización, donde se obtienen las rutas mínimas entre cada par de nodos. El primer algoritmo requiere de conocer previamente el turno en que se quema cada nodo por adelantado, construyendo secuencias en las que se comienza con la selección de un nodo al azar, y

se sigue con el nodo más alejado, y se sigue repitiendo el proceso anterior. El segundo algoritmo es prácticamente análogo al primero, con la diferencia que el segundo no se detiene hasta que la secuencia ha cubierto todos los nodos. Mientras que el primero se detiene previamente puesto que conoce previamente el número de quema de los nodos.

Otros trabajos que han abordado el FP. [Anshelevich et al. \(2009\)](#) propusieron algoritmos de aproximación al FP basados en el algoritmo greedy para resolver el problema análogo de maximizar el número de nodos defendidos y minimizar el número de nodos quemados. [Develin & Hartke \(2007\)](#) propusieron un modelo de programación lineal entera para el FP en grillas 2D. Además, se ha mostrado que para contener el fuego en grillas 2D son suficientes dos bomberos. Los autores probaron que para grillas cuadradas d -dimensionales, con $d \geq 3$, basta con $2d - 1$ bomberos por periodo de tiempo para contener el fuego. Una versión estocástica del FP puede ser encontrada en [Michalak & Knowles \(2016\)](#), donde el fuego se expande a los nodos adyacentes con una probabilidad entre 0 y 1.

[Ng & Raff \(2008\)](#) presentaron una versión con un número variable para la cantidad de bomberos en el FP, es decir, el número de bomberos disponible en cada periodo de tiempo no es constante en una grilla infinita 2D. El mismo problema anterior ha sido estudiado por [Bazgan et al. \(2014\)](#) mostrando que el problema es NP-hard y probando parametrizaciones del problema.

En FP, se han propuesto enfoques principalmente en árboles. [Adjishvili et al. \(2017\)](#) introdujeron un problema dual del FP original, centrado en el estudio de cortes para evitar la propagación del fuego. Por su parte, [Hartnell & Li \(2000\)](#) demostraron que, en árboles donde el fuego comienza en un único nodo, un algoritmo greedy permite salvar más de la mitad de los nodos que serían salvados por una solución óptima. [Cai et al. \(2008\)](#) presentaron algoritmos para aproximaciones para FP en árboles, empleando un algoritmo de aproximación $(1 - 1/e)$, cinco algoritmos de aproximación y un algoritmo sub-exponencial para obtener soluciones óptimas en este tipo de problemas. [Coupechoux et al. \(2019\)](#) estudiaron el FP fraccional, donde cada nodo puede tener una protección parcial. [Iwaikawa et al. \(2011\)](#) desarrollaron dos algoritmos mejorados para el FP en árboles, a partir de un algoritmo de aproximación $(1 - 1/e)$, otras aproximaciones han sido desarrolladas por [Adjishvili et al. \(2017\)](#), donde se busca una aproximación al FP en árboles, mediante enfoques en un problema similar conocido como el Resource Minimization for Fire Containment en árboles.

King & MacGillivray (2010) presentaron una versión del FP con función objetivo que protege un subconjunto específico de nodos. De igual manera, lo hace Bazgan et al. (2013), estudiando el caso de árboles.

El Geometric Firefighter Problem (GFP) es otra variante similar al MFP. El GFP considera la propagación del fuego a tiempo constante, mientras que el desplazamiento de la protección al fuego, se mueve a tiempos diferentes en regiones en lugar de grafos. Algoritmos de aproximación para GFP se encuentran en Klein et al. (2018), una variante del GFP conocida como Geometric Firefighter Routing Problem ha sido estudiada en Zambon et al. (2019).

Farhadi et al. (2021) introdujeron el Traveling Firefighter Problem, como una variante del problema del vendedor viajero. Donde el bombero busca apagar nodos encendidos y lo que se busca es minimizar la sumatoria de elementos del vector que representa los tiempos de llegada a los distintos nodos donde hay un incendio.

Capítulo 3

Modelos de programación matemática propuestos

Este capítulo presenta dos formulaciones propuestas para el MFP. Un modelo de MILP que linealiza las restricciones del modelo cuadrático existente y facilitar su resolución computacional, y un segundo modelo basado en CP.

3.1. Modelo de MILP

Este modelo propuesto es una linealización del modelo de MIQCP propuesto por [Gutiérrez-De-La-Paz et al. \(2022\)](#). MIQCP es cuadrático debido a que las restricciones (19), (20) y (22) son cuadráticas porque en cada una de ellas aparecen productos entre variables de decisión, lo que los hace lineales de segundo grado. En las restricciones (19), el lado derecho contiene un producto entre p_{jik-1} y una suma de variables binarias (restadas entre sí), en consecuencia, el producto completo es una expresión cuadrática. Caso análogo, sucede con las restricciones (20). Mientras, en las restricciones (22) también aparecen productos entre variables binarias (p). Por todo esto, es necesario linealizar estas restricciones. Las restricciones (19) son reemplazadas por las restricciones (27), que verifican si no hay diferencia entre la cantidad de nodos defendidos en una ronda de defensa y la anterior. Entonces, p_{jik} toma el valor de 1 en el mismo nodo $i \in \bar{V}$ de la ronda $j \in R^B$ que la ronda de defensa anterior ($k-1 \in R^D \setminus \{1\}$). Luego, si no hay aumento de un nodo defendido entre una ronda de defensa y la siguiente, entonces, lo que está dentro de la sumatoria se hace 0 y p_{jik} toma el valor de p_{jik-1} .

$$p_{jik} \geq p_{jik-1} - \sum_{l \in \bar{V}} (\bar{d}_{jlk} - \bar{d}_{jlk-1}) \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \setminus \{1\} \quad (27)$$

De igual manera, las restricciones (20) son reemplazadas por las restricciones (28). Estas últimas restricciones son la aplicación análoga de lo anterior, cuando se trata de la primera ronda de defensa.

$$p_{ji1} \geq p_{j-1iD} - \sum_{l \in \bar{V}} (\bar{d}_{jl1} - d_{lj-1}) \quad \forall i \in \bar{V}, j \in R^B \quad (28)$$

Para la linealización de las restricciones (22) es necesario definir variables de decisión binaria adicionales como z_{jkl} ($j \in R^B, k \in R^D, l \in \bar{V}, i \in \bar{V}$), que representa el paso del bombero en la ronda de quema $j \in R^B$, en la ronda de defensa $k \in R^D$, desde el nodo $l \in \bar{V}$ al nodo $i \in \bar{V}$. Adicionalmente, se recurre al parámetro definido como M , conocido como Big-M, una constante suficientemente grande que permite modelar relaciones condicionales mediante desigualdades lineales. Este parámetro se utiliza para activar o desactivar restricciones en función del valor de variables binarias, facilitando la representación de comportamientos lógicos dentro del MILP. La elección adecuada de M es crítica, ya que un valor excesivamente grande puede generar inestabilidad numérica o aumentar los tiempos de cómputo de manera significativa.

Las restricciones (22) son reemplazadas por las nuevas restricciones (29) y (30), que requieren de las restricciones (31) para garantizar que solo sea uno el paso escogido por el bombero al ir del nodo $i \in \bar{V}$ al nodo $l \in \bar{V}$ en la ronda de quema $j \in R^B$ y en la ronda de defensa $k \in R^D$.

$$2 - (p_{ji1} + p_{j-1iD}) \leq M(1 - z_{jkl}) \quad \forall j \in R^B, i, l \in \bar{V} \quad (29)$$

$$2 - (p_{jik} + p_{jlk-1}) \leq M(1 - z_{jkl}) \quad \forall j \in R^B, i, l \in \bar{V}, k \in R^D \setminus \{1\} \quad (30)$$

$$\sum_{l \in \bar{V}} \sum_{i \in \bar{V}} z_{jkl} = 1 \quad \forall j \in R^B, k \in R^D \quad (31)$$

Adicionalmente, se requieren las restricciones (32) y (33) que garantizan que la variable z_{jkl} solo adicione tiempo de desplazamiento cuando hay un desplazamiento efectivo de un nodo $i \in \bar{V}$ al nodo $l \in \bar{V}$, lo que solo sucede una sola vez por ronda de defensa $k \in R^D$ en la ronda de quema $j \in R^B$.

$$t_j = t_{j-1} + \sum_{k \in R^D} \sum_{l \in \bar{V}} \sum_{i \in \bar{V}} \tau_{li} z_{jkli} \quad \forall j \in R^B \quad (32)$$

$$z_{jkli} \in \{0, 1\} \quad \forall i, l \in \bar{V}, \forall j \in R^B, \forall k \in R^D \quad (33)$$

Finalmente, el modelo propuesto de MILP considera la función objetivo (2) y las restricciones (5)–(18), (21) y (23)–(26) y las nuevas restricciones (27)–(33), utilizando $M = 2$.

3.2. Modelo de programación con restricciones

Dado que el problema abordado necesita ser modelado con condiciones lógicas, tanto el modelo de MILP como el modelo de MIQCP deben usar técnicas de linealización y reformulación para incorporar estas condiciones lógicas. Esto a menudo implica el uso de variables binarias adicionales y la inclusión de restricciones de tipo Big-M para traducir el comportamiento lógico en un formato algebraico. Como consecuencia, la complejidad del modelo aumenta significativamente, lo que impacta negativamente en el tiempo de resolución, particularmente en los modelos cuadráticos. En cambio, en CP, las variables se definen en dominios discretos finitos y admiten restricciones lógicas directas (Rossi et al., 2006). Esto reduce la necesidad de aproximaciones numéricas y facilita la representación del problema. En consideración de lo anterior, se busca pasar a restricciones lógicas a las restricciones que generan mayor dificultad computacional, es decir, las restricciones cuadráticas (19) y (20). Las restricciones cuadráticas (19) se reemplazan por las restricciones (34).

$$\text{if} \left(\sum_{l \in \bar{V}} (\bar{d}_{jlk} - \bar{d}_{jlk-1}) = 0 \wedge p_{jik-1} = 1 \right) \Rightarrow p_{jik} = 1 \quad \forall i \in \bar{V}, j \in R^B, k \in R^D \setminus \{1\} \quad (34)$$

Donde, si la cantidad de nodos $l \in \bar{V}$ en la ronda de quema $j \in R^B$, no aumenta de una ronda de defensa $k \in R^D \setminus \{1\}$ a su sucesor, entonces $p_{j,i,k}$ tomará el mismo valor que su antecesor con respecto a la ronda de defensa $k \in R^D \setminus \{1\}$. De igual manera, las restricciones (20) se reemplazan por las restricciones (35).

$$\text{if} \left(\sum_{l \in \bar{V}} (\bar{d}_{jl1} - d_{lj-1}) = 0 \wedge p_{j-1,iD} = 1 \right) \Rightarrow p_{ji1} = 1 \quad \forall i \in \bar{V}, j \in R^B \quad (35)$$

Este conjunto de restricciones es análogo al caso anterior, con la salvedad que está formulada para el caso de la primera ronda defensa, es decir, $k = 1$.

Dado que los modelos de CP no admiten variables continuas, es necesario discretizar la variable asociada al tiempo. Esto implica reemplazar las restricciones (22) por las restricciones (36).

$$t_j = t_{j-1} + \sum_{l \in \bar{V}} \sum_{i \in \bar{V}} p_{jil} \tau_i p_{j-1l} D \omega + \sum_{k \in R^D \setminus \{1\}} \sum_{l \in \bar{V}} \sum_{i \in \bar{V}} p_{jik} \tau_i p_{jlk-1} \omega \quad \forall j \in R^B \quad (36)$$

Donde, el parámetro ω actúa como un factor de escalamiento que permite aproximar los tiempos reales mediante valores enteros. Para que la discretización sea adecuada, ω debe ser suficientemente grande en comparación con las distancias del problema. Se usa $\omega = 1000$.

De igual manera, lo anterior afecta las restricciones (23), debiendo reemplazarse por las restricciones (37).

$$t_j \leq \omega j \quad \forall j \in R^B \quad (37)$$

Las restricciones (36) y (37) aseguran que el tiempo t_j requerido para defender una secuencia de nodos durante la $j \in R^B$ ronda de propagación del fuego no exceda la cantidad de intervalos de tiempo disponibles en dicha ronda. Finalmente, el modelo de CP propuesto considera la función objetivo (2) y las restricciones (3)–(6) y (24)–(26) y las nuevas restricciones (34)–(37).

3.3. Obtención de los parámetros B y D

Dado que los parámetros B y D repercuten en el tiempo de resolución debido a que aumentan el número de restricciones. En este trabajo, D se ha obtenido a partir del método propuesto por [Gutiérrez-De-La-Paz et al. \(2022\)](#), mientras que para B se propone un nuevo método. Así, un procedimiento para obtener el peor caso de cota superior de B es que el número de rondas de propagación del fuego sea el número de rondas, $B = |V|$. Sin embargo, debido al alto costo computacional que esto conlleva, se define una cota menor. Esta consiste en la longitud del camino más largo, encontrada mediante un algoritmo de enumeración de todos los caminos

simples, en términos de tiempo discreto de propagación, entre los nodos inicialmente incendiados y cualquier otro nodo aún no quemado.

El procedimiento para determinar la cota superior D , propuesto por [Gutiérrez-De-La-Paz et al. \(2022\)](#), se basa en un algoritmo recursivo de fuerza bruta que explora todas las posibles secuencias de defensa sobre un grafo conexo. En este contexto, D representa la longitud máxima alcanzable por un camino antes de que la suma de las distancias exceda 1. Así, este procedimiento evalúa rutas factibles bajo dos esquemas: permitiendo que cualquier nodo pueda ser el punto inicial, o desde el depósito como ancla para iniciar la ruta. Finalmente, el valor de D corresponde a la cantidad de nodos del camino más largo encontrado dentro del intervalo de tiempo considerado, según este análisis exhaustivo.

Capítulo 4

Metaheurística propuesta

Este capítulo presenta y describe el algoritmo propuesto que está basado en ILS y SA. Además, se describe la representación de la solución junto a la generación de la solución inicial y todos los componentes del algoritmo propuesto.

4.1. Algoritmo general

El algoritmo propuesto combina los principios de ILS (Lourenço et al., 2003) y SA (Kirkpatrick et al., 1983) para resolver de manera eficiente en tiempos de cómputo el MFP. La estrategia se basa en aplicar repetidamente una búsqueda local con dos tipos de intensidades, partiendo desde soluciones perturbadas que permiten explorar nuevas regiones del espacio de soluciones. Para evitar quedar atrapado en óptimos locales, se incorpora un mecanismo usado habitualmente en el SA, donde ciertas soluciones de menor calidad pueden ser aceptadas con una probabilidad que depende de un parámetro de temperatura. Esta probabilidad decrece gradualmente a lo largo del proceso, favoreciendo la exploración en las primeras etapas y la intensificación en fases finales. La combinación de ambas técnicas permite equilibrar la diversificación y la intensificación, mejorando la capacidad del algoritmo para aproximarse a mejores soluciones.

El funcionamiento del ILS-SA es descrito en el Algoritmo 1. El proceso comienza con la generación de una solución inicial mediante una heurística greedy que es guardada como solución actual (s), solución candidata (\bar{s}) y mejor solución encontrada (s^*). Luego, en la línea 2, se inicializa el contador de iteraciones y la temperatura se fija en su valor inicial. Posteriormente, la solución inicial es sometida a un proceso iterativo

(líneas 3-15), con un número de iteraciones g menor que el máximo permitido I de iteraciones y que la temperatura no sea igual a la temperatura final. Dentro del ciclo, se aplican primero las perturbaciones bajo los parámetros β de probabilidad de elección (línea 4). A continuación, según las probabilidades γ , se puede aplicar alguna de las búsquedas locales (línea 5). Luego, se aplica un criterio de aceptación: si la solución candidata presenta una mejor evaluación que la solución actual, esta última se actualiza con la solución candidata, almacenándola como la solución actual (líneas 6-7). Posteriormente, si la solución candidata es mejor que la mejor solución, entonces, esta última se actualiza con la solución candidata. Cuando el costo de la solución candidata es igual al costo de la mejor solución encontrada, entonces, se almacena como la nueva mejor solución, solo si el tiempo en que se recorren los nodos (hasta el último nodo defendido o protegido) es menor que el tiempo correspondiente a la mejor solución previamente almacenada (líneas 8-9).

Algoritmo 1 Pseudocódigo para ILS-SA

Salida: s^*

```

1:  $s, s^*, \bar{s} \leftarrow \text{generar-solucion-inicial}()$ 
2:  $g \leftarrow 0; t \leftarrow T^i$ 
3: while  $g < I$  and  $t > T^f$  do
4:    $\bar{s} \leftarrow \text{perturbación}(s, \beta)$ 
5:    $\bar{s} \leftarrow \text{busqueda-local}(\bar{s}, \gamma)$  ▷ Aceptación de mejor solución
6:   if  $f(\bar{s}) < f(s)$  then
7:      $s \leftarrow \bar{s}$ 
8:     if  $f(\bar{s}) \leq f(s^*)$  then
9:        $s^* \leftarrow \bar{s}$ 
10:  else if  $\text{random}(0, 1) < e^{-\frac{|f(\bar{s}) - f(s^*)|}{t}}$  then ▷ Diversificación de solución
11:     $s \leftarrow \bar{s}$ 
12:     $t \leftarrow \alpha \cdot t$ 
13:    if  $g = \phi$  then
14:       $t \leftarrow T^i$ 
15:     $g \leftarrow g + 1$ 

```

Adicionalmente, se incluye un criterio de diversificación: con base en una probabilidad entre 0 y 1, una función exponencial de la diferencia de evaluaciones entre soluciones, dividida por la temperatura t , esto corresponde al criterio de Metropolis ([Metropolis et al., 1953](#)) que decide si la solución candidata puede ser aceptada para la siguiente iteración (líneas 10-11). Posteriormente, la temperatura se reduce en un factor α (línea 12). En las líneas 13-14, si el número de iteraciones alcanza las ϕ iteraciones, entonces, la temperatura se reinicia a la temperatura inicial. Luego, el contador de

iteraciones g es incrementado (línea 15). Finalmente, el algoritmo propuesto retorna la mejor solución.

4.2. Representación y evaluación de la solución

La representación de la solución utilizada está compuesta por una tupla $s = (\pi^1, \pi^2)$. π^1 corresponde a una lista de tamaño $|V|$, que consiste en el orden de visita de cada nodo y π^2 es la posición del último nodo en ser defendido o protegido. Aunque el bombero tiene la capacidad de desplazarse a cualquier nodo, ciertas secuencias de visita permiten maximizar la cantidad de nodos protegidos. En este sentido, una solución eficiente prioriza la visita a nodos estratégicos que contribuyan a la defensa de la red, relegando al final aquellos que ya están quemados o protegidos de manera indirecta.

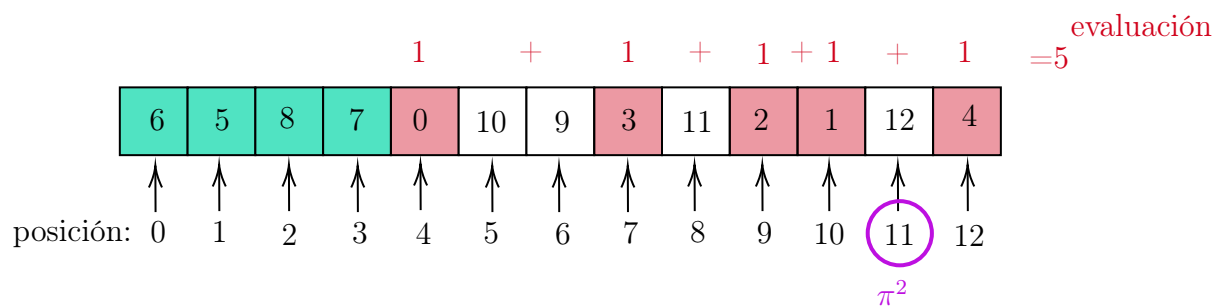


Figura 2: Representación de la solución.

La Figura 2 muestra una representación de una solución factible del problema presentado en la Figura 1. En ella, la lista de solución indica un recorrido por los nodos 6, 5, 8 y 7, los cuales son defendidos exitosamente. Como resultado, solo los nodos 0, 1, 2, 3 y 4 terminan quemados, mientras que el resto queda protegido debido a la acción previa del bombero. De esta forma, el valor de la función de evaluación es 5. Además, el valor de π^2 es 11, puesto que es la posición del nodo 12 que está protegido.

La representación de la solución debe ser decodificada mediante la función evaluación de la solución y propagación del fuego. Para ello, se define una función de evaluación que evalúa cada elemento de la lista de solución, que es presentada en el Algoritmo 2. Este algoritmo tiene como parámetros de entrada el grafo G , los nodos quemados inicialmente F y la solución s . En la línea 1, se define un valor de función de evaluación f como el número de nodos $|V|$. Luego, por cada nodo $i \in \{1, \dots, V\}$, se

verifica si corresponde a el primer elemento de la lista solución (línea 3) o a uno posterior (línea 11). A la lista de quemados actuales inicialmente se le asigna la lista de quemados en un inicio (línea 4). Luego, se almacena el tiempo de desplazamiento entre el nodo inicial y el primer nodo en el vector w (línea 5). donde N corresponde al número de rondas a propagar el fuego, tomando el valor truncado hacia el entero más pequeño (línea 6). A continuación, se llama a la función **propagar-fuego** que propaga el fuego a partir del grafo G , la lista de quemados actuales y un número de rondas N obtenido anteriormente. Posteriormente, si el nodo no está quemado o si la solución es defendida exactamente en el mismo tiempo que alcanza a quemarse, es decir, el tiempo de desplazamiento es entero y el nodo se encuentra dentro del conjunto de quemados en la última ronda (F^u), entonces, prevalece el nodo como defendido y a la función de evaluación f se le resta uno, junto con restar el nodo del grafo (líneas 8-10). Para los nodos visitados que no correspondan al primero, en la línea 11, se sigue un proceso análogo al anterior, con la salvedad que w almacena el tiempo de desplazamiento entre un nodo y el siguiente en la lista, junto a los tiempos ya recorridos, propagando el fuego solo en la diferencia de rondas entre el nodo actual y el anterior. El proceso termina cuando ya se han visitado todos los nodos de la lista solución.

Algoritmo 2 Función de evaluación de la solución

Entrada: G, F, s

Salida: f

```

1:  $f \leftarrow |V|$ 
2: for  $i \in \{1, \dots, V\}$  do
3:   if  $i = 0$  then
4:      $F_0^a \leftarrow F$ 
5:      $w_i \leftarrow \tau(a, s_0)$ 
6:      $N \leftarrow \lfloor w_i \rfloor$ 
7:      $F_{i+1}^a, F^u \leftarrow \text{propagar-fuego}(G, F_i^a, N)$ 
8:     if  $s_i \notin F_{i+1}^a$  or  $(w_i = \lfloor w_i \rfloor \text{ and } s_i \in F^u)$  then
9:        $f \leftarrow f - 1$ 
10:       $G \leftarrow G \setminus s_i$ 
11:   else
12:      $w_i \leftarrow w_{i-1} + \tau(s_i, s_{i-1})$ 
13:      $N \leftarrow \lfloor w_i \rfloor - \lfloor w_{i-1} \rfloor$ 
14:      $F_{i+1}^a, F^u \leftarrow \text{propagar-fuego}(G, F_i^a, N)$ 
15:     if  $s_i \notin F_{i+1}^a$  or  $(w_i = \lfloor w_i \rfloor \text{ and } s_i \in F^u)$  then
16:        $f \leftarrow f - 1$ 
17:       $G \leftarrow G \setminus s_i$ 

```

La función de propagar-fuego es presentada en el Algoritmo 3. Recibe como parámetros de entrada: el grafo G , el conjunto de nodos inicialmente quemados F , y un número de rondas N que determinan la duración de la propagación. En la línea 1, se inicializan los nodos actualmente quemados F^a y los nuevos nodos quemados F^n , ambos iguales a F al inicio. Se agrega un conjunto vacío que sirve para almacenar los nodos quemados en la última ronda (línea 2). A partir de la línea 3, por cada ronda de propagación hasta N , se itera sobre cada nodo actualmente quemado. Si el nodo pertenece al grafo, se revisan sus vecinos. Si alguno de estos vecinos no ha sido quemado previamente (línea 4), se agrega al conjunto de nuevos nodos quemados (línea 5). Si el nuevo nodo quemado, se quema durante la última ronda, se agrega a F^u (líneas 6-7). Si al finalizar una ronda no hay nuevos nodos que se hayan quemado (líneas 8-9), el proceso se interrumpe. Posteriormente, los nuevos nodos quemados se incorporan a los nodos actualmente quemados (línea 10), y la lista de nuevos quemados se reinicia para la siguiente ronda. Por último, la función retorna los nodos quemados actuales y los nodos quemados en la última ronda de propagación.

Algoritmo 3 Función de propagación del fuego

Entrada: G, F, N

Salida: F^a, F^u

```

1:  $F^a \leftarrow F^n \leftarrow F$ 
2:  $F^u \leftarrow \emptyset$ 
3: for  $r \in \{1, \dots, N\}, i \in F^a : i \in G$  do
4:   for  $v \in N(i) : i \notin F^a$  do
5:      $F^n \leftarrow F^n \cup \{v\}$ 
6:     if  $i = R^f$  then
7:        $F^u \leftarrow F^u \cup \{v\}$ 
8:   if  $F^n = \emptyset$  then
9:     break
10:   $F^a \leftarrow F^a + F^n$ 
11:   $F^n \leftarrow \emptyset$ 

```

4.3. Solución inicial

El Algoritmo 1 comienza desde una solución inicial que es construida mediante una heurística que crea una lista de recorrido de los nodos a partir de la selección aleatoria de estos. La heurística, en cada iteración revisa los nodos disponibles, es decir, aquellos que pueden ser defendidos o protegidos, excluyendo aquellos ya recorridos (quemados, defendidos o protegidos). Si no hay nodos disponibles, se termina la heurística y se

completa la solución con los nodos restantes. Si hay nodos disponibles, se selecciona uno aleatoriamente, se actualizan las distancias acumuladas y se simula la propagación del fuego. Si el nodo cumple las condiciones de elegibilidad (no estar quemado, ni haber sido visitado), se añade al conjunto solución y se actualizan las estructuras. El proceso continúa hasta que no se puedan seleccionar más nodos válidos. Una vez alcanzado este punto, se completa la lista con los nodos restantes, asegurando que la lista de soluciones esté correctamente representada.

4.4. Perturbaciones

Se definieron cinco perturbaciones que son llamadas desde el Algoritmo 1. Así, solo se aplica uno de los cinco procedimientos de perturbación. La elección del procedimiento de perturbación es determinado por un valor generado aleatoriamente entre 0 y 1, que corresponde a las probabilidades específicas asociadas a cada perturbación, denotadas entre β_1 y β_5 . Es importante destacar que estas probabilidades son parámetros de ajuste, y su suma está limitada por la condición $\beta_1 + \beta_2 + \beta_3 + \beta_4 + \beta_5 = 1$. A continuación, se describen los procedimientos de perturbación.

perturbation-1: Selecciona de manera aleatoria un nodo u de π^1 , y luego, de igual manera selecciona otro nodo v de π^1 . Finalmente, el nodo u es intercambiado por el nodo v .

perturbation-2: Selecciona un segmento de π^1 de manera aleatoria entre un nodo u y un nodo v ($u < v$) dentro del rango de 0 hasta π^2 . Luego, se inserta cada elemento en una posición menos dentro de la lista.

perturbation-3: A partir de π^1 , se selecciona de manera aleatoria un nodo u dentro del rango de 0 hasta π^2 . Luego, un nodo v también aleatoriamente seleccionado pero fuera del rango, $[\pi^2, \pi^1]$. Finalmente, el nodo u es intercambiado por el nodo v .

perturbation-4: A partir de π^1 , se selecciona de manera aleatoria un nodo u dentro del rango de 0 hasta π^2 . Luego, un nodo v también aleatoriamente seleccionado dentro del mismo rango. Finalmente, el nodo u es intercambiado por el nodo v .

perturbation-5: A partir de π^1 , se selecciona de manera aleatoria un nodo u dentro del rango de 0 hasta π^2 . Luego, el nodo u es intercambiado con el siguiente nodo en la lista de π^1 .

4.4.1. Búsqueda local

La elección del procedimiento de búsqueda local es determinado por un valor generado aleatoriamente entre 0 y 1, que corresponde a las probabilidades específicas asociadas a no aplicar una búsqueda local, o aplicar **LS-1** o **LS-2**, denotadas entre γ_1 y γ_3 , respectivamente. Es importante destacar que estas probabilidades son parámetros de ajuste, y su suma está limitada por la condición $\gamma_1 + \gamma_2 + \gamma_3 = 1$. A continuación, se describen los procedimientos de búsqueda local.

- Si la probabilidad está dentro del rango generado por γ_1 , entonces, no se aplica una búsqueda local.
- **LS-1** implementa una estrategia de búsqueda local para mejorar la solución candidata \bar{s} . Primero, evalúa \bar{s} obteniendo su costo actual. Posteriormente, selecciona aleatoriamente un nodo u dentro del rango de los nodos hasta donde se han defendido o protegido (π^2). Luego, se intercambia el nodo u con los nodos fuera de este rango, uno por uno. Cada nueva solución generada se evalúa, y si mejora el valor de la función de evaluación, se detiene la búsqueda y se retorna. Caso contrario, se mantiene la solución original y se sigue iterando.
- **LS-2** implementa una estrategia similar a la primera búsqueda local **LS-1**. Sin embargo, a diferencia de **LS-1**, intenta con todos los nodos de la lista uno por uno hasta encontrar una mejora o hasta el último nodo en proteger o defender, por lo que es un proceso más exhaustivo.

Capítulo 5

Resultados

Este capítulo presenta las instancias, hardware y software de la implementación de parámetros del algoritmo propuesto, los resultados obtenidos de los modelos y el algoritmo propuesto, seguidos de una discusión detallada y una comparación con el mejor algoritmo disponible en la literatura.

5.1. Instancias

Se generaron un total de 160 instancias para la experimentación utilizando el generador de instancias propuesto por [Gutiérrez-De-La-Paz et al. \(2022\)](#). Para estos experimentos, se construyeron instancias (G, F, a, τ, f, T) del MFP, donde el número de bomberos f es igual a uno, la duración de los intervalos de tiempo T también es uno, y el número de incendios iniciales F . El grafo $G = (V, E)$, con $|V| = n \in \{10, 20, 30, 40\}$.

El conjunto de 160 instancias se obtuvo a partir de 10 grafos generados para cada valor de $|V|$. Para cada uno de estos grafos, se generaron dos variantes con 1 y 3 incendios iniciales, respectivamente, considerando dos valores del factor de escalamiento de distancias euclidianas, $\lambda = 2$ y $\lambda = 5$, que aumentan las distancias entre nodos. Por último, el parámetro τ es calculado a partir de las distancias euclidianas entre las ubicaciones de los nodos y el punto de salida del bombero. Adicionalmente, se muestra a modo de ejemplo visual de instancia, los resultados obtenidos para la instancia número 47 en el Anexo.

5.2. Configuración de los experimentos

Los modelos de MIQCP y MILP fueron implementados utilizando el solver Gurobi 9.5, mientras que el modelo de CP fue desarrollado con el solver CPLEX 12.9. Todos los modelos y algoritmos fueron programados en Python 3.9. Los experimentos se llevaron a cabo en la infraestructura de supercomputación del NLHPC, utilizando un nodo Lenovo ThinkSystem SR645 V3 equipado con dos procesadores AMD EPYC 9754 a 2,25 GHz, cada uno con 128 núcleos, y 768 GB de memoria RAM. Todas las ejecuciones se realizaron utilizando un solo hilo, bajo el sistema operativo CentOS Linux 7 (64 bits).

Todas las instancias utilizadas para la evaluación, junto con los resultados detallados para cada conjunto de instancias y el código fuente, se encuentran disponibles en línea en la siguiente URL.¹

5.3. Parámetros de ILS-SA

El algoritmo propuesto tiene 11 parámetros. Los parámetros $\beta_1, \beta_2, \beta_3, \beta_4$ y β_5 regulan la probabilidad de ingreso a las perturbaciones, siguiendo un orden decreciente en términos de aleatoriedad. Los parámetros γ_1, γ_2 y γ_3 están asociados a la búsqueda local. Adicionalmente, los parámetros α y T^i corresponden a la reducción de temperatura por iteración y a la temperatura de entrada, respectivamente. Finalmente, el parámetro ϕ indica el número de iteraciones en que la temperatura vuelve a ser la inicial.

Dada la elevada cantidad de parámetros involucrados, una calibración adecuada permite un desempeño computacional eficiente. En particular, es crucial encontrar una buena combinación de parámetros de la metaheurística propuesta. En este estudio, se emplea un enfoque eficiente para la calibración de parámetros mediante la herramienta irace, que corresponde a un sistema de configuración automática desarrollado en R (López-Ibáñez et al., 2016). El objetivo principal de irace es automatizar el proceso de configuración de parámetros de los algoritmos de optimización en función de su desempeño sobre un conjunto de instancias. Así, irace utiliza una estrategia de eliminación progresiva de configuraciones poco prometedoras mediante pruebas estadísticas, enfocando la búsqueda en las regiones más prometedoras del espacio de parámetros.

¹<https://11nq.com/2nACj>

Para la selección de una buena combinación de parámetros, se utiliza un subconjunto de 10 instancias basado en resultados preliminares de ILS-SA. Para ello, se fija una cantidad máxima de 30000 iteraciones y un tiempo máximo de 60 segundos por instancia. En total, el proceso de calibración de irace requirió de aproximadamente 70,13 horas de ejecución, utilizando 40 hilos del computador anteriormente mencionado. La Tabla 2 presenta los parámetros evaluados, rangos de búsqueda y valores finales obtenidos tras la calibración.

Tabla 2: Parámetros utilizados en el algoritmo y sus valores calibrados.

Parámetro	Rango	Valor definido
β_1	[0,1]	0,2700
β_2	[0,1]	0,0725
β_3	[0,1]	0,0230
β_4	[0,1]	0,2393
β_5	[0,1]	0,3952
γ_1	[0,1]	0,1794
γ_2	[0,1]	0,1457
γ_3	[0,1]	0,6749
α	[0,9500, 1,0000]	0,9972
T^i	[1,1000]	778,0895
ϕ	[1,30000]	1863

5.4. Resultados de los modelos

Esta sección presenta los resultados computacionales de los modelos propuestos de MILP y CP, comparándolos con el modelo de MIQCP. Los modelos son ejecutados con un tiempo límite de 3600 segundos. La Tabla 3 presenta los resultados sobre las 160 instancias. Por cada fila, se presentan los promedios de las 40 instancias con la cantidad de nodos indicada en la primera columna ($|V|$), los cuales varían de 10, 20, 30 y 40. Luego, cada grupo de columnas proporciona información sobre el gap promedio porcentual de optimalidad (GAP^{LB}) (Laporte & Toth, 2022) (es decir, el porcentaje de la diferencia entre la mejor cota superior y la mejor cota inferior encontradas al final del proceso de resolución), el gap promedio porcentual de la cota superior (GAP^{UB}) (es decir, el porcentaje de diferencia entre la mejor cota superior encontrada y la mejor solución conocida BKS²), el tiempo de cómputo

²BKS, por sus siglas en inglés Best Known Solution

correspondiente (expresado en segundos) y finalmente Hits o el número de aciertos, que indica las instancias en que se sabe que se alcanza el valor del BKS. Este formato se repite por cada modelo. Cabe destacar que el costo BKS es calculado como el costo mínimo obtenido entre todos los modelos y algoritmos reportados para cada instancia. Finalmente, la última fila muestra los promedios de cada una de las columnas anteriormente mencionadas, exceptuando en la columna de Hits, que corresponde a la suma de los hits.

Tabla 3: Comparación entre los modelos de MIQCP, MILP y CP.

V	MIQCP				MILP				CP			
	GAP ^{LB} (%)	GAP ^{UB} (%)	tiempo (s)	Hits	GAP ^{LB} (%)	GAP ^{UB} (%)	tiempo (s)	Hits	GAP ^{LB} (%)	GAP ^{UB} (%)	tiempo (s)	Hits
10	0,00	0,00	0,29	40	0,00	0,00	0,21	40	0,00	0,00	0,20	40
20	1,73	0,00	323,27	40	0,23	0,00	156,14	40	0,00	0,00	127,92	40
30	17,87	17,09	1773,16	31	6,25	3,67	1058,16	37	14,45	0,00	1026,36	40
40	38,41	92,35	2633,08	17	25,49	51,91	1977,82	22	33,00	0,00	1822,35	40
Promedio	14,50	27,36	1182,45	128	7,99	13,90	798,08	139	11,86	0,00	744,21	160

Los resultados en las instancias más pequeñas ($|V| = 10$) muestran que todos los modelos logran resolver de manera óptima y en menos de un segundo en promedio, como se observa en la Tabla 3. Además, todos los modelos alcanzan la optimalidad (40 hits en MIQCP, MILP y CP). A medida que el tamaño del problema aumenta, se observa una degradación en el rendimiento de los modelos. Sin embargo, la magnitud de esta degradación varía entre ellos, reflejando diferencias en su capacidad para aproximarse a los BKSs y en su eficiencia computacional. El modelo de MIQCP muestra un comportamiento variable dependiendo del tamaño de las instancias. Para $|V| = 20$, el GAP^{LB} es 1,73 %, pero a medida que el tamaño aumenta, los GAPs crecen significativamente. Esto sugiere que el modelo enfrenta problemas para alcanzar buenas soluciones enteras cercanas al BKS en instancias más grandes y también demostrar optimalidad según su GAP^{LB}. En cuanto a su GAP^{UB}, en particular, para $|V| = 40$, se alcanza un valor de 92,35 %, lo que evidencia dificultades para aproximarse al BKS. Además, la cantidad de hits disminuye considerablemente con el incremento del tamaño, pasando de 40 hits para $|V| = 10$ a solo 17 hits para $|V| = 40$.

El modelo de MILP presenta mejoras en comparación con MIQCP. En general, los GAPs son menores, el GAP^{LB} es 25,49 % y el GAP^{UB} es 51,91 % en $|V| = 40$, respecto a MIQCP con 38,41 % y 92,35 %, respectivamente. Con respecto al GAP^{UB}, esto sugiere que, aunque el modelo muestra un desempeño más consistente en instancias

grandes, aún no logra alcanzar los BKSs en todos los casos. Además, los tiempos de cómputo son inferiores, lo que indica una mayor eficiencia computacional. Sin embargo, los hits solo aumentan levemente, con 17 hits de MIQCP a 22 hits en MILP.

El modelo de CP destaca su capacidad para encontrar soluciones enteras cercanas a los BKSs en todas las instancias, notar que esto ocurre de igual manera en instancias grandes. En todas las instancias se observa que el GAP^{UB} es 0,00 %, lo que indica que se han encontrado soluciones enteras que coinciden con los BKSs. Además, los hits alcanzan todas las instancias. Esto refuerza la robustez del modelo ante instancias más complejas, consolidándolo como una alternativa efectiva en la resolución del problema. Sin embargo, los $GAPs^{LB}$ del modelo de CP son más altos que los obtenidos por MILP. Esto se alinea con las dificultades que tiene el enfoque de CP para demostrar la optimalidad de las soluciones.

En términos de tiempos de cómputo, todos los modelos muestran tiempos muy bajos en $|V| = 10$. Esto indica que las instancias pequeñas pueden resolverse de manera eficiente sin diferencias significativas entre los enfoques. Sin embargo, conforme el tamaño aumenta, el tiempo de cómputo también se incrementa en los tres modelos. MIQCP es el que más tiempo requiere, alcanzando los 2633,08 segundos para 40 nodos. En comparación, MILP tiene un tiempo de 1977,82 segundos, mientras que CP alcanza los 1822,35 segundos, lo que indica que estos modelos son más eficientes en términos de tiempo de cómputo.

Aunque CP tiene un tiempo de ejecución similar a MILP en instancias grandes, su desempeño es superior en términos de calidad de solución entera. En particular, presenta mejores valores de $GAPs^{UB}$ y hits, lo que lo convierte en una alternativa más robusta para encontrar soluciones cercanas al BKS en problemas de mayor tamaño. Aunque no es tan buena demostrando optimalidad como MILP según su gap porcentual de optimalidad. Esto es debido a que el enfoque CP está diseñado principalmente para encontrar eficientemente soluciones factibles que satisfagan todas las restricciones dadas (Rossi et al., 2006).

5.5. Resultados de la metaheurística

Esta sección compara el algoritmo propuesto, ILS-SA, con los tres modelos MIQCP, MILP y CP. Con el fin de realizar una comparación justa, los tres modelos y el algoritmo propuesto son ejecutados con un tiempo límite de 600 segundos por

instancia. Se considera el gap porcentual de la cota superior, tiempo de cómputo y número de aciertos del valor de BKSs (Hits) para las 160 instancias consideradas. Además, ILS-SA considera un máximo de 30000 iteraciones y cada instancia es ejecutada 10 veces con semillas independientes.

La Tabla 4 resume los resultados obtenidos de ILS-SA y los tres modelos, permitiendo visualizar el desempeño relativo de cada enfoque frente a diferentes tamaños de instancia. Se organiza en cinco grupos de columnas, en la primera columna $|V|$, representa el número de nodos de cada grupo de instancias ($|V| = \{10, 20, 30, 40\}$). En el caso de MIQCP, la primera columna muestra los resultados del gap porcentual de integralidad (es decir, el porcentaje de diferencia entre la mejor cota superior encontrada y el BKS), el tiempo de cómputo correspondiente (expresado en segundos) y la columna de hits, que indica el número de instancias en las que se alcanza el costo BKS. Este formato es el mismo para las columnas de MIQCP, MILP y CP. A continuación, la columna de ILS-SA presenta el gap porcentual promedio de las 10 ejecuciones por instancia (GAP^{avg}), seguido por el gap porcentual mínimo alcanzado entre 10 ejecuciones (GAP^{best}), el tiempo de cómputo correspondiente (expresado en segundos) y la columna de Hits. Finalmente, en la última fila se muestran los promedios correspondientes a cada columna, exceptuando en la columna de Hits, que corresponde a la suma de los hits.

Tabla 4: Comparación entre los métodos MIQCP, MILP, MCP e ILS-SA.

$ V $	MIQCP			MILP			CP			ILS-SA			
	$GAP^{UB}(\%)$	tiempo (s)	Hits	$GAP^{UB}(\%)$	tiempo (s)	Hits	$GAP^{UB}(\%)$	tiempo (s)	Hits	$GAP^{avg}(\%)$	$GAP^{best}(\%)$	tiempo (s)	Hits
10	0,00	0,30	40	0,00	0,21	40	0,00	0,20	40	0,06	0,00	7,22	40
20	0,25	90,82	38	1,73	44,28	38	0,00	38,21	40	0,53	0,00	52,40	40
30	52,62	372,52	23	16,08	304,55	27	0,21	254,50	39	2,33	0,35	188,29	38
40	100,82	545,33	11	55,81	405,47	19	11,34	379,07	28	5,64	1,88	386,72	29
Promedio	38,42	252,24	112	18,40	188,63	124	2,89	167,99	147	2,14	0,56	158,66	147

ILS-SA evidencia un desempeño competitivo en comparación con los modelos exactos MIQCP, MILP y CP, como muestra la Tabla 4. A medida que aumenta el tamaño de las instancias, se observa un deterioro en los gaps porcentuales para los modelos, especialmente en MIQCP y MILP, cuyos gaps promedio alcanzan 52,62 % y 16,08 % para $|V| = 30$, y superan el 100 % y 55 %, respectivamente para $|V| = 40$. En contraste, CP e ILS-SA mantienen gaps promedio considerablemente más bajos (0,21 % y 2,33 % para instancias de 30 nodos y 11,34 % y 5,64 % para instancias de 40 nodos, respectivamente). En cuanto a CP, con valores mínimos que en muchos casos alcanzan el óptimo (GAP^{best} con 0,00 % para instancias de 10 y 20 nodos).

Además, el número total de hits logrados por CP iguala a ILS-SA (147) superando a MIQCP y MILP. Si bien el tiempo de cómputo de ILS-SA crece con el tamaño de las instancias, permanece dentro del límite establecido y resulta competitivo frente a los modelos, los cuales, en muchos casos, agotan el tiempo de cómputo sin alcanzar el BKS. Notar que para instancias más grandes (40 nodos), el gap promedio porcentual resulta ser el menor, con 5,64% y que se reduce a un 1,88% para los mejores casos. Con resultados de gap porcentual promedio similares 2,89% y 2,14% para CP e ILS-SA, respectivamente. Pero con gaps porcentuales asociados al mínimo de 1,08% en promedio. En conjunto, estos resultados respaldan la efectividad y robustez de ILS-SA como alternativa eficiente para abordar instancias de mayor tamaño.

Los gaps porcentuales por cada método utilizado con tiempo límite de 600 segundos se presentan en diagramas de cajas en la Figura 3. Para el ILS-SA se presentan los resultados de gap porcentual mínimo. En general, los resultados muestran que los modelos MIQCP y MILP presentan una alta dispersión en sus soluciones, con múltiples instancias en las que el gap respecto al BKS es considerablemente elevado, alcanzando incluso valores atípicos extremos, especialmente para grafos de mayor tamaño. Por el contrario, CP e ILS-SA logran desempeños notablemente más estables y cercanos al BKS, mostrando un comportamiento muy similar entre ambos, aunque con una ligera ventaja para ILS-SA en cuanto a menor presencia de valores atípicos.

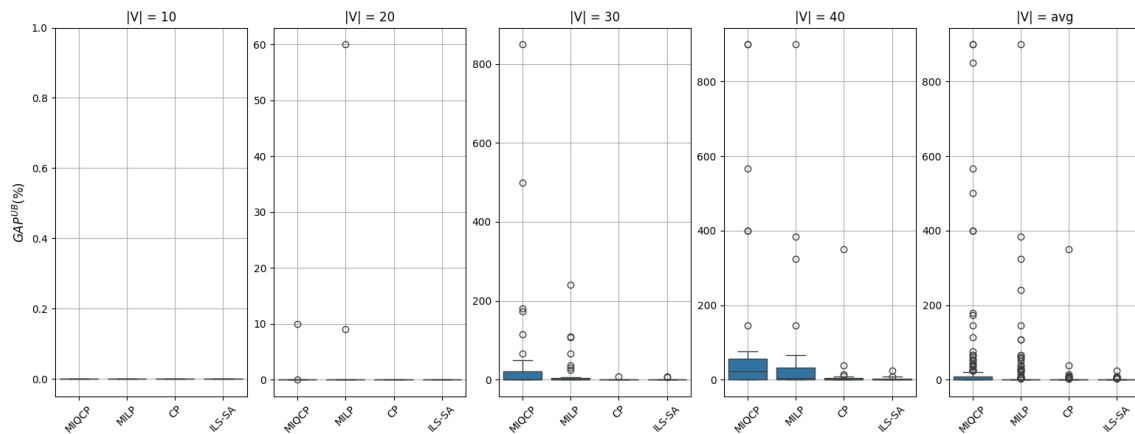


Figura 3: Gráficos de caja de los tres modelos e ILS-SA.

5.6. Análisis de los componentes de ILS-SA

Esta sección presenta la comparación de ILS-SA con siete versiones del ILS-SA con diferentes parámetros. Para una comparación justa, realizamos los experimentos

adicionales ejecutando todos los algoritmos diez veces de forma independiente, con semillas aleatorias diferentes para cada instancia. Se establece un límite de tiempo de 600 segundos y 30000 iteraciones por cada instancia.

A continuación, se detallan las siete versiones de los algoritmos utilizados en esta sección.

- **ILS-SA1:** Considera todas probabilidades equiprobables de las perturbaciones y búsqueda local ($\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0,2$, $\gamma_1 = \gamma_2 = \gamma_3 = 0,3333$). Mientras, los parámetros asociados a la temperatura son los mismos del ILS-SA.
- **ILS-SA2:** Considera las probabilidades de perturbaciones equiprobables ($\beta_1 = \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0,2$). Los parámetros asociados a la búsqueda local y a la temperatura son los mismos del ILS-SA.
- **ILS-SA3:** Considera las probabilidades de la búsqueda local equiprobables ($\gamma_1 = \gamma_2 = \gamma_3 = 0,3333$). Los demás parámetros se mantienen igual a ILS-SA.
- **ILS-SA4:** No considera búsquedas locales ($\gamma_1 = 1$, $\gamma_2 = \gamma_3 = 0$). Los demás parámetros se mantienen iguales.
- **ILS-SA5:** Considera solo la búsqueda local con más alta probabilidad LS2 ($\gamma_1 = \gamma_2 = 0$, $\gamma_3 = 1$). Los demás parámetros se mantienen igual a ILS-SA.
- **ILS-SA6:** Considera solo la perturbación con más alta probabilidad ($\beta_1 = \beta_2 = \beta_3 = \beta_4 = 0$, $\beta_5 = 1$). Los demás parámetros se mantienen iguales a ILS-SA.
- **ILS-SA7:** No considera los criterios de SA, reducción de temperatura ($\alpha = 10^{-6}$), y no hay reseteo de la temperatura ($\phi = 30000$). Los demás parámetros se mantienen iguales a ILS-SA.

Para evaluar el rendimiento del ILS-SA, se realiza una comparación con distintas configuraciones (ILS-SA1–ILS-SA7) que se presentan en la Tabla 5. Las métricas utilizadas para esta comparación fueron el gap porcentual promedio y el mejor, junto al número de hits, el tiempo promedio y p -value para el test de Wilcoxon (Wilcoxon, 1945).

Tabla 5: Resultados comparativos entre algoritmos propuestos e ILS-SA.

Algoritmo	GAP ^{avg} (%)	GAP ^{best} (%)	Hits	tiempo (s)	<i>p</i> -value
ILS-SA1	3,51	1,71	134	90,94	$5,00 \times 10^{-5}$
ILS-SA2	2,59	1,01	144	141,50	$1,74 \times 10^{-2}$
ILS-SA3	2,88	0,89	143	110,07	$3,89 \times 10^{-3}$
ILS-SA4	11,77	5,19	108	4,34	$2,21 \times 10^{-10}$
ILS-SA5	1,91	0,49	148	193,68	$7,55 \times 10^{-1}$
ILS-SA6	29,51	4,81	98	142,13	$8,39 \times 10^{-12}$
ILS-SA7	5,33	2,53	131	46,05	$4,91 \times 10^{-6}$
ILS-SA	2,14	0,56	147	158,66	–

Los resultados muestran que el algoritmo ILS-SA presenta un desempeño general mejor que todos, exceptuando ILS-SA5. Con un GAP^{avg} de 2,14 %, un GAP^{best} de 0,56 % y el número máximo de hits 147. Al comparar ILS-SA con los otros algoritmos, destaca que ILS-SA5 obtuvo un GAP^{avg} de 1,91 %, un GAP^{best} de 0,49 %, con 148 hits, aunque con tiempos de cómputo mayores (193,68 segundos contra los 158,66 segundos de ILS-SA). Esta versión emplea exclusivamente el procedimiento LS-2 dentro de la búsqueda local, sugiere que dicha estrategia es especialmente eficaz para mejorar la calidad de las soluciones.

Por otra parte, los algoritmos ILS-SA1, ILS-SA2 y ILS-SA3, con distintas combinaciones de condiciones equiprobables, muestran rendimientos intermedios, con GAPs promedio más altos y menos hits en comparación con ILS-SA. Asimismo, ILS-SA7 que mantiene todos los parámetros del ILS-SA sin los criterios de SA, logra un rendimiento intermedio en comparación a ILS-SA, con un GAP^{avg} de 5,33 %, un GAP^{best} de 2,53 % y 131 hits. Aunque con un tiempo de cómputo promedio mucho menor (46,05 segundos). En contraste, ILS-SA4, que desactiva completamente la búsqueda local, tiene bastante mal desempeño (GAP^{avg} de 11,77 % y solo 108 hits), lo que enfatiza la importancia crítica de contar con una etapa de búsqueda local para mejorar las soluciones en este tipo de problemas. El peor desempeño lo obtiene ILS-SA6, que aplica solo una de las perturbaciones (`perturbation-5`), lo que muestra que la aplicación de las diferentes perturbaciones contribuye a la mejora

de la solución obtenida.

En lo que se refiere a los tiempos de cómputo, se aprecia que ILS-SA1, ILS-SA2, ILS-SA3, ILS-SA4, ILS-SA6 e ILS-SA7 tienen tiempos de cómputo menores que ILS-SA, sin embargo sus gaps porcentuales de integralidad son menores, además ILS-SA4 presenta tiempos mucho menores que ILS-SA, con peores resultados. Solo ILS-SA5 tiene tiempos de cómputo bastante mayores aunque con unos resultados levemente mejores.

En cuanto al test de Wilcoxon, este test para rangos con signo es una prueba no paramétrica utilizada para comparar dos muestras relacionadas cuando no se puede asumir normalidad. El método se basa en las diferencias entre pares de datos, que se ordenan por magnitud y se les asignan rangos con sus respectivos signos. La suma de los rangos positivos y negativos permite evaluar si hay una diferencia significativa entre las medianas de las muestras. Los resultados del test de Wilcoxon confirman que las diferencias en rendimiento entre ILS-SA y algunos algoritmos alternativos son estadísticamente significativas. En particular, ILS-SA1, ILS-SA2, ILS-SA3, ILS-SA4, ILS-SA6 e ILS-SA7 presentan valores p inferiores a 0,05, lo que indica que su comportamiento difiere significativamente de ILS-SA con un nivel de confianza de 95 %. Por el contrario, ILS-SA5 muestra valores p más altos, lo que sugiere que no existen diferencias estadísticamente significativas con respecto a ILS-SA, reforzando la idea de que estas variantes son comparables en términos de calidad de soluciones.

El comportamiento descrito anteriormente se puede observar en la Figura 4, que muestra la convergencia del algoritmo propuesto y sus siete variantes para la instancia 135, utilizando una semilla igual a 0. La instancia número 135, corresponde a `instancia_40_7_1_2.json`. Esta instancia (G, F, a, τ, f, T) con número de bomberos $f = 1$, la duración de los intervalos de tiempo T también es uno, y el número de incendios iniciales $F = 1$. El grafo $G = (V, E)$, con $|V| = 40 \in \{10, 20, 30, 40\}$, pertenecientes al grafo de instancia 7 y un factor de escalamiento de la distancia euclidiana igual a 2. Cada curva en la figura representa uno de los algoritmos, donde el eje y indica el valor del costo y el eje x corresponde al tiempo transcurrido, representado en escala logarítmica para facilitar su visualización.

Se observa que ILS-SA obtuvo el mejor costo correspondiente al BKS. Por otro lado, ILS-SA4 fue la versión que completó la ejecución en menor tiempo, aunque con una solución considerablemente más alejada del óptimo. De manera similar, ILS-SA6 alcanzó una de las peores soluciones y requirió incluso más tiempo que ILS-SA.

Asimismo, ILS-SA1, ILS-SA2 e ILS-SA3 presentan trayectorias de convergencia muy similares entre sí, tal como se observa en la Tabla 5. En cuanto a ILS-SA5, esta variante logró soluciones competitivas en menor tiempo y estuvo bastante cerca del desempeño de ILS-SA, aunque sin superarlo.

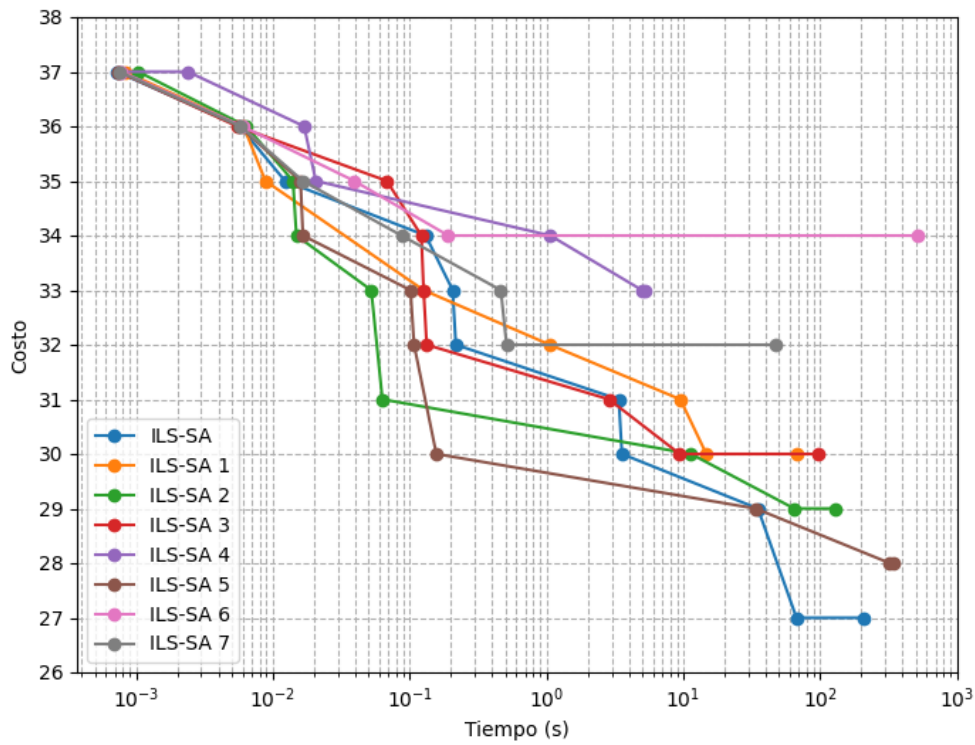


Figura 4: Gráficos de convergencia de algoritmos propuestos para la instancia 135.

Capítulo 6

Conclusión

La presente tesis aborda el MFP, una variante del FP que considera tiempos de desplazamiento variables entre nodos en función de la distancia euclidiana. Dada la alta complejidad computacional del modelo de programación cuadrática propuesto en la literatura, presentando tiempos de resolución poco prácticos. Esta investigación propuso tres métodos de resolución, un modelo de MILP, un modelo de CP y una metaheurística que combina iterated local search y simulated annealing (ILS-SA), con el objetivo de mejorar la eficiencia en el tiempo de cómputo sin sacrificar la calidad de las soluciones.

Los resultados obtenidos a partir de la ejecución de los modelos y el algoritmo propuestos revelan importantes hallazgos en términos de calidad de la solución y tiempo de cómputo. Los modelos del MIQCP, MILP y CP mostraron comportamientos diferenciados según el tamaño de las instancias. El modelo de CP destaca por su capacidad para encontrar soluciones óptimas en instancias más grandes, con un GAP^{UB} de 0,00 % en varios casos. El análisis de los tiempos de cómputo también reflejó variaciones significativas, con el modelo de MIQCP mostrando los tiempos más altos, mientras que los modelos MILP y CP fueron más eficientes en términos de tiempo computacional.

A partir del análisis de componentes, se observó que el algoritmo ILS-SA depende principalmente de la búsqueda local LS-2. Su desempeño también está influenciado por la reducción de temperatura, lo que resalta la importancia de esta variable en el proceso de búsqueda local dentro de la metaheurística.

El análisis comparativo demuestra que, si bien el modelo CP ofrece una alta precisión

en la calidad de las soluciones, el enfoque ILS-SA destaca por su capacidad para alcanzar de manera consistente las mejores soluciones conocidas y por su eficiencia computacional en instancias de mayor tamaño. Así, ILS-SA se posiciona como una alternativa robusta y escalable para abordar eficientemente el MFP, frente a las limitaciones de los modelos bajo restricciones de cortos tiempos de cómputo.

Respecto a líneas de trabajo futuro, se propone explorar una metaheurística que integre enfoques de CP o MILP, junto con la metaheurística ILS-SA. Se sugiere investigar nuevas metaheurísticas (como Variable Neighborhood Search, Ant Colony Optimization, Algoritmos Genéticos, etc.) con el fin de mejorar aún más la eficiencia y calidad de las soluciones en instancias de gran escala. Asimismo, sería relevante extender el MFP a escenarios con múltiples bomberos o con mayores condiciones que asemejen a la realidad, lo que permitiría modelar situaciones más realistas y complejas. Otra dirección interesante consiste en estudiar estrategias de contención en grillas infinitas, lo cual contribuiría al análisis teórico del problema.

Bibliografía

- Adjiashvili, D., Baggio, A., & Zenklusen, R. (2017). Firefighting on trees beyond integrality gaps. In *Proceedings of the 2017 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2364–2383.
- Anshelevich, E., Chakrabarty, D., Hate, A., & Swamy, C. (2009). Approximation Algorithms for the Firefighter Problem: Cuts over Time and Submodularity. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Dong, Y., Du, D.-Z., & Ibarra, O., editors, *Algorithms and Computation*, volume 5878, pages 974–983. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Bazgan, C., Chopin, M., Cygan, M., Fellows, M. R., Fomin, F. V., & Van Leeuwen, E. J. (2014). Parameterized complexity of firefighting. *Journal of Computer and System Sciences*, 80(7):1285–1297.
- Bazgan, C., Chopin, M., & Ries, B. (2013). The firefighter problem with more than one firefighter on trees. *Discrete Applied Mathematics*, 161(7-8):899–908.
- Bonato, A., Janssen, J., & Roshanbin, E. (2014). Burning a Graph as a Model of Social Contagion. In Bonato, A., Graham, F. C., & Prałat, P., editors, *Algorithms and Models for the Web Graph*, volume 8882, pages 13–22. Springer International Publishing, Cham.
- Cai, L., Verbin, E., & Yang, L. (2008). Firefighting on Trees: $(1 - 1/e)$ -Approximation, Fixed Parameter Tractability and a Subexponential Algorithm. In Hong, S.-H., Nagamochi, H., & Fukunaga, T., editors, *Algorithms and Computation*, volume 5369, pages 258–269. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Chen, X., Hu, X., Wang, C., & Zhang, Y. (2017). Continuous Firefighting on Infinite Square Grids. In Gopal, T., Jäger, G., & Steila, S., editors, *Theory and Applications of Models of Computation*, pages 158–171, Cham. Springer International Publishing.
- CONAF (2024). Conaf entregó balance de incendios temporada 2023–2024. Accedido: 2025-05-08.
- Coupechoux, P., Demange, M., Ellison, D., & Jouve, B. (2019). Firefighting on trees. *Theoretical Computer Science*, 794:69–84.
- Develin, M. & Hartke, S. G. (2007). Fire containment in grids of dimension three and higher. *Discrete Applied Mathematics*, 155(17):2257–2268.

- Farhadi, M., Toriello, A., & Tetali, P. (2021). The traveling firefighter problem. In *SIAM Conference on Applied and Computational Discrete Algorithms (ACDA21)*, pages 205–216. SIAM.
- García-Díaz, J., Perez-Sansalvador, J. C., Rodríguez-Henríquez, L. M. X., & Cornejo-Acosta, J. A. (2022a). Burning Graphs Through Farthest-First Traversal. *IEEE Access*, 10:30395–30404.
- García-Díaz, J., Rodríguez-Henríquez, L. M. X., Pérez-Sansalvador, J. C., & Pomares-Hernández, S. E. (2022b). Graph Burning: Mathematical Formulations and Optimal Solutions. *Mathematics*, 10(15):2777.
- Gutiérrez-De-La-Paz, B. R., García-Díaz, J., Menchaca-Méndez, R., Montenegro-Meza, M. A., Menchaca-Méndez, R., & Gutiérrez-De-La-Paz, O. A. (2022). The Moving Firefighter Problem. *Mathematics*, 11(1):179.
- Hartnell, B. & Li, Q. (2000). Firefighting on trees: How bad is the greedy algorithm? *Congressus Numerantium*, 145.
- Hartnell, B. & Rall, D. F. (1995). A characterization of graphs in which some minimum dominating set covers all the edges. *Czechoslovak Mathematical Journal*, 45(2):221–230.
- INFOR (2023). Información regional - región del biobío. Instituto Forestal de Chile, Disponible en: <https://ifn.infor.cl/index.php/informacion-regional/region-del-bio-bio>. Accedido: 2025-04-16.
- Iwaikawa, Y., Kamiyama, N., & Matsui, T. (2011). Improved Approximation Algorithms for Firefighter Problem on Trees. *IEICE Transactions on Information and Systems*, E94-D(2):196–199.
- Kempe, D., Kleinberg, J., & Tardos, E. (2005). Influential Nodes in a Diffusion Model for Social Networks. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Caires, L., Italiano, G. F., Monteiro, L., Palamidessi, C., & Yung, M., editors, *Automata, Languages and Programming*, volume 3580, pages 1127–1138. Springer Berlin Heidelberg, Berlin, Heidelberg.
- Khakzad, N. (2018). A Graph Theoretic Approach to Optimal Firefighting in Oil Terminals. *Energies*, 11(11):3101.
- King, A. & MacGillivray, G. (2010). The firefighter problem for cubic graphs. *Discrete Mathematics*, 310(3):614–621.
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220(4598):671–680.
- Klein, R., Levkopoulos, C., & Lingas, A. (2018). Approximation Algorithms for the Geometric Firefighter and Budget Fence Problems. *Algorithms*, 11(4):45.
- Laporte, G. & Toth, P. (2022). A gap in scientific reporting. *4OR*, 20(1):169–171.

- Lourenço, H. R., Martin, O. C., & Stützle, T. (2003). *Iterated Local Search*, pages 320–353. Springer US, Boston, MA.
- López-Ibáñez, M., Dubois-Lacoste, J., Pérez Cáceres, L., Birattari, M., & Stützle, T. (2016). The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, 3:43–58.
- MacGillivray, S. F. . G. (2009). The firefighter problem: a survey of results, directions and questions. *Australasian Journal of Combinatorics*, 43:57–78.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. 21(6):1087–1092.
- Michalak, K. & Knowles, J. D. (2016). Simheuristics for the Multiobjective Nondeterministic Firefighter Problem in a Time-Constrained Setting. In Squillero, G. & Burelli, P., editors, *Applications of Evolutionary Computation*, volume 9598, pages 248–265. Springer International Publishing, Cham.
- Ng, K. & Raff, P. (2008). A generalization of the firefighter problem on $z \times z$. *Discrete Applied Mathematics*, 156(5):730–745.
- Rossi, F., van Beek, P., & Walsh, T., editors (2006). *Handbook of Constraint Programming*. Elsevier, Amsterdam.
- Wang, K., Yuan, Y., Chen, M., Lou, Z., Zhu, Z., & Li, R. (2022). A Study of Fire Drone Extinguishing System in High-Rise Buildings. *Fire*, 5(3):75.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Zambon, M. J. O., De Rezende, P. J., & De Souza, C. C. (2019). Solving the geometric firefighter routing problem via integer programming. *European Journal of Operational Research*, 274(3):1090–1101.

Anexo

Ejemplo de aplicación

Para facilitar la visualización del funcionamiento del modelo ILS-SA, se ha seleccionado la instancia número 47, correspondiente a `instancia_20_3_1_2.json`. Esta instancia cuenta con un total de 20 nodos, pertenecientes al grafo de instancia 3, con un único nodo inicialmente quemado y un factor de escalamiento de la distancia euclidiana igual a 2.

En la Figura 5 y Figura 6 se muestra una representación gráfica de la evolución del fuego en el caso en que el bombero no intervenga, es decir, cuando la propagación ocurre libremente desde el nodo inicialmente quemado (nodo 16) hacia sus vecinos en intervalos de tiempo discretos. Como se observa, la propagación del fuego culmina en seis rondas, consumiendo la totalidad de los 20 nodos del grafo.

La aplicación del algoritmo produjo como mejor solución la siguiente secuencia de movimientos para el bombero: [5, 19, 0, 9, 2, 3, 11, 14, 15, 1, 18, 12, 16, 13, 4, 10, 8, 7, 6, 17]. Como se aprecia en la Figura 7, el bombero parte desde el depósito y se dirige primero al nodo 5, al que llega en 1,32 unidades de tiempo, antes de que el fuego lo alcance. En este punto, en el tiempo 1, los nodos 0, 2 y 17 ya han sido consumidos por las llamas. Posteriormente, el bombero se traslada al nodo 19, arribando en el tiempo 2,88, nuevamente antes de que el fuego lo alcance, mientras que en el tiempo 2 los nodos 4 y 9 han sido quemados.

Gracias a esta estrategia, la propagación del fuego se detiene tras la quema del nodo 1 en el tiempo 3, evitando que se extienda a los nodos 3, 6, 7, 8, 10, 11, 12, 13, 14, 15 y 18. Como resultado, la minimización del daño ha permitido reducir la cantidad de nodos quemados a solo 7.

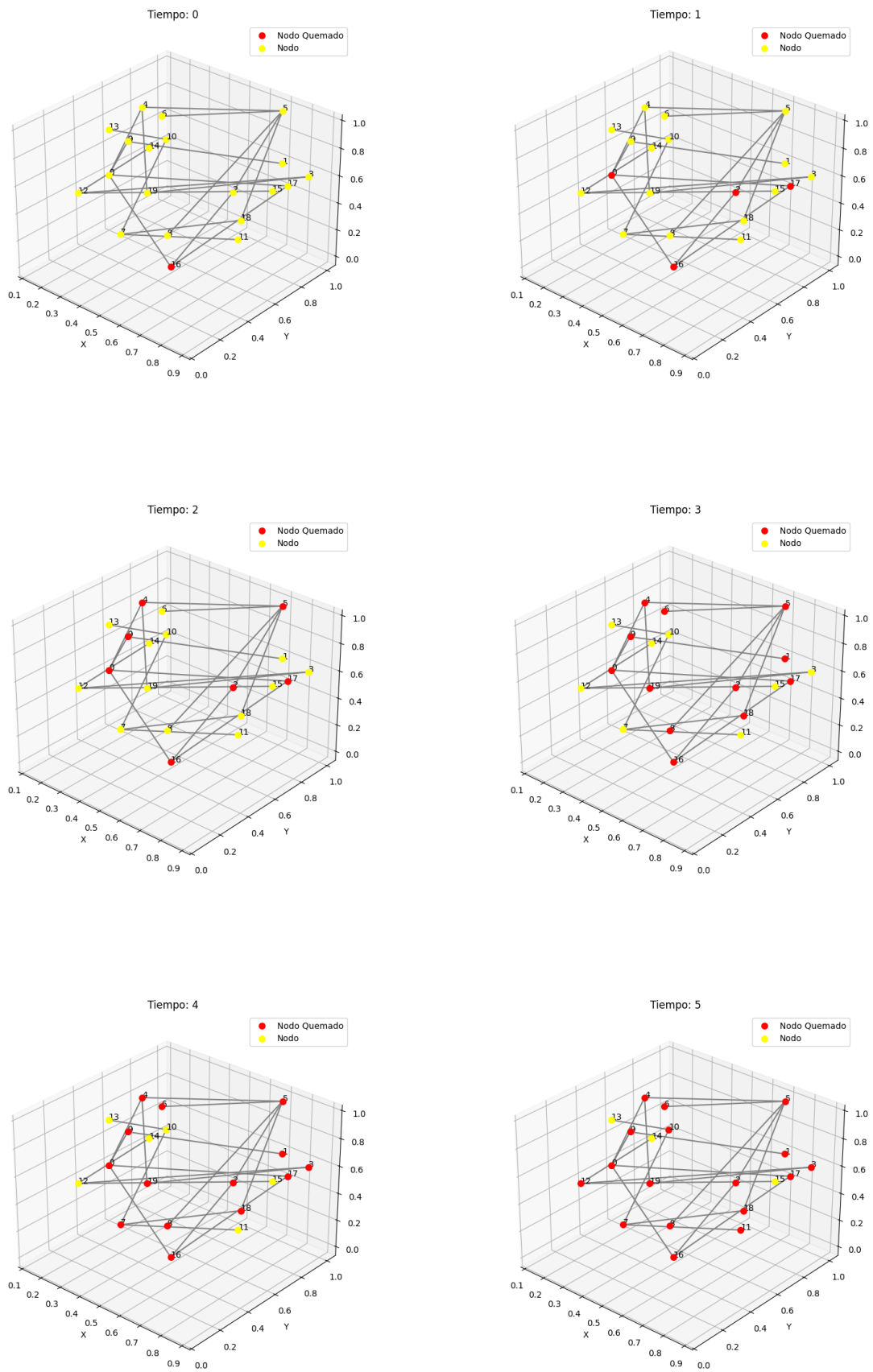


Figura 5: Propagación del fuego caso en que bombero no interviene parte a.

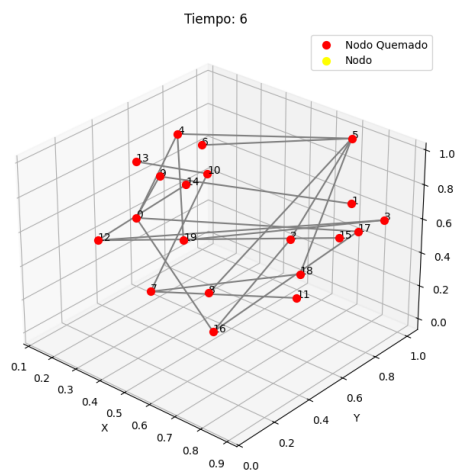


Figura 6: Propagación del fuego caso en que bombero no interviene parte b.

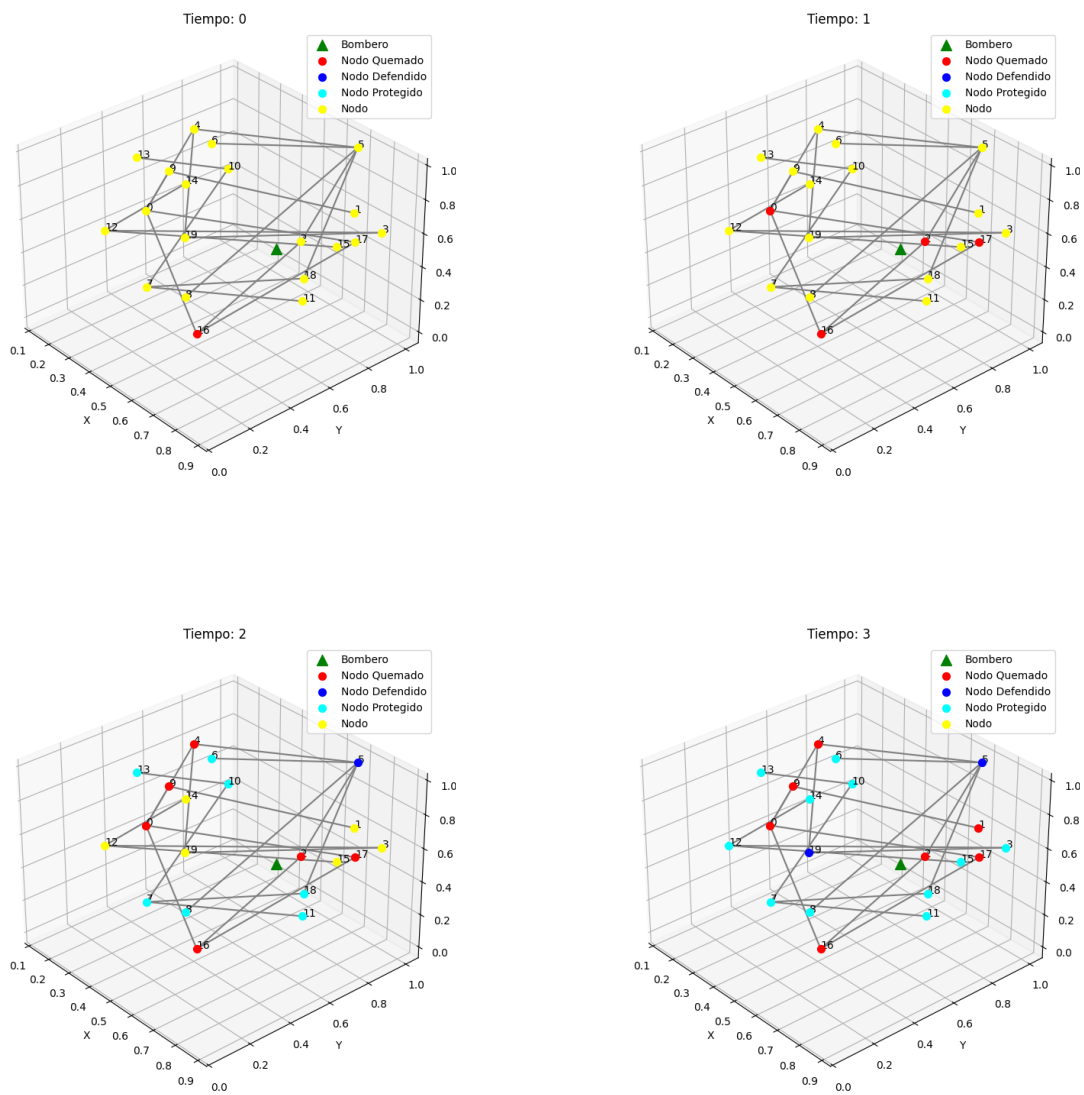


Figura 7: Propagación del fuego con solución presentada.