Trabajo Fin de Máster

Selección y asignación de recursos para la contención de un incendio forestal

Autor: Jorge Rodríguez Veiga
Directora: Balbina Casas Méndez

Julio 2014

Máster Oficial en Técnicas Estadísticas
Universidad de Santiago de Compostela

Trabajo Fin de Máster

Selección y asignación de recursos para la contención de un incendio forestal

Autor: Jorge Rodríguez Veiga
Directora: Balbina Casas Méndez

El presente documento recoge el Trabajo de Fin de Máster para el Máster en Técnicas Estadísticas realizado por D. Jorge Rodríguez Veiga con el título "Selección y asignación de recursos para la contención de un incendio forestal".

Ha sido realizado bajo la dirección de Dña. Balbina Casas Méndez que lo considera terminado y da su conformidad para la presentación y defensa del mismo.

Santiago de Compostela, a 1 de julio de 2014

Fdo. Balbina Casas Méndez

Índice general

Resumen

Determinar una planificación óptima, que incluya el número y tipo, de recursos necesarios para la extinción de un incendio forestal no es una tarea sencilla, pues se cuenta con gran número de posibilidades. La dificultad se incrementa si la tarea no consiste únicamente en seleccionar el conjunto de recursos, sino que también estos se han de asignar a los diferentes periodos de tiempo en el transcurso del incendio.

En el trabajo se exponen dos primeros modelos de programación lineal entera, uno determinista y otro estocástico, para dar solución al problema de la selección de recursos. A partir de estos modelos iniciales, se realiza una generalización determinista con el fin de abordar un conjunto de nuevas restricciones sobre los tiempos totales de empleo de los recursos, sugeridas por la empresa con la que hemos colaborado durante la realización del trabajo, con amplia experiencia, así como la asignación horaria de los recursos. Este último modelo no se abordará por ahora con detalle desde el punto de vista estocástico, aunque podemos anticipar su complejidad y consecuente coste temporal de ejecución.

Finalmente, se incluyen los algoritmos implementados en el software libre R. Con ellos, analizamos los resultados obtenidos en diferentes escenarios para comprobar el adecuado funcionamiento de los modelos.

Capítulo 1

Introducción

El fenómeno de los incendios forestales se ha convertido en uno de los mayores problemas ecológicos que sufren nuestros montes, debido a la elevada frecuencia e intensidad que ha adquirido en las últimas décadas.

El fuego es un elemento natural que forma parte de los fenómenos que modelan el paisaje. Especialmente en los ambientes mediterráneos, debemos admitir que el fuego es un incómodo compañero de viaje con el que hay que convivir. Precisamente, gran parte de nuestra vegetación está adaptada a la acción del fuego, con estrategias rebrotadoras o de germinación tras el incendio.

Sin embargo, el fenómeno de los incendios forestales ha dejado de ser una perturbación natural que modela el paisaje para convertirse en una terrible amenaza que en más de un 95 % de los casos está ocasionada por el ser humano. Es un problema ambiental de primer orden en nuestro país, por lo que cualquier tipo de herramienta para poder prevenirlos o combatirlos es de gran importancia.

Es imprescindible para el control de estos incendios catastróficos tomar decisiones eficientes debido a que el presupuesto y los recursos contra incendios son limitados. Una posible forma de abordar este problema es invertir en el esfuerzo de prevención. El esfuerzo en prevención se puede abordar desde la educación, la campaña o la contratación de patrullas. Algunos incendios forestales de origen humano pueden ser prevenidos a través de este esfuerzo. Sin embargo, los incendios forestales naturales no se pueden prevenir. Por lo tanto, es necesario contener los incendios, mientras son pequeños para reducir al mínimo los costes y los daños asociados. El fracaso para contener un fuego pequeño puede resultar en un gran incendio destructivo. Por tanto, es importante utilizar todos los medios de extinción contra incendios de forma eficiente, empleando diferentes tipos de estrategias y tácticas en la gestión de los incendios forestales.

En este sentido, la teoría económica juega un papel central en la gestión de incendios forestales. Los precursores en el estudio económico de los incendios forestales fueron Headley [10] y Sparhawk [19], en los años 20, quienes describen cómo establecer un programa de gestión de incendios forestales óptimo. El marco teórico utilizado para identificar la forma más eficiente de gestionar los costes de un incendio forestal ha sido el denominado *Cost plus Net Value Change* (C + NVC) (Gorte y Gorte [8]). De este modo, se pretende minimizar el coste por la utilización de los recursos en la lucha contra un incendio, más un coste producido por las hectáreas quemadas de terreno, donde no sólo se debe tener en cuenta las pérdidas materiales en el incendio (árboles, zona urbana, etc.), sino también la repoblación o reconstrucción de estas zonas.

En el ámbito nacional y muy recientemente, año 2010, surge con motivo de mejorar la eficiencia en la lucha contra incendios, el proyecto PROMETEO, uno de los mayores proyectos de investigación aplicada concedido a un consorcio empresarial en nuestro país en materia de lucha contra incendios forestales. El proyecto, cuya empresa coordinadora es INAER, cuenta con más de 16 empresas y está subvencionado en casi un 44 % por el Centro para el Desarrollo Tecnológico Industrial (CDTI), mediante el denominado Programa CENIT.

Además, PROMETEO cuenta con la participación activa de las administraciones públicas autonómicas y el Ministerio de Medio Ambiente y Medio Rural y Marino, quienes participan en la concepción de las necesidades del sector, en pro de alcanzar los siguientes objetivos generales.

- Optimizar los recursos de que disponen las administraciones públicas para minimizar el riesgo ante incendios forestales y mitigar los daños medioambientales en caso de incendio.
- Reducir el número y la magnitud de los grandes incendios forestales, minimizando la seguridad de los dispositivos de extinción.

Como continuación del proyecto PROMETEO, surge en el año 2013, adscrito al programa INNTERCONECTA GALICIA, el proyecto LUMES. El objetivo principal del proyecto es el desarrollo de nuevas tecnologías avanzadas para la lucha integral contra los grandes incendios forestales, permitiendo reducir el número de estos y su superficie, generando una envolvente de seguridad en las operaciones, que reduzca notablemente la siniestralidad de los intervinientes (técnicos, brigadistas y pilotos).

Para ello, se pretende alcanzar los siguientes objetivos específicos.

- Desarrollar tecnologías para la extinción nocturna con medios aéreos, que permitan incrementar un 100 % el tiempo de presencia de los medios aéreos (de 12 a 24 horas de servicio), permitiendo la lucha integral contra incendios forestales cuando son más vulnerables, al descender las temperaturas y amainar los vientos por la noche.
- Desarrollar un sistema eficiente y seguro de coordinación del tráfico aéreo en tiempo real, que permita la gestión de los medios aéreos en cuanto a su distribución dentro del escenario de incendios y sus instrucciones operativas.
- Desarrollar sistemas y herramientas helitransportadas para la localización, comunicación y rescate de las brigadas y otros medios de extinción en situaciones de escasa visibilidad y alto riesgo.

- Desarrollar un sistema VANT (Vehículo aéreo no tripulado) certificable, basado en el desarrollo e integración de una carga de pago específica para realizar misiones de observación, telecomunicaciones y de sondeos meteorológicos en operaciones de extinción de incendios.
- Desarrollar un sistema experto, que funcionando en tiempo real, integre todas las variables que influyen en la extinción de incendios y defina el procedimiento de actuación óptimo, asistiendo al director de extinción en la toma de decisiones.

Con el proyecto LUMES, se contribuirá fuertemente a desarrollar tecnologías para optimizar los esfuerzos dedicados a la extinción de incendios y que permitan reducir la superficie afectada, así como aumentar la seguridad de las brigadas, mejorar la coordinación en las operaciones de extinción, ampliar la franja horaria de actuación e introducir nuevos recursos aéreos en la lucha contra grandes incendios forestales.

Uno de nuestro objetivos en el marco de este proyecto, es desarrollar un modelo con el fin de determinar la selección y asignación óptima de los recursos, en diferentes periodos de tiempo, para la contención de un único incendio. Los modelos propuestos en este trabajo muestran la viabilidad del uso de técnicas de optimización para la planificación y presupuestación de incendios forestales. Posteriormente, se requerirá adaptar/ajustar los problemas de programación lineal entera a una situación real y comparar los resultados obtenidos con los datos históricos aportados, para analizar y discutir los resultados que nos devuelven los modelos.

En el trabajo se expondrán tres modelos diferentes. En una primera etapa, y como introducción al planteamiento del problema, se estudia y trabaja con un modelo determinista, inspirado en gran medida en el trabajo realizado por Donovan y Rideout [6]. Tras la comprensión del primer modelo, se estudió la posibilidad de mejorarlo, proporcionándole un mayor alcance. Examinando la literatura, se encontró el trabajo de tesis de Lee [13], en el cual se desarrolla el modelo determinista expuesto por Donovan y Rideout, incorporando estocasticidad sobre los parámetros referentes al incendio. Estos dos modelos, tienen como objetivo determinar el número de recursos necesarios para contener un incendio, desde un instante inicial (cuando aún no se ha realizado el despliegue de ningún tipo de recurso), minimizando los costes y daños relacionados con el fuego. Al identificar la combinación óptima de los recursos de extinción de incendios, los modelos aplican la teoría C+NVC, mencionada anteriormente, al incendio.

El último modelo que se expone en el trabajo, se realiza tras estudiar las necesidades demandadas por la empresa INAER. En éste, no sólo se debe obtener el número necesario de aeronaves para contener el incendio, sino que se han de asignar con precisión a los distintos periodos de tiempo. Además, a petición de la empresa demandante, se han de tener en consideración ciertas restricciones sobre el tiempo de empleo de los recursos. Este último modelo no se abordará por ahora con detalle desde el punto de vista estocástico, aunque podemos anticipar su complejidad y coste temporal de ejecución.

El análisis de sensibilidad se lleva a cabo mediante la alteración de los supuestos sobre el comportamiento del fuego y de las características de los recursos, para demostrar la capacidad y respuesta de los modelos de programación entera ante diferentes situaciones. Además, se hace una ampliación de éste para resolver la combinación óptima de los recursos enfrentándonos a diferentes tipos de restricciones presupuestarias (en los dos primeros modelos), o de tiempo (último modelo). Este tipo de optimización con restricciones ilustra la capacidad del modelo para dar cabida a ciertas limitaciones, en caso de ser necesario.

Capítulo 2

Revisión bibliográfica

Los incendios forestales se han vuelto cada vez más destructivos y amenazan seriamente nuestros ecosistemas. Por tanto, es imperativo hacer grandes esfuerzos para reducir los daños ocasionados por los incendios forestales mediante la creación de planes de manejo eficaces. El ámbito de la gestión de los incendios forestales es amplia. Incluye la contención de incendios forestales y la represión, la planificación a largo plazo para la ubicación de las bases aéreas y asignación de recursos, la planificación y programación de los recursos necesarios con respecto a la ocurrencia de incendios forestales reales a corto plazo, etc.

Los métodos de programación y simulación matemática son ampliamente utilizados en la toma de estas decisiones estratégicas y tácticas. Este capítulo se compone de tres secciones. La primera, proporciona referencias sobre conocimientos previos relacionados con los sistemas de gestión de incendios forestales. La segunda, proporciona información concerniente a modelos de decisiones tácticas de manejo de incendios forestales, y, finalmente, la última sección presenta una breve revisión de trabajos de programación estocástica, que se emplearán para resolver el modelo estocástico propuesto.

2.1. Información general sobre la gestión de incendios forestales

En cuanto a la gestión de incendios forestales, analizar la economía de los mismos o la estimación de su probabilidad de ocurrencia, son temas importantes, que dan pie a la investigación en problemas relacionados. También es interesante conocer cómo los sistemas de gestión de incendios forestales han evolucionado y lo que se necesita para hacerlos más efectivos.

Los economistas han ampliado los métodos para evaluar y cuantificar el valor económico total de los bosques. Algunos métodos de evaluación, han sido propuestos por Morton [16]. Los ecosistemas forestales pueden considerarse como capital natural que puede producir una amplia gama de bienes y servicios para la humanidad. En general, la madera, se puede cuantificar fácilmente en términos de precio. Sin embargo, hay muchos otros aspectos que se deben cuantificar en las áreas silvestres, como el almacenamiento de carbono, los minerales, la productividad del suelo, uso recreativo, etc. Algunos de estos aspectos no se cuantifican fácilmente como valor económico, sin embargo, cuando se trata de los incendios forestales, la evaluación del valor de las tierras silvestres que hay que proteger es extremadamente importante, ya que la solución para muchos tipos de supresión depende del valor económico de la zona. Para los dos primeros modelos propuestos en el trabajo, este aspecto cobra una importancia crucial. Por otro lado en el último modelo propuesto, si bien es cierto que se tiene en cuenta el valor económico del suelo donde se produce el incendio, este sólo se emplea para ponderar lo importante que es o no una contención más o menos rápida.

Dado que los daños causados por los incendios forestales se componen de factores tangibles e intangibles, es muy difícil cuantificar el valor económico de los daños y perjuicios. En el trabajo de Butry et al. [3], se cuantifica el valor económico de los daños causados por los incendios forestales en Florida. En éste, se definen siete categorías principales de daños: los costes de pre-extinción, los costes de extinción, los gastos de socorro, las pérdidas de madera, los daños a la propiedad, las pérdidas relacionadas con el turismo y los efectos sobre la salud humana. Esta investigación cuantifica los impactos económicos de los incendios forestales de manera sistemática y empírica.

En el trabajo de Gorte y Gorte [8] ya mencionado, se propone el marco teórico de la economía de los incendios forestales, C+NVC. Este modelo se ha utilizado como modelo principal de la evaluación de la economía de incendios forestales. El modelo, pretende minimizar el coste de los incendios forestales, reduciendo al mínimo la suma de los costes de pre-supresión, de supresión, y el coste por hectáreas de terreno quemadas (NVC). El coste de pre-supresión es el coste fijo que se gasta antes de que la temporada de incendios empiece, a través de la educación, la patrulla, la campaña o la inversión en los recursos o nuevas instalaciones. El coste de supresión son los gastos que se producen durante la temporada de incendios. La mayor parte del coste está asociado con la supresión del fuego y coste de operación de contención. Por último, el NVC es el coste que se incurre por el daño de los incendios forestales durante la temporada de incendios. El C+NVC se ha convertido en la teoría económica más empleada en el contexto de la gestión de incendios forestales.

Antes de examinar la literatura detallada sobre la contención y supresión, vale la pena realizar una buena revisión literaria sobre la gestión de los incendios forestales. En Minas et al. [14], se recogen varios estudios de investigación operativa en la gestión de incendios forestales, teniendo en cuenta temas como la prevención y la gestión de combustible, la planificación estratégica y la táctica de la detección, etc.

2.2. Modelos de toma de decisiones en incendios forestales

Las decisiones tácticas en la gestión de los incendios forestales se han de realizar en un periodo de tiempo relativamente corto. Estas decisiones pueden incluir la combinación de un cierto tipo de recursos para contener un incendio en particular o la planificación de los tiempos de actuación de las aeronaves basándose en las demandas emergentes del incendio. El objetivo de estos enfoques es el de analizar los problemas que se pueden producir en la gestión de un incendio forestal y proporcionar decisiones óptimas con el fin de reducir al mínimo los daños o los riesgos.

Un modelo de decisión táctica que determina la combinación óptima de los recursos de extinción para reducir al mínimo la función de C+NVC es el propuesto por Donovan y Rideout [6]. Este modelo se ajusta muy bien cuando hay que responder sobre qué recursos emplear y cuándo se puede tener contenido el incendio para minimizar los costes. Sin embargo, este modelo asume que el perímetro y el área del incendio en cada periodo de tiempo tiene caracter determinista.

En este sentido, el trabajo de Lee [13] pretende solventar el inconveniente de suponer que el perímetro y el área del incendio en cada periodo de tiempo es un valor conocido. Para abordar este problema, Lee recurre al uso de la programación estocástica, obteniendo de este modo soluciones fuertemente robustas ante la incertidumbre de cómo va a evolucionar el incendio.

En la línea de la programación estocástica, nos encontramos con otros trabajos, como el de Hu y Ntaimo [11]. En este caso, la estocasticidad se vincula a la aparición simultánea de un número finito de incendios. Los autores presentan un modelo de programación lineal entera mixta estocástica, donde se resuelve de forma óptima la asignación de los recursos disponibles a cada uno de los incendios, minimizando los costes estimados que conlleva la operación completa. Además, en el artículo, se expone un modelo que simula la extinción de los incendios en base a los resultados obtenidos del modelo de optimización estocástica, y a su vez, ofrece información al modelo de optimización para la revisión de los planes de envío de recursos (en caso de ser necesario).

En el trabajo de Greulich [9], se presenta un procedimiento de cálculo, asociado a la modelización de dónde han de establecerse las bases de las aeronaves contra incendios dentro de una región protegida, para obtener una buena estrategia de actuación en el ataque inicial. Este modelo, proporciona al usuario gran flexibilidad al especificar la distribución espacial de la ocurrencia de incendios en la zona. Una vez que se ha especificado esta distribución, la distancia de vuelo previsto de la base a un punto aleatorio de fuego, se calcula fácilmente empleando una hoja de cálculo. En el trabajo, se describen las principales ventajas del método de cálculo presentado, así como ejemplos numéricos detallados y las posibles mejoras a desarrollar en el futuro.

Por último, en el trabajo de Finney, [7], se desarrolla un algoritmo que permite simular la propagación de incendios forestales. El trabajo se desarrolla desde un enfoque denominado celular, que proporciona una mejora en la simulación, al considerar únicamente aquellas células activas para el cálculo y transmisión de mensajes. Además, las células en la simulación se pueden eliminar dinámicamente según sea necesario. El modelo considera células estáticas, que almacenan la información geográfica, y células dinámicas que pueden recoger diferentes características estocásticas correspondientes a condiciones meteorológicas. Estas células son las responsables de la decisión acerca de la forma de la propagación y la intensidad de la línea de fuego, elementos que juegan un papel clave en la simulación.

2.3. Programación estocástica

En los modelos de toma de decisiones en incendios forestales, existe generalmente mucha aleatoriedad o incertidumbre en los parámetros asociados al comportamiento del fuego. Por lo tanto, la programación estocástica cobra gran sentido, pues una de sus características más significativas es el reflejo de dicha incertidumbre en el modelo.

Dado que la incertidumbre se tiene en el comportamiento del fuego, es importante tener en cuenta esa aleatoriedad en la modelización de problemas de incendios forestales. La programación entera mixta estocástica aúna dos clases de problemas que, por lo general, son de gran dificultad: los problemas estocásticos y los problemas enteros. En general, un programa estocástico evalúa el problema mediante la optimización de los diferentes escenarios posibles. El llamado modelo de dos etapas, es ampliamente utilizado en la solución de problemas estocásticos. En su configuración, podemos diferenciar en líneas generales dos etapas. En la primera etapa, se trabaja con el llamado problema máster, en el que se resuelve un modelo con información incompleta, pues se elimina la información de los eventos aleatorios. En la segunda, se consideran los llamados subproblemas. Tras discretizar la aleatoriedad en un número finito de escenarios, se toma la información de cada uno de ellos y junto con la información obtenida en la primera etapa, se toman acciones que a su vez condicionarán las decisiones sobre el problema máster. Este procedimiento se repite hasta que se hace converger la solución. La explicación se realizará de forma más detallada en el Capítulo 4.

Cuando el problema estocástico únicamente trabaja con variables de decisión continuas, el método L-shaped (cf. Slyke y Wets [18]) es el algoritmo más ampliamente utilizado. Este método, se basa en el método de descomposición de Benders [1]. Debido a la linealidad y convexidad asumidos en el método L-shaped, éste funciona de forma muy eficiente en problemas a gran escala. Sin embargo, no puede ser usado para resolver modelos con variables de decisión enteras. Una forma de superar esta dificultad, es la utilización de cortes (de modo similar a los métodos de planos de corte, empleados en la programación lineal entera determinista). Si todas las variables de primera etapa son binarias, y sólo hay variables continuas en la segunda etapa, el método en forma de L todavía se puede utilizar para solucionar este tipo de problemas. Sin embargo, si ambas etapas tienen variables enteras, especialmente binarias, hemos de recurrir al método L-shaped entero o algoritmo L^2 (cf. Birge y Louveaux [2]), cuyos detalles se presentan en el Capítulo 4.

Otros libros de interés sobre el método L-shaped son los de Caroe y Tind [5], Caroe y Schultz [4] y Laporte y Louveaux [12].

Capítulo 3

Modelo determinista

El modelo que vamos a considerar es una extensión del propuesto por Donovan y Rideout [6], en el que se supone que el comportamiento del incendio forestal se puede predecir.

El problema de determinar la selección de recursos que consigan contener el incendio (que se traduce en construir una línea alrededor del incendio) a mínimo coste (C+NVC) se plantea como un problema de programación lineal entera, pues los medios de lucha son unidades indivisibles.

El problema tiene similitud con el denominado problema de la mochila (*knapsack problem*, Winston [20]). Este último modela la siguiente situación: imaginemos que se hace una excursión a la que sólo podemos llevar una mochila que, lógicamente, tiene una capacidad limitada. Cada objeto que introducimos ocupa un volumen dentro de la misma y en contrapartida durante el viaje nos proporcionará un beneficio o utilidad (ejemplo: una cantimplora). El problema surge cuando debemos elegir qué objetos seleccionar para llevar en la mochila, de forma que nuestro beneficio sea máximo (tengamos todo lo necesario) sin exceder su capacidad. Para indicar la selección o no de un objeto, se crea una variable de decisión binaria que toma el valor 1 si el elemento está seleccionado y 0 en otro caso.

Al igual que en el problema de la mochila, nosotros optimizaremos una función objetivo mediante la selección de un conjunto de posibilidades que estarán sometidas a una serie de restricciones. El objetivo será minimizar la suma de los costes relacionados con el incendio, donde se tendrán en cuenta, costes fijos por la selección de los recursos, costes variables por el empleo de los mismos y costes asociados a las hectáreas (en general, superficie) de terreno afectadas. La selección se realizará a partir de un conjunto de posibles recursos de extinción, mientras que las restricciones forzarán a que efectivamente se consiga la contención del incendio.

Una diferencia importante a destacar del problema de la mochila con respecto al que nos ocupa, es que en éste tendremos una dimensión temporal. En el primer problema, trabajamos en un único instante de tiempo, mientras que en el segundo, una vez elegido un recurso, éste se podrá utilizar durante más de un periodo. La dimensión temporal condiciona la restricción de contención, pues si el fuego se contiene en un periodo dado, su perímetro final será menor que si el incendio está contenido en uno posterior.

3.1. El modelo

Comenzamos introduciendo la notación necesaria, para posteriormente formular el problema.

Conjuntos:

$\mathcal{I} = \{1, \ldots, n\}$:	Conjunto de índices de los recursos, indexado por i.
$\mathcal{J} = \{1, \ldots, m\}$:	Conjunto de índices de los periodos de tiempo, indexado por j.

Parámetros referentes a los recursos:

C_i :	Coste por hora del recurso i.
P_i :	Coste fijo por emplear el recurso i.
PR_i :	Rendimiento del recurso i (en km/h).
A_i :	Tiempo que tarda el recurso i en llegar al incendio.

Parámetros referentes al incendio:

PER_j :	Incremento del perímetro del incendio en el periodo j.
NVC_j :	Incremento del NVC (hectáreas quemadas por coste de la hectárea) en el periodo j .
SP_j :	Perímetro del incendio en el periodo j.
H_j :	Periodo de tiempo j.

Variables de decisión:

D_{ij} :	Variable binaria que toma el valor 1 si el recurso i se emplea hasta el periodo j y 0 en otro caso.
L_j :	Perímetro construido por los recursos hasta el periodo j (en km).
Y_j :	Variable binaria que toma el valor 1 cuando el incendio no está contenido en el instante j y 0 en otro caso.
N_j :	Variable binaria que se define como $N_j = Y_{j-1}$ (siendo en el periodo inicial $Y_0 = 1$).
Z_i :	Variable binaria que toma el valor 1 si el recurso i se emplea y 0 en otro caso.

La formulación matemática siguiente modela la función objetivo (suma de los costes) y las limitaciones que se imponen para identificar la asignación óptima en el problema de la selección de recursos para la contención de un incendio forestal.

$$min \quad \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_iH_jD_{ij} + \sum_{i\in\mathcal{I}} P_iZ_i + \sum_{j\in\mathcal{J}} NVC_jN_j \tag{1.1}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}}(H_j - A_i)PR_iD_{ij} \geq \sum_{j\in\mathcal{J}} PER_jN_j \tag{1.2}$$

$$\forall i \in \mathcal{I}, \ \sum_{j\in\mathcal{J}} D_{ij} \leq Z_i \tag{1.3}$$

$$\forall j \in \mathcal{J}, \quad SP_jN_j - L_j \leq M\ Y_j \tag{1.4}$$

$$\forall j \in \mathcal{J}, \ \sum_{i\in\mathcal{I}}(H_j - A_i)PR_iD_{ij} = L_j \tag{1.5}$$

$$\forall j \in \mathcal{J}, \quad N_{j+1} = Y_j \tag{1.6}$$

La función objetivo (1.1), minimiza la suma de los costes involucrados en la extinción del incendio forestal. El primer término indica el coste variable por utilizar los recursos seleccionados, el segundo, el coste fijo asociado a la utilización de cada uno de ellos y el último será el producido por las hectáreas de terreno quemado.

La desigualdad (1.2) nos indica que en algún periodo el perímetro construido por los recursos tiene que cubrir el perímetro del incendio. Además, impone la condición de que el lado izquierdo de la desigualdad sea no negativa.

Las desigualdad (1.3) impone que si un recurso i se usa en algún periodo j, Z_i tomará el valor 1, con lo que esta variable aparecerá en el objetivo (en otro caso sería 0 porque estamos minimizando). Además, si se emplea el recurso i, éste sólo podrá operar hasta un periodo determinado, pues la suma de los D_{ij} no podrá ser mayor que 1.

La restricción (1.4) establece si el fuego está contenido o no en el periodo j. Si el perímetro construido por los recursos es menor que el perímetro del incendio, entonces el fuego no está contenido, y por tanto Y_j valdrá 1. Si por el contrario el perímetro construido por los recursos es mayor que el del incendio, Y_j podrá valer 0 (la función objetivo, que busca minimizarse, prefiere que los valores de los Y_j sean 0, pues así también lo serán los N_{j+1}). Debemos mencionar que el valor de M lo fijamos como $M = \max\{SP_j : \ j \in \mathcal{J}\}$ pues este valor será una cota superior para $SP_j - L_j$.

Por último, las restricciones (1.5) y (1.6) se establecen por propia definición de L_j y N_j respectivamente.

Además, al modelo se pueden incorporar restricciones adicionales, por ejemplo de naturaleza económica. Si se dispone de un presupuesto limitado en la lucha contra un incendio, este tipo de restricciones serán de gran importancia. De este modo, si quisiésemos acotar los costes fijos o los costes variables referentes a los recursos, deberemos incorporar al modelo las siguientes restricciones:

$$\sum_{i\in\mathcal{I}} P_i Z_i \leq B_1 \tag{1.7}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_i H_j D_{i,j} \leq B_2 \tag{1.8}$$

dónde B_1 será la cota superior del gasto fijo en el empleo de los recursos, y B_2 será la asociada a los gastos variables.

3.2. Función `seleccion_recursos`

Para el cálculo de la selección óptima de recursos, en el ataque inicial a un incendio forestal, se implementó en `R` la función `seleccion_recursos` (A.1).

Para el manejo correcto de la función implementada, realizaremos una descripción detallada de los argumentos de entrada y de los valores de salida que ésta proporciona. Cabe mencionar, que los argumentos de entrada de la función no coinciden con los parámetros del modelo, pues la función realiza cálculos internos que proporcionan, a partir de los argumentos de entrada, todos los parámetros necesarios para resolver el problema de programación lineal.

```
> seleccion_recursos(incendio,recursos,coste_area,B1=F,B2=F,imp=T)
```

<u>Argumentos de entrada:</u>

El primer argumento de entrada, `incendio`, es un `data.frame` que contiene la información estimada de la evolución del incendio forestal. Los elementos de `incendio` contienen la siguiente información:

`incendio$Hj`: vector formado por los distintos periodos de tiempo en los que se realiza la predicción del incendio. De este modo, el instante inicial será considerado el momento 0, y los sucesivos se corresponderán con cada uno de los periodos en los que se realiza la predicción de los parámetros (medidos en horas).

`incendio$SPj`: vector de la misma longitud que `Hj`, con el perímetro total del incendio en cada uno de los periodos (medido en km).

`incendio$area`: vector de la misma longitud que `Hj`, con el área total del incendio en cada uno de los periodos (medido en hectáreas).

El segundo argumento de entrada, `recursos`, también es un `data.frame` que contiene la información de los recursos (aeronaves) disponibles. Los elementos de `recursos` contienen la siguiente información:

`recursos$i`: vector que contiene el índice de cada tipo de recurso.

`recursos$DESi`: vector de la misma longitud que `i`, con una breve descripción o siglas de cada tipo de recurso.

`recursos$NUMi`: vector de la misma longitud que `i`, con el número de recursos de cada tipo.

`recursos$Ai`: vector de la misma longitud que `i`, con el tiempo que tardan en llegar los recursos al incendio (en horas).

`recursos$Ci`: vector de la misma longitud que `i`, con el coste por hora (variable) que supone emplear cada tipo de recurso (medido en €/hora).

`recursos$Pi`: vector de la misma longitud que `i`, con el coste (fijo) por emplear cada tipo de recurso (medido en €).

`recursos$PRi`: vector de la misma longitud que `i`, con el rendimiento que tiene cada tipo de recurso (en km/hora).

Otros argumentos de entrada son:

`coste_area`: valor numérico que establece el coste que supone la quema de una hectárea de terreno (en €/hectárea).

`B1`: Opcional. Valor numérico que establece una cota superior para el coste fijo por la utilización de recursos (en €). Por defecto, con `B1=F`, la función no fija ninguna restricción en este tipo de coste.

`B2`: Opcional. Valor numérico que establece una cota superior para el coste variable por la utilización de recursos (en €). Por defecto, con `B2=F`, la función no fija ninguna restricción en este tipo de coste.

`imp`: Opcional. Parámetro lógico que define el tipo de salida que queremos que nos devuelva la función. Si `imp=T`, la función hará una impresión de pantalla con una breve explicación del resultado óptimo. En caso contrario, si `imp=F`, se obtendrán las variables del modelo matemático.

Valores de salida:

Diferenciaremos dos tipos de salida, una con `imp=T` y otra con `imp=F`. Si `imp=T`, la salida es en forma de texto y en ella se detalla: si el algoritmo encontró una solución óptima, el tiempo de resolución del problema, la selección de recursos adecuados para la extinción (es decir, la solución óptima, que abarca los tipos de recursos seleccionados, la cantidad de cada uno de ellos y hasta que periodo han de operar), el perímetro cubierto por los recursos y el periodo en el que se contiene el incendio. Además, se muestra el coste total que supone la operación de contención y se indican los costes por utilización de los recursos y los gastos por hectáreas quemadas.

En caso contrario, si `imp=F`, la función devuelve una lista, `result`, con la siguiente información:

`result$Dij`: una matriz entera con número de filas igual a la longitud de `i` y número de columnas igual a la longitud de `Hj`. En ella se detalla hasta qué periodo ha de actuar cada uno de los recursos, para realizar una extinción con coste óptimo.

`result$Yj`: vector binario, de la misma longitud que `Hj`. Si el valor j-ésimo es 1, indica que el incendio no está contenido en el periodo j.

`result$Zi`: vector entero, de misma la longitud que `i`. En él se detalla el número de unidades de cada tipo, que se deben emplear para extinguir el incendio de forma óptima.

3.3. Asignación de recursos en un escenario simulado

Simularemos un escenario donde se consideran 6 periodos de tiempo, dados por intervalos de 1 hora. Los parámetros asociados al comportamiento del incendio se recogen en la Tabla 3.1 (perímetro y área totales del incendio en cada uno de los periodos). Estos datos son los empleados en el artículo de Donovan y Rideout [6] que fueron generados utilizando el software FARSITE (Finney [7]), un programa de simulación de incendios.

Para esta simulación, suponemos que el coste por hectárea quemada de terreno es de 100 €. Además dispondremos de siete tipos de recursos, cuyas características vienen recogidas en la Tabla 3.2. En esta tabla se añade el parámetro "Número", recogido en la tabla de datos de `R` como `NUMi`, que será el número de recursos del tipo i. Para tener en consideración este parámetro, la función considerará cada uno de estos recursos como uno nuevo, es decir, si hay dos unidades disponibles de cada recurso ($NUM_i = 2$ para cada $i = 1, \ldots, n$), la función considerará que hay 2 recursos de cada tipo, haciendo un total de $2n$ recursos. El resto de los parámetros de los recursos se encuentran dentro de los rangos indicados en *National Wildfire Coordinating Group Fireline Handbook* [17].

Horas	Perímetro (km)	Área (ha)
1	0.3	0.7
2	1.0	5.6
3	1.3	9.6
4	1.8	15.9
5	2.0	20.3
6	2.2	24.3

Tabla 3.1: Parámetros del incendio forestal.

Rec.	Descripción	Núm.	Llegada (h)	Coste (€/h)	Coste Inicial (€)	Rend. (km/h)
1	Dozer	1	2.0	175	300	0.36
2	Tractor	1	2.5	150	500	0.45
3	Grupo I	1	0.5	125	500	0.20
4	Grupo II	1	1.0	175	600	0.25
5	Máquina I	1	1.5	75	400	0.09
6	Máquina II	1	1.5	100	900	0.10
7	Máquina III	1	1.0	125	600	0.15

Tabla 3.2: Parámetros de los recursos.

Con estos datos tendremos información suficiente sobre todos los parámetros del problema de selección de recursos. En general, la idea será que cada una de las bases aéreas cubran una tabla con los recursos disponibles. Esta información se introducirá en la función implementada en R, `seleccion_recursos` (A.1), de la siguiente forma:

```
> # Tabla crecimiento del fuego:
> incendio<-data.frame(Hj=seq(1,6),SPj=c(0.3,1,1.3,1.8,2,2.2),
+                        area=c(0.7,5.6,9.6,15.9,20.3,24.3))
>
> # Tabla de recursos contra el fuego:
> (recursos<-data.frame(i=seq(1,7),
+                       Des=c("Dozer","Tract","Grp1","Grp2","Maq1","Maq2","Maq3"),
+                       NUMi=rep(1,7),Ai=c(2,2.5,0.5,1,1.5,1.5,1),
+                       Ci=c(175,150,125,175,75,100,125),
+                       Pi=c(300,500,500,600,400,900,600),
+                       PRi=c(0.36,0.45,0.20,0.25,0.09,0.10,0.15)))
>
> # Coste hectarea
> coste_area<-100 # euros/hectarea
>
```

```
> #   Resolucion del problema mediante la funcion "seleccion_recursos"
> seleccion_recursos(incendio,recursos,coste_area)
```

Una vez que los datos son leídos por la función implementada, obtendremos la siguiente salida:

```
Iniciando resolucion del problema
Resolucion del problema terminada
Tiempo compilacion: 0.001 s
Seleccion de Recursos:
  Indice Descripcion Cantidad Periodo
1       1       Dozer        1       3
2       3        Grp1        1       3
3       4        Grp2        1       3
Perimetro cubierto:  1.36
El incendio se contiene en el periodo:  3
-----------------------------------------------------
Coste asignacion optima:  3785
Pre:  1400
Cost:  1425
NVC:  960
```

Por tanto, para contener el incendio minimizando el coste total, tendremos que emplear los recursos 1 (Dozer), 3 (Grupo1) y 4 (Grupo2) hasta el periodo 3. En este periodo, el incendio se contiene y el gasto que entraña la operación es de 3785 € (1400 en gastos iniciales por el uso de los recursos, 1425 en coste por las horas de uso y 960 en hectáreas de terreno quemado).

3.4. Análisis de sensibilidad

De forma análoga, podremos realizar el análisis de sensibilidad para observar cómo afecta la disponibilidad de un mayor o menor número de recursos a la solución del problema. Para ello, analizaremos los resultados que nos devuelve la función al considerar el número de recursos original (una unidad de cada tipo de recurso), dos unidades de cada recurso, 5 unidades y únicamente 10 unidades del primer recurso, respectivamente. Los resultados obtenidos se resumen en la siguiente tabla:

Escenario	Pre	Cost	NVC	C+NVC	Rec.	Cant.	Periodo
Original	1400	1425	960	3785	1	1	3
					3	1	3
					4	1	3
2 unidades de cada recurso	1300	1275	960	3535	1	1	3
					3	2	3
5 unidades de cada recurso	1500	375	70	1945	3	3	1
10 unidades del recurso 1	1200	2100	960	4260	1	4	3

Tabla 3.3: Variación del número de recursos.

Una ventaja importante del uso de la programación lineal entera es que el análisis de sensibilidad se puede realizar fácilmente sobre los parámetros del modelo para identificar aquéllos que pueden tener un efecto significativo en la solución óptima. En la Tabla 3.3 realizamos un análisis de sensibilidad sobre el número de recursos disponibles. En la primera situación disponemos de dos unidades de cada recurso; en este caso el coste total no aumentará, pues incrementamos el conjunto de posibles soluciones. En el siguiente escenario, con cinco unidades de cada recurso, pasa algo similar pues vemos como el coste total se reduce drásticamente; esto es debido fundamentalmente a que conseguimos contener el incendio en el periodo inicial (periodo 1).

En el último caso, tenemos únicamente diez unidades del primer recurso disponibles. Para contener el incendio de forma óptima, necesitaremos emplear 4 unidades del recurso 1 hasta el periodo 3. Esta selección de recursos tendrá asociado un coste de 4260 €.

De forma análoga, podemos ver cómo afecta el tiempo de llegada de los recursos al incendio, en la selección de un conjunto u otro de éstos y en sus periodos de actividad. Para realizar este análisis, suponemos que se dispone de diez unidades del primer recurso:

Escenario	Pre	Cost	NVC	C+NVC	Rec.	Cant.	Periodo
10 unidades del recurso 1	1200	2100	960	4260	1	4	3
Los recursos tardan el doble	1200	4025	2430	7655	1	1	5
					1	3	6
Los recursos tardan el triple	-	-	-	-	-	-	-

Tabla 3.4: Variación del tiempo de llegada de los recursos.

En la Tabla 3.4, observamos dos situaciones nuevas. En la primera se comprueba el buen funcionamiento de la función, que diferencia los recursos del tipo 1 que actúan hasta el periodo 5 de los que lo hacen hasta el 6. En la segunda situación, vemos como la lejanía de los recursos imposibilita la contención del incendio antes del último periodo (periodo 6). Ante esta última situación, la función `seleccion_recursos` proporciona la siguiente salida:

```
> seleccion_recursos(incendio,recursos,coste_area)
Iniciando resolución del problema
Resolución del problema terminada
Error al resolver el problema
```

Este resultado es lógico, ya que inicialmente el recurso 1 tardaba 2 horas en llegar al incendio. Si el tiempo de llegada se multiplica por 3, tardará 6 horas en llegar a éste, con lo que su producción será 0.

Supongamos ahora que disponemos, para el mismo incendio (Tabla 3.1), una cantidad diferente de los recursos cuyas características recogíamos en la Tabla 3.2:

Rec.	Descripción	Núm.	Llegada (h)	Coste (€/h)	Coste Inicial (€)	Rend. (km/h)
1	Dozer	0	2.0	175	300	0.36
2	Tractor	2	2.5	150	500	0.45
3	Grupo I	2	0.5	125	500	0.20
4	Grupo II	5	1.0	175	600	0.25
5	Máquina I	6	1.5	75	400	0.09
6	Máquina II	3	1.5	100	900	0.10
7	Máquina III	4	1.0	125	600	0.15

Tabla 3.5: Recursos disponibles.

Vamos a estudiar, bajo este escenario, el comportamiento de la función ante las dos restricciones presupuestarias establecidas (1.7) y (1.8).

Introduciendo en la función `seleccion_recursos` las cotas $B1$ y $B2$ respectivamente,

```
seleccion_recursos(incendio,recursos6,coste_area,B1=1000)
seleccion_recursos(incendio,recursos6,coste_area,B2=1100)
```

obtenemos las distintas selecciones óptimas para contener el incendio:

Escenario	Pre	Cost	NVC	C+NVC	Rec.	Cant.	Periodo
Sin Restricciones	1600	1125	960	3685	3	2	3
					7	1	3
$B_1 = 1000$	1000	1375	2030	4405	2	1	5
					3	1	5
$B_2 = 1100$	2200	1100	560	3860	3	2	2
					4	1	2
					7	1	2

Tabla 3.6: Restricciones sobre los costes.

En la Tabla 3.6 observamos cómo al marcar una cota para los costes fijos de los recursos, éstos se reducen 600 €. Sin embargo, este ahorro supondrá sobre todo un sacrificio en las hectáreas de terreno no calcinado. Mientras que en el caso sin restricciones, el incendio se contenía en el periodo 3, ahora se contiene en el 5, lo cual supone 10.7 hectáreas de terreno adicionales que se queman.

Por otro lado, al fijar la cota de los costes variables de los recursos, $B_2 = 1100$, vemos que se toman los recursos que menos tardan en llegar al incendio. Además, puesto que con estos recursos se intenta apagar el incendio lo antes posible (se intenta minimizar el tiempo de actuación), las hectáreas quemadas de terreno se reducen de 9.6 a 5.6 ha.

Capítulo 4

Modelo estocástico

La programación estocástica considera problemas de programación matemática en cuya formulación aparece algún elemento estocástico. Por tanto, mientras que en un problema determinístico de programación matemática, todos los parámetros que aparecen en su formulación son números conocidos, en programación estocástica dichos parámetros (o al menos algunos de los mismos) son desconocidos, aunque para ellos se conoce o se puede estimar su distribución de probabilidad.

Nuestro modelo estocástico se centra en el trabajo de tesis de Lee [13], basándose en la misma idea que el determinista, pero aumentando su alcance tal y como se comenta a continuación.

En este nuevo modelo, se supone que el número de recursos disponibles, rendimiento, tiempo necesario de llegada al incendio y costes tanto fijos como variables, son deterministas. Sin embargo, las características de crecimiento del incendio, tales como perímetro y área quemada, se suponen estocásticas. Para solventar el problema de la estocasticidad, se emplearán técnicas estadísticas, que permitirán reducir todos los posibles escenarios en un número finito y representativo de ellos.

Otro modelo aparecido en la literatura de lucha contra incendios, pero que no vamos a considerar aquí, es el descrito por Hu y Ntaimo [11]. Dicho modelo supone aleatoriedad en la aparición simultánea de un número finito de incendios. Establece un modelo de programación lineal entera mixta estocástica donde se resuelve de forma óptima la asignación de los recursos disponibles a cada uno de los incendios.

4.1. El modelo

Para la presentación del nuevo modelo, partimos de una simplificación del modelo anteriormente estudiado (1.1)-(1.8), donde eliminamos las variables L_j y N_j que se establecían por definición. De este modo, el problema a resolver quedaría simplificado del siguiente modo:

$$min \quad \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_i H_j D_{ij} + \sum_{i\in\mathcal{I}} P_i Z_i + \sum_{j\in\mathcal{J}} NVC_j Y_{j-1} \tag{2.1}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} (H_j - A_i) PR_i D_{ij} \geq \sum_{j\in\mathcal{J}} PER_j Y_{j-1} \tag{2.2}$$

$$\forall i \in \mathcal{I}, \quad \sum_{j\in\mathcal{J}} D_{ij} \leq Z_i \tag{2.3}$$

$$\forall j \in \mathcal{J}, \quad SP_j Y_{j-1} - \sum_{i\in\mathcal{I}} (H_j - A_i) PR_i D_{ij} \leq M\ Y_j \tag{2.4}$$

$$\sum_{i\in\mathcal{I}} P_i Z_i \leq B_1 \tag{2.5}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_i H_j D_{ij} \leq B_2 \tag{2.6}$$

siendo nuevamente $Y_0 = 1$, $M = \text{máx}\{SP_j :\ j \in \mathcal{J}\}$.

Además, se tomará por defecto $B1 = \sum_{i\in\mathcal{I}} P_i$ y $B2 = \text{máx}\{Hj \sum_{i\in\mathcal{I}} C_i :\ j \in \mathcal{J}\}$, por no ser restrictivos (no se puede obtener un valor mayor). De este modo, si no se indica nada en la función, ésta establece una cota superior suficientemente grande, tanto para los costes fijos de los recursos, como para los variables.

Partiendo de este modelo, diferenciaremos los parámetros estocásticos de los deterministas y de igual modo las variables de *primera etapa* de las de *segunda etapa*:

Conjuntos:

$\mathcal{I} = \{1, \ldots, n\}$:	Conjunto de índices de los recursos, indexado por i.
$\mathcal{J} = \{1, \ldots, m\}$:	Conjunto de índices de los periodos de tiempo, indexado por j.
Ω :	Conjunto de los distintos escenarios, indexado por ω.

Parámetros deterministas:

C_i :	Coste por hora del recurso i.
P_i :	Coste fijo por emplear el recurso i.
PR_i :	Rendimiento del recurso i (medido en km/h).
A_i :	Tiempo que tarda el recurso i en llegar al incendio.
H_j :	Periodo de tiempo j en el que se mide el incendio.
$B1$:	Cota superior para los costes fijos por la utilización de los recursos.
$B2$:	Cota superior para los costes variables por la utilización de los recursos.

Parámetros estocásticos:

Pr^ω :	Probabilidad de ocurrencia del escenario $\omega \in \Omega$.
PER_j^ω :	Incremento del perímetro del incendio en el periodo j, en el escenario $\omega \in \Omega$.
NVC_j^ω :	Incremento del NVC en el periodo j, en el escenario $\omega \in \Omega$.
SP_j^ω :	Perímetro del incendio en el periodo j, en el escenario $\omega \in \Omega$.

Variables de primera etapa:

Z_i :	Variable binaria que toma el valor 1 si el recurso i se emplea y 0 en otro caso.

Variables de segunda etapa:

D_{ij}^ω :	Variable binaria que toma el valor 1 si el recurso i se emplea en el escenario ω hasta el periodo j y 0 en otro caso.
Y_j^ω :	Variable binaria que toma el valor 1 cuando el incendio del escenario ω no está contenido en el instante j y 0 en otro caso (siendo $Y_0^\omega = 1$ para todo $\omega \in \Omega$).

En el modelo de programación lineal entera estocástica de dos etapas, en una primera etapa, se seleccionan una serie de recursos. Posteriormente, los recursos dados se emplean para comprobar la posible contención del incendio, o mejora (disminución) del valor de la función objetivo.

Ahora estamos en condiciones de formalizar nuestro modelo estocástico de dos etapas:

Problema máster: primera etapa.

$$min \quad \sum_{i \in \mathcal{I}} P_i Z_i + E[h(Z_i, \tilde{\omega})] \tag{3.1}$$

Sujeto a:

$$\sum_{i \in \mathcal{I}} P_i Z_i \leq B_1 \tag{3.2}$$

$$\forall i \in \mathcal{I}, \; Z_i \in \{0, 1\} \tag{3.3}$$

Subproblemas: segunda etapa.

Para cada $\omega \in \Omega$ tendremos un subproblema de la forma,

$$h(Z_i,\omega) = \quad min \quad \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_i H_j D_{ij}^{\omega} + \sum_{j\in\mathcal{J}} NVC_j^{\omega} Y_{j-1}^{\omega} \tag{3.4}$$

Sujeto a:

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_i H_j D_{ij}^{\omega} \leq B_2 \tag{3.5}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} (H_j - A_i) PR_i D_{ij}^{\omega} \geq \sum_{j\in\mathcal{J}} PER_j^{\omega} Y_{j-1}^{\omega} \tag{3.6}$$

$$\forall j \in \mathcal{J},\;\; SP_j^{\omega} Y_{j-1}^{\omega} - \sum_{i\in\mathcal{I}} (H_j - A_i) PR_i D_{ij}^{\omega} \leq M\; Y_j^{\omega} \tag{3.7}$$

$$\forall i \in \mathcal{I},\;\; \sum_{j\in\mathcal{J}} D_{ij}^{\omega} \leq Z_i \tag{3.8}$$

$$\forall i \in \mathcal{I},\; \forall j \in \mathcal{J},\; D_{ij}^{\omega},\; Y_j^{\omega} \in \{0,1\} \tag{3.9}$$

En la función objetivo de la primera etapa (3.1) se asegura que el coste fijo de los recursos, más el valor esperado de la suma de los costes variables de los recursos y los ocasionados por la quema del terreno, se reduzcan al mínimo. La restricción (3.2) asegura que la selección de los recursos satisface el presupuesto fijado para gastos fijos en el empleo de los recursos.

La función objetivo de la segunda etapa (3.4) asegura que para una selección de recursos determinada en la primera etapa, la suma de los costes variables por el empleo de los recursos, más el coste ocasionado por la devastación del terreno, se minimizan para cada escenario $\omega \in \Omega$. Las exigencias que establecen las restricciones (3.5)-(3.8) son análogas a las comentadas en el planteamiento determinista (1.2)-(1.8), teniendo en cuenta que cada una de estas restricciones se impone sobre cada posible escenario $\omega \in \Omega$.

4.2. Método para la resolución del problema estocástico

Nuestro modelo estocástico de dos etapas tiene variables de decisión binarias tanto de primera como de segunda etapa. Por tanto, pertenece a la clase de los denominados *stochastic integer programming* (SIP). Debido a la dificultad en el tratamiento de los SIP, han sido desarrollados muy pocos algoritmos para esta clase de problemas hasta ahora (cf. Birge and Louveaux [2]). En está sección, se revisa y aplica el algoritmo L^2 *con los cortes de Benders* para la resolución del modelo estocástico propuesto, que a su vez hace uso del denominado método L-shaped cuando se considera el problema relajado.

A grandes rasgos, se puede dividir el proceso de resolución en dos etapas:

Etapa 1, en la que se resuelve el problema relajado mediante el *método L-shaped* para obtener una cota inferior del objetivo del problema máster, la cual denotaremos por L. Este método se basa en la idea de resolver iterativamente el problema máster añadiéndole dos tipos de cortes resultantes de las sucesivas resoluciones de los subproblemas, denominados *cortes de Benders* (Benders [1]). Distinguiremos entre *cortes de factibilidad*, que añadiremos cuando algún subproblema no sea factible, y *cortes de optimalidad*, que añadiremos en caso contrario.

Mediante este método obtendremos el límite inferior de (3.1), que será esencial para que el algoritmo L^2 con los cortes de Benders converja más rápido.

Etapa 2, en la que se resuelve el modelo estocástico mediante el algoritmo L^2 con los cortes de Benders. Este método sigue la linea de la Etapa 1, pero diferenciándose en la incorporación de dos nuevos tipos de cortes, los *cortes de optimalidad* L^2, que requieren del cálculo de la cota inferior L y se emplean para la aproximación lineal por tramos del valor esperado de la función objetivo (3.1) (Birge and Louveaux [2]) y los *cortes de factibilidad* L^2 que desechan una solución en caso de no ser factible para algún subproblema no relajado (puede ocurrir que una solución sea factible para todo subproblema relajado pero no para algún subproblema entero, sin relajar).

Antes de exponer el algoritmo, introducimos una serie de preliminares. En cualquier caso, la presentación que realizamos aquí se orienta directamente a la aplicación en el problema real que nos ocupa. Para una revisión más profunda de la metodología, que incluye la demostración de la convergencia de los algoritmos de una manera constructiva, se puede acudir a Birge and Louveaux [2].

De forma más genérica denotaremos nuestro problema original como sigue:

$$\begin{aligned} min \quad & c^\top x + q^\top y && (4.1) \\ \text{Sujeto a:} \quad & && \\ & Ax \geq b && (4.2) \\ & Tx + Wy \geq r && (4.3) \\ & x, y \in \{0, 1\} && (4.4) \end{aligned}$$

donde q, W y r poseen aleatoriedad. Denotemos por $\tilde{\omega}$ la correspondiente variable aleatoria (discreta), y por ω una realización de $\tilde{\omega}$. Entonces, podemos descomponer el problema anterior en un problema de dos etapas, donde la primera tendrá únicamente parámetros deterministas y la segunda podrá tenerlos estocásticos y deterministas o únicamente estocásticos.

Primera etapa:

$$\begin{aligned} min \quad & c^\top x + E_{\tilde{\omega}}[f(x,\tilde{\omega})] & (5.1)\\ \text{Sujeto a:} \quad & & \\ & Ax \geq b & (5.2)\\ & x \in \{0,1\} & (5.3)\end{aligned}$$

donde para cada escenario $\omega \in \Omega$ tenemos:

Segunda etapa:

$$\begin{aligned} f(x,\omega) = \quad min \quad & q_\omega^\top y & (5.4)\\ \text{Sujeto a:} \quad & & \\ & W_\omega y \geq r_\omega - Tx & (5.5)\\ & y \in \{0,1\} & (5.6)\end{aligned}$$

Siguiendo la nueva notación aportada, describiremos los pasos a seguir para resolver las Etapas 1 y 2, que se corresponden con los métodos, L-shaped y L^2 con los cortes de Benders, respectivamente.

4.2.1. Método L-shaped

El método L-shaped es un algoritmo eficiente para la resolución de modelos estocásticos de dos etapas con todas las variables de decisión continuas. Está basado en el método de Benders (Benders [1]), que considera la estructura del problema dual, en problemas deterministas, y se adapta al caso estocástico prestando atención especial a la factibilidad. Aunque en nuestro caso el modelo es de variables enteras, el método L-shaped se emplea para la resolución del modelo propuesto relajado. De este modo, obtendremos la cota inferior L, que se empleará en la Etapa 2 para la generación de los cortes de optimalidad L^2. Para simplificar la notación, se supone que sólo hay restricciones de igualdad, aunque el procedimiento es análogo si se consideran las oportunas restricciones de desigualdad. Seguimos aquí Birge and Louveaux [2], p. 183.

Paso 0: Inicialización

Fijamos $r = s = 0$, que serán los contadores del número de restricciones de factibilidad y optimalidad respectivamente, $v = 0$ contador para el número de iteración del algoritmo, $\theta = -\infty$ y $\lambda = \infty$ las cotas inferior y superior del valor esperado de la suma ponderada de los objetivos de los subproblemas relajados y $\epsilon = 0{,}001$ el margen de error para la convergencia de la solución.

Paso 1: Resolución problema máster

Fijamos $v = v + 1$ y resolvemos el problema máster,

$$\begin{aligned}
min \quad & c^\top x + \theta & (6.1)\\
\text{Sujeto a:} \quad & Ax = b, & (6.2)\\
& D_l x \geq d_l, \qquad l = 1, \ldots, r & (6.3)\\
& E_l x + \theta \geq e_l, \quad l = 1, \ldots, s & (6.4)\\
& x \geq 0, \quad \theta \in \mathbb{R} & (6.5)
\end{aligned}$$

(x^v, θ^v) es una solución óptima. En el momento inicial no tendremos restricciones del tipo (6.4), por tanto, θ^v será lo más pequeño posible.

Paso 2: Factibilidad de los subproblemas y cortes de factibilidad

De acuerdo con Birge and Louveaux [2], p. 191, este paso se puede realizar como sigue. Comprobamos si la solución x^v obtenida, es factible para todos los subproblemas relajados. Para cada $\omega \in \Omega$, tomando x^v como un parámetro más, resolvemos el siguiente problema de programación lineal

$$\begin{aligned}
min \quad & \psi' = e^\top v^+ + e^\top v^- & (7.1)\\
\text{Sujeto a:} \quad & W_\omega y + Iv^+ - Iv^- = r_\omega - Tx^v & (7.2)\\
& y, v^+, v^- \geq 0 & (7.3)
\end{aligned}$$

donde $e^\top = (1, \ldots, 1)$ e I es la matriz identidad de dimensión igual al número de restricciones contempladas en la expresión (5.5).

Si para algún $\omega \in \Omega$, $\psi' > 0$, la solución x^v no es factible, por lo que tendremos que introducir un corte de factibilidad para cada escenario infactible como se indica a continuación.

Fijamos $r = r + 1$ y denotando por σ^v a los multiplicadores del simplex del problema (7.1)-(7.3), definimos para cada escenario infactible

$$\begin{aligned}
D_r &= (\sigma^v)^\top T & (8.1)\\
d_r &= (\sigma^v)^\top r_\omega & (8.2)
\end{aligned}$$

para generar restricciones (llamadas *cortes de factibilidad*) del tipo (6.3). Añadimos estas restricciones al conjunto (6.3) y volvemos al Paso 1.

En caso de que $\psi' = 0$ para todo $\omega \in \Omega$, tendremos que todos los escenarios son factibles y pasaremos al Paso 3.

Paso 3: Cortes de optimalidad

Para cada $\omega \in \Omega$ resolvemos el subproblema

$$\begin{aligned} min \quad & q(\omega)^\top y && (8.3)\\ \text{Sujeto a:} \quad & && \\ & W(\omega)y \geq r(\omega) - Tx^v, && (8.4)\\ & y \geq 0. && (8.5) \end{aligned}$$

Sean π_ω^v los multiplicadores del simplex asociados con la solución óptima de cada subproblema ω. Fijamos $s = s + 1$ y definimos

$$E_s = \sum_{\omega \in \Omega} p_\omega (\pi_\omega^v)^\top T \tag{9.1}$$

$$e_s = \sum_{\omega \in \Omega} p_\omega (\pi_\omega^v)^\top r_\omega \tag{9.2}$$

donde p_ω es la probabilidad asociada al escenario $\omega \in \Omega$.

Generamos las restricciones (llamadas *cortes de optimalidad*) del tipo (6.4). Añadimos estas restricciones al conjunto (6.4). Tomamos $\lambda^v = e_s - E_s x^v$ y comprobamos si $\lambda^v - \theta^v \leq \epsilon$, si se cumple paramos, x^v es una solución óptima y tomamos $L = c^\top x^v + \theta^v$. En caso contrario volvemos al Paso 1.

4.2.2. Método L^2 con los cortes de Benders

Una vez obtenida la cota inferior L mediante el método L-shaped, el algoritmo L^2 con los cortes de Benders se aplica para resolver el modelo propuesto (5.1)-(5.6). Éste es una variante del algoritmo L^2 o método L-shaped entero, introducido por Laporte and Louveaux [12] y también descrito con detalle en Birge and Louveaux [2]. Los pasos detallados del algoritmo son los siguientes.

Paso 0: Inicialización

Fijamos $r = s = l = z = 0$, que serán los contadores del número de restricciones de factibilidad, optimalidad, factibilidad L^2 y optimalidad L^2 respectivamente. También tomamos $v = 0$ contador para el número de iteración del algoritmo, $\theta = -\infty$ y $\lambda = \infty$ las cotas inferior y superior del valor esperado de la suma ponderada de los objetivos de los subproblemas, y $\epsilon = 0{,}001$ el margen de error para la convergencia de la solución.

Paso 1: Resolución problema máster

Fijamos $v = v + 1$ y resolvemos el problema máster,

$$\begin{aligned}
&min && c^\top x + \theta && && (10.1)\\
&\text{Sujeto a:} && Ax = b, && && (10.2)\\
& && D_l x \geq d_l, && l = 1, \dots, r && (10.3)\\
& && E_l x + \theta \geq e_l, && l = 1, \dots, s && (10.4)\\
& && G_l x \geq g_l, && l = 1, \dots, h && (10.5)\\
& && F_l x + \theta \geq f_l, && l = 1, \dots, z && (10.6)\\
& && x \in \{0, 1\}, \theta \in \mathbb{R} && && (10.7)
\end{aligned}$$

(x^v, θ^v) es una solución óptima. En el momento inicial no tendremos restricciones del tipo (10.4) ni (10.6), por tanto, θ^v será lo más pequeño posible.

Paso 2: Factibilidad de los subproblemas relajados y cortes de factibilidad

Al igual que en el Paso 2 del método L-shaped, comprobamos si la solución x^v es factible para todos los subproblemas relajados. Para ello resolvemos para cada $\omega \in \Omega$ el problema de programación lineal (7.1)-(7.3).

Si para algún $\omega \in \Omega$, $\psi' > 0$, la solución x^v no es factible para el subproblema relajado, por lo que tendremos que introducir un corte de factibilidad para cada escenario infactible, del mismo modo que en el Paso 2 del método L-shaped. Fijamos $r = r + 1$, añadimos los cortes de factibilidad generados al conjunto (10.3) y volvemos al Paso 1.

En caso de que $\psi' = 0$ para todo $\omega \in \Omega$, tendremos que todos los escenarios son factibles para los subproblemas relajados y vamos al Paso 3.

Paso 3: Cortes de Optimalidad

Como $\psi' = 0$, introducimos un corte de optimalidad del mismo modo que en el Paso 3 del método L-shaped. Fijamos $s = s + 1$, añadimos el corte de optimalidad al conjunto (10.4) y vamos al Paso 4.

Paso 4: Factibilidad de los subproblemas y cortes de factibilidad L^2

Comprobamos si la solución x_v obtenida, es factible para todos los subproblemas relajados. Para cada $\omega \in \Omega$, tomando x_v como un parámetro más del modelo, resolvemos el siguiente problema de programación lineal

$$
\begin{aligned}
min \quad & \psi'' = e^\top v^+ + e^\top v^- && (11.1)\\
\text{Sujeto a:} \quad & W_\omega y + Iv^+ - Iv^- = r_\omega - Tx^v && (11.2)\\
& y \in \{0,1\},\ v^+, v^- \geq 0 && (11.3)
\end{aligned}
$$

Si para algún $\omega \in \Omega$, $\psi'' > 0$, la solución x^v no es factible para el subproblema, por lo que tendremos que introducir un corte de factibilidad L^2. Fijamos $h = h + 1$ y usando x^v definimos

$$
\begin{aligned}
G_h &= \sum_{j \notin S^v} x_j - \sum_{j \in S^v} x_j && (12.1)\\
g_h &= 1 - |S^v| && (12.2)
\end{aligned}
$$

donde $S^v = \{j : x_j^v = 1\}$.

Mediante G_h y g_h se generan las restricciones (llamadas *cortes de factibilidad* L^2) del tipo (10.5). Añadimos estas restricciones al conjunto (10.5) y volvemos al Paso 1.

En caso de que $\psi'' = 0$ para todo $\omega \in \Omega$, tendremos que todos los escenarios son factibles y pasaremos al Paso 5.

Paso 5: Cortes de optimalidad L^2

Fijamos $z = z + 1$ y $\lambda^v = \min\{E[x^v, \tilde{\omega}],\ \lambda\}$, siendo $E[x^v, \tilde{\omega}]$ el valor esperado de la suma ponderada de los objetivos de los subproblemas, y usando L, x^v y $E[x^v, \tilde{\omega}]$ definimos

$$
\begin{aligned}
F_z &= (L - E[x^v, \tilde{\omega}])(\sum_{j \in S^v} x_j - \sum_{j \notin S^v} x_j) && (13.1)\\
f_z &= (L - E[x^v, \tilde{\omega}])(|S^v| - 1) + L && (13.2)
\end{aligned}
$$

De este modo generamos la restricción (llamada *corte de optimalidad* L^2) del tipo (10.6). Añadimos esta restricción al conjunto (10.6). Comprobamos si $\lambda^v - \theta^v \leq \epsilon$, si se cumple paramos, x^v es una solución óptima. En caso contrario volvemos al Paso 1.

4.3. Función `seleccion_recursos_L2`

Para el cálculo de la selección óptima de recursos, en el ataque inicial a un incendio forestal, suponiendo que se desconoce la evolución exacta y precisa del incendio, se implementa en `R`, la función `seleccion_recursos_L2` (A.2).

Para posibilitar el manejo correcto de la función implementada, realizaremos una descripción detallada de los argumentos de entrada y de los valores de salida que ésta proporciona. Nuevamente, deberemos mencionar que los argumentos de entrada de la función no coinciden con los parámetros del modelo, pues la función realiza cálculos internos que proporcionan, a partir de los argumentos de entrada, todos los parámetros necesarios para resolver el problema de programación lineal.

```
> seleccion_recursos_L2(incendio,recursos,coste_area,
                        Pw=rep(1/length(incendio),length(incendio)),
                        B1=F,B2=F,imp=T)
```

Argumentos de entrada:

El primer argumento de entrada, `incendio`, es una `lista` que contiene cada uno de los posibles escenarios `w`, relativos a posibles evoluciones para el incendio que pueden ocurrir. De este modo, si estimásemos que la evolución del incendio pudiese ser de 5 formas posibles, tendríamos que considerar un `incendio[[1]]`, `incendio[[2]]`, ..., `incendio[[5]]`. A su vez, los elementos de cada incendio w, `incendio[[w]]`, serán listas con la siguiente información:

`incendio[[w]]$Hj`: vector formado por los periodos de tiempo en los que se realiza la predicción acerca de la evolución del incendio. De este modo, el instante inicial será el periodo 0, y los sucesivos se corresponderán con cada uno de los periodos en los que se realiza la predicción de los parámetros (medidos en horas).

`incendio[[w]]$SPj`: vector de la misma longitud que `Hj`, que contiene el perímetro total del incendio en cada uno de los periodos (medido en km).

`incendio[[w]]$area`: vector de la misma longitud que `Hj`, que contiene el área total del incendio en cada uno de los periodos (medido en hectáreas).

El segundo argumento de entrada, `recursos`, es un `data.frame` que contiene la información de los recursos (aeronaves) disponibles. Los elementos de `recursos` contienen la siguiente información:

`recursos$i`: vector que contiene los índices asociados a cada tipo de recurso.

`recursos$DESi`: vector de la misma longitud que `i`, con una breve descripción o siglas de cada tipo de recurso.

`recursos$NUMi`: vector de la misma longitud que `i`, que indica el número de recursos de cada tipo.

`recursos$Ai`: vector de la misma longitud que `i`, que indica el tiempo que tardan en llegar los recursos al incendio dependiendo de su tipo y de la situación de su base (en horas).

`recursos$Ci`: vector de la misma longitud que `i`, con el coste por hora (variable) que supone emplear cada tipo de recurso (medido en €/hora).

`recursos$Pi`: vector de la misma longitud que `i`, con el coste (fijo) por emplear cada tipo de recurso (medido en €).

`recursos$PRi`: vector de la misma longitud que `i`, con el rendimiento que tiene cada tipo de recurso (en km/hora).

Otros argumentos de entrada son:

`coste_area`: valor numérico que establece el coste que supone la quema de una hectárea de terreno (medido en €/hectárea).

`Pw`: Opcional. Un vector con la misma longitud que el número de posibles evoluciones del incendio. Este vector informa de la probabilidad que tiene asociada la ocurrencia de cada una de las posibles evoluciones del incendio que se estima que pueden ocurrir. Por defecto, todas esas probabilidades son iguales.

`B1`: Opcional. Valor numérico que establece una cota superior para el coste fijo por la utilización de recursos (en €). Por defecto, con `B1=F`, la función no fija ninguna restricción en este tipo de coste.

`B2`: Opcional. Valor numérico que establece una cota superior para el coste variable por la utilización de recursos (en €). Por defecto, con `B2=F`, la función no fija ninguna restricción en este tipo de coste.

`imp`: Opcional. Parámetro lógico que define el tipo de salida que queremos que nos devuelva la función. Si `imp=T`, la función hará una impresión de pantalla con una breve explicación del resultado óptimo. En caso contrario, si `imp=F`, se obtendrán las variables del modelo matemático.

Valores de salida:

Diferenciaremos dos tipos de salida, según sea `imp=T` o `imp=F`. Si `imp=T`, la salida es en forma de texto y en ella se detalla: si el algoritmo encontró una solución óptima, el tiempo de resolución empleado por el algoritmo, el número de iteraciones realizadas para encontrar una solución óptima, la selección óptima de recursos para la contención de los posibles incendios (indicando el índice del recurso, descripción y unidades necesarias), el periodo en el que se contendrá cada incendio y hasta cuándo deberán actuar los recursos para la contención. Además, se muestra el coste total que supone la operación de contención y se indican los gastos por utilización de los recursos y los costes por hectáreas quemadas.

En caso contrario, si `imp=F`, la función devuelve una lista, `result`, con la siguiente información:

`result$Dij`: una lista con tantos elementos como posibles incendios. Cada uno de los elementos de la lista será una matriz entera con número de filas igual a la longitud de `i` y número de columnas igual a la longitud de `Hj`. En ella se detalla hasta qué periodo ha de actuar cada uno de los recursos, para realizar una extinción óptima en cada uno de los posibles incendios.

`result$Yj`: una lista con tantos elementos como posibles incendios. Cada uno de los elementos de la lista será un vector binario, de la misma longitud que `Hj`. Si el valor j-ésimo en el incendio `w` es 1, indica que el incendio `w` no está contenido en el periodo j.

`result$Zi`: vector entero, de la misma longitud que `i`. En el se detalla el número de unidades de cada tipo, que se deben emplear para extinguir de forma óptima, cualquiera de los posibles incendios estimados.

`result$Obj`: valor numérico que indica el óptimo de la función objetivo.

4.4. Asignación de recursos en un escenario simulado

Vamos a considerar 5 escenarios con diferentes valores para los parámetros correspondientes a la evolución de un incendio durante 6 periodos de tiempo de 1 hora. Estos parámetros se recogen en la Tabla 4.1 y serán, en cada uno de esos periodos, el perímetro y el área. Estos datos son los empleados en la tesis de Lee [13], que han sido generados, al igual que los tomados en Donovan y Rideout [6], mediante el software FARSITE (Finney [7]).

Para esta simulación, suponemos que el coste por hectárea quemada de terreno es de 100 €. Además dispondremos de siete tipos de recursos, cuyas características vienen recogidas en la Tabla 4.2 y que son los mismos que se toman para el análisis del modelo determinista en la Tabla 3.2. Los parámetros de los recursos se encuentran dentro de los rangos indicados en *National Wildfire Coordinating Group Fireline Handbook* [17].

Horas	Escenario (perímetro)					Escenario (área)				
	1	2	3	4	5	1	2	3	4	5
1	0.4	0.4	0.3	0.3	0.5	1.0	1.0	0.7	0.8	1.1
2	1.3	1.3	1.2	1.1	1.2	7.3	7.2	6.6	6.3	6.9
3	1.6	1.8	1.6	1.3	1.7	11.6	13.0	11.5	9.8	12.3
4	2.2	2.2	2.0	2.0	2.0	19.7	19.3	18.1	17.4	17.7
5	2.5	2.4	2.3	2.2	2.2	25.4	24.8	22.9	22.2	22.1
6	2.8	2.7	2.4	2.4	2.4	30.5	30.0	26.7	26.4	27.0

Tabla 4.1: Parámetros de la posible evolución del incendio forestal.

Recurso	Descripción	Número	Llegada (h)	Coste (€/h)	Coste Inicial (€)	Rend. (km/h)
1	Dozer	1	2.0	175	300	0.36
2	Tractor	1	2.5	150	500	0.45
3	Grupo I	1	0.5	125	500	0.20
4	Grupo II	1	1.0	175	600	0.25
5	Máquina I	1	1.5	75	400	0.09
6	Máquina II	1	1.5	100	900	0.10
7	Máquina III	1	1.0	125	600	0.15

Tabla 4.2: Parámetros de los recursos.

Con estos datos tendremos información suficiente sobre todos los parámetros del problema de selección de recursos. La idea será análoga al caso determinista, y es, que cada base cubra una tabla con los recursos disponibles. Esta información se introducirá en la función implementada en R, `seleccion_recursos_L2` (A.2), de la siguiente forma:

```
> # Tabla crecimiento del fuego, distintos escenarios:
> incendio<-list()
> (incendio[[1]]<-list(Hj=seq(1,6),SPj=c(0.4,1.3,1.6,2.2,2.5,2.8),
+                      area=c(1,7.3,11.6,19.7,25.4,30.5)))
> (incendio[[2]]<-list(Hj=seq(1,6),SPj=c(0.4,1.3,1.8,2.2,2.4,2.7),
+                      area=c(1,7.2,13,19.3,24.8,30)))
> (incendio[[3]]<-list(Hj=seq(1,6),SPj=c(0.3,1.2,1.6,2,2.3,2.4),
+                      area=c(0.7,6.6,11.5,18.1,22.9,26.7)))
> (incendio[[4]]<-list(Hj=seq(1,6),SPj=c(0.3,1.1,1.3,2,2.2,2.4),
+                      area=c(0.8,6.3,9.8,17.4,22.2,26.4)))
> (incendio[[5]]<-list(Hj=seq(1,6),SPj=c(0.5,1.2,1.7,2,2.2,2.4),
+                      area=c(1.1,6.9,12.3,17.7,22.1,27)))
>
> # Probabilidad de cada uno de los escenarios:
```

```
> Pw<-c(0.2,0.2,0.2,0.2,0.2)
> #==============================================================================
> # Tabla de recursos contra el fuego:
> (recursos<-data.frame(i=seq(1,7),NUMi=c(1,1,1,1,1,1,1),
+               DESi=c("Dozer","Tract","Grp1","Grp2","Maq1","Maq2","Maq3"),
+               Ai=c(2,2.5,0.5,1,1.5,1.5,1),Ci=c(175,150,125,175,75,100,125),
+               Pi=c(300,500,500,600,400,900,600),
+               PRi=c(0.36,0.45,0.20,0.25,0.09,0.10,0.15)))
> #==============================================================================
> # Coste hectarea
> coste_area<-100 # euros/hectarea
```

Una vez que los datos son leídos por la función implementada obtendremos la siguiente salida:

```
> seleccion_recursos_L2(incendio,recursos,coste_area)

Iniciando resolucion del problema
Resolucion del problema terminada
Tiempo compilacion: 15 s  ( 0.24 min)
Numero iteraciones:  121
=======================================================
Seleccion de Recursos:
Recurso: 1 ( Dozer ), Unidades: 1
Recurso: 2 ( Tract ), Unidades: 1
===================================================
Escenario 1 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 6
Recurso: 2 ( Tract ),Unidades: 1 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 2 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 6
Recurso: 2 ( Tract ),Unidades: 1 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 3 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 2 ( Tract ),Unidades: 1 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 4 : Periodo 5
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 2 ( Tract ),Unidades: 1 , Periodo: 5
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 5 : Periodo 5
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 2 ( Tract ),Unidades: 1 , Periodo: 5
=======================================================
Coste asignacion optima:  5215
Pre:  800
Cost:  1785
NVC:  2630
```

Por tanto, el problema se resuelve con 121 iteraciones en un tiempo de 15 segundos. Para contener el incendio minimizando el coste total, tendremos que emplear los recursos 1 (Dozer) y 2 (Tract). Estos recursos se emplearán dependiendo del escenario hasta los periodos 6 o 5 tal y como se muestra en la salida. Además, en caso de que se diese alguno de estos escenarios estimados, el incendio se contendría en los periodos indicados, Escenario 1 periodo 6, Escenario 2 periodo 6, Escenario 3 periodo 6, etc. Por último, obtenemos que el gasto estimado que entraña la operación es de 5215 € (800 en gastos iniciales, 1785 en coste por las horas de uso de los recursos y 2630 en hectáreas de terreno quemado).

Escenario	1	2	3	4	5
Coste óptimo	4960	5680	4910	3805	4635
Periodo de contención	3	5	4	3	5
	Recursos Seleccionados				
Dozer	1	1	1	1	1
Tractor		1	1		1
Grupo I	1		1	1	
Grupo II	1			1	
Máquina I		1			
Máquina II					
Máquina III	1				

Tabla 4.3: Resultados bajo conocimiento de la evolución del incendio.

En la Tabla 4.3 se muestra cuál sería el resultado óptimo en cada uno de los escenarios en caso de que se conociese con certeza la ocurrencia de ellos, dicho de otro modo cual sería el óptimo de cada uno de los escenarios bajo el planteamiento determinista.

Para detallar el análisis del planteamiento estocástico (información imperfecta) frente al determinista (información perfecta), recurrimos al *Expected Value of Perfect Information* (EVPI), que se define como la diferencia entre el valor objetivo del modelo estocástico y el promedio de los valores objetivos obtenidos bajo información perfecta, y al *Value of Stochastic Solution* (VSS), que se define como la diferencia entre el valor esperado (sobre todos los escenarios) tomando los recursos obtenidos al resolver el problema determinista con los parámetros estocásticos promediados, y el valor óptimo obtenido al resolver el problema estocástico.

De este modo tendremos que el $EVPI$ es,

$$EVPI = 5215 - (0{,}2 \cdot 4960 + 0{,}2 \cdot 5680 + 0{,}2 \cdot 4910 + 0{,}2 \cdot 3805 + 0{,}2 \cdot 4635) = 417.$$

Éste puede ser pensado como el valor que merece la pena pagar por la información perfecta.

Por otro lado, al resolver el modelo determinista con el promedio de los parámetros, obtenemos que la selección de recursos óptima es emplear los recursos 1, 2 y 3 (Dozer, Tractor y Grupo I). Estos recursos se emplean hasta el periodo 4, obteniendo un valor óptimo de 4944 €. Si ahora resolvemos los subproblemas con esta selección de recursos fijada, obtenemos que el valor estimado es de 4048 €, y si a esto le añadimos los costes fijos 1300 € (no están contemplados en los subproblemas), obtenemos que el coste por emplear esta selección es de 5348 €. Por tanto, tenemos que

$$VSS = 5348 - 5215 = 133.$$

Este valor puede ser pensado como el coste que merece la pena pagar por el uso de la solución estocástica en lugar de la solución de valor medio.

Se puede concluir, que sin información perfecta, el enfoque de la programación estocástica es de gran interés, pues proporciona un método de toma de decisiones robusto en condiciones de incertidumbre para esta clase de problemas.

4.5. Análisis de sensibilidad

Una ventaja importante del uso de la programación lineal entera es que el análisis de sensibilidad se puede realizar fácilmente sobre los parámetros del modelo para identificar aquéllos que pueden tener un efecto significativo en la solución óptima. Observemos cómo varía la solución en función del número de recursos disponibles:

- Sin disponer de ningún recurso del tipo 1 y 2, y disponiendo de dos unidades del 3 y una del 4, 5, 6 y 7, respectivamente:

```
Iniciando resolucion del problema
Resolucion del problema terminada
Tiempo compilacion: 5 s  ( 0.08 min)
Numero iteraciones:  43
=======================================================
Seleccion de Recursos:
Recurso: 3 ( Grp1 ), Unidades: 2
Recurso: 4 ( Grp2 ), Unidades: 1
Recurso: 7 ( Maq3 ), Unidades: 1
===================================================
Escenario 1 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 2 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 3 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 4 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 5 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
```

```
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
=======================================================
Coste asignacion optima:  4909
Pre:  2200
Cost:  1545
NVC:  1164
```

Observamos cómo los recursos seleccionados en este caso son 2 unidades del recurso 3 (Grupo 1), 1 unidad del recurso 4 (Grupo 2) y otra del recurso 7 (Máquina 3), con coste óptimo de 4909 €. Si observamos la actuación de los recursos en los diferentes escenarios, vemos que todos ellos actúan en los distintos incendios a excepción del recurso 4 que no actúa en el 3. Esto significa que en caso de que se produjese el escenario 3, el Grupo 2 dejaría de trabajar en el incendio.

- Sin disponer de los recurso del tipo 2, 3 y 7, y disponiendo de dos unidades del 1 y una del 4, 5 y 6:

```
Iniciando resolucion del problema
Resolucion del problema terminada
Tiempo compilacion: 2 s  ( 0.03 min)
Numero iteraciones:  27
=======================================================
Seleccion de Recursos:
Recurso: 1 ( Dozer ), Unidades: 2
===================================================
Escenario 1 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 2 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 2 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 2 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 3 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
Escenario 4 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 6
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 5 : Periodo 6
---------------------------------------------------
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 5
Recurso: 1 ( Dozer ),Unidades: 1 , Periodo: 6
=======================================================
Coste asignacion optima:  5407
Pre:  600
Cost:  1995
NVC:  2812
```

En este caso se seleccionan dos unidades del recurso 1 (Dozer), con un coste asociado de 5407 €. En este caso, los recursos, aunque son de un mismo tipo, actúan en algunos escenarios hasta un mismo periodo (indicando que las dos unidades se emplean hasta el mismo periodo) y en otros casos no (indicando hasta que periodo actúa cada unidad).

- Sin disponer del recurso del tipo 1, y disponiendo de dos unidades del 2,3,6 y 7 y una del 4 y 5:

```
Iniciando resolucion del problema
Resolucion del problema terminada
Tiempo compilacion: 663 s  ( 11.04 min)
Numero iteraciones:  904
=======================================================
Seleccion de Recursos:
Recurso: 3 ( Grp1 ), Unidades: 2
Recurso: 4 ( Grp2 ), Unidades: 1
Recurso: 7 ( Maq3 ), Unidades: 1
===================================================
Escenario 1 : Periodo 3
---------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
Escenario 2 : Periodo 3
--------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 3 : Periodo 3
--------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 4 : Periodo 3
--------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Escenario 5 : Periodo 3
--------------------------------------------------
Recurso: 3 ( Grp1 ),Unidades: 2 , Periodo: 3
Recurso: 4 ( Grp2 ),Unidades: 1 , Periodo: 3
Recurso: 7 ( Maq3 ),Unidades: 1 , Periodo: 3
======================================================
Coste asignacion optima:  4909
Pre:  2200
Cost:  1545
NVC:  1164
```

Vemos que se seleccionan dos unidades del recurso 3 (Grupo 1) y una de los recursos 4 (Grupo 2) y 7 (Máquina 3) con un coste asociado de 4909 €. Además, se contiene el incendio en todos los posibles escenarios en el periodo 3, y empleándose todos los recursos seleccionados hasta este mismo periodo a excepción del recurso 4, que no se emplea en el escenario 4.

Una observación importante se refiere a la diferencia que se observa en los tiempos de resolución del problema según se trabaja con un número de recursos u otro. Cuando el número de recursos que se pueden emplear es menor que 7 unidades, los tiempos de resolución son en media menores de 15 segundos, sin embargo, cuando aumentamos dicho número de recursos, los tiempos crecen de forma considerable, estando próximos a los 30 segundos con 8 recursos, 2 minutos con 9 y hasta

10 minutos con 10 unidades.

De forma análoga, podremos realizar el análisis de sensibilidad modificando los tiempos de llegada al incendio de cada uno de los recursos. Para realizar este análisis, disponemos de siete unidades del primer recurso:

Escenario	Pre	Cost	NVC	C+NVC	Recurso	Cantidad
10 unidades del recurso 1	1500	2520	1164	5184	1	5
tiempos de llegada el doble	1200	4095	2812	8107	1	4
tiempos de llegada el triple	-	-	-	-	-	-

Tabla 4.4: Resultados cuando se varía el tiempo de llegada de los recursos.

En la Tabla 4.4 observamos dos situaciones nuevas. En la primera, se aprecia cómo al duplicarse los tiempos de llegada, el incendio tarda mucho más en apagarse. Esto se puede apreciar claramente en el NVC, que se incrementa en más del doble. En la segunda situación, vemos como la lejanía de los recursos imposibilita la contención del incendio antes del último periodo (periodo 6). Ante esta última situación, la función `seleccion_recursos_L2` produce la siguiente salida:

```
> seleccion_recursos_L2(incendio,recursos7,coste_area)

Iniciando resolucion del problema
El problema no tiene solución
Escenarios Infactibles: 1 2 3 4 5
```

Este resultado es lógico, ya que inicialmente el recurso 1 tardaba 2 horas en llegar al incendio. Si el tiempo de llegada se multiplica por 3, tardará 6 horas en llegar a éste, con lo que su producción será 0 en cualquiera de los escenarios.

Veamos cómo se ve afectada la selección de los recursos y los costes que ésta supone, cuando el coste por hectárea quemada se ve modificado. En este caso, observaremos las modificaciones que se producen al considerar una unidad de cada recurso sobre tres situaciones distintas, una con un coste por hectárea quemada de 20 €, otra de 100 € y una tercera con un coste de 200 €.

Coste por hectárea	Pre	Cost	NVC	C+NVC	Recurso	Cantidad
20 €	800	1785	526	3111	1,2	1,1
100 €	800	1785	2630	5215	1,2	1,1
200 €	2000	1835	2796	6631	1,3,4,7	1,1,1,1

Tabla 4.5: Efecto de la variación de los costes por hectárea de terreno quemada.

En la Tabla 4.5, vemos que entre el primer y el segundo caso (20 €/ha y 100 €/ha), no se produce ninguna modificación en la selección de los recursos ni en el tiempo de uso de éstos (pese a que este dato no se refleje en la tabla, debido a la complejidad en su exposición, sí se puede observar que tanto los costes fijos, Pre, como los variables, Cost, se mantienen inalterados), sin embargo, el coste por hectáreas de terreno quemado si se modifica considerablemente.

Si comparamos estas dos situaciones con la última, en la que el coste por hectárea de terreno quemado es de 200 €, vemos que la selección óptima se modifica considerablemente. En este caso, para evitar que se queme una extensión amplia de terreno, lo cual ocasionaría un incremento considerable del coste, se selecciona un mayor número de recursos con tiempos de llegada relativamente bajos. Esta selección ocasiona un gran aumento en los costes fijos por la utilización de recursos, pero evita que el incendio crezca tanto y ocasione grandes pérdidas en terreno quemado.

Por último, analizamos cómo influye la imposición de restricciones presupuestarias, tanto computacionalmente, como en la selección de recursos. Supongamos que disponemos para los mismos escenarios (Tabla 4.1) de los siguientes recursos:

Rec.	Descripción	Núm.	Llegada (h)	Coste (€/h)	Coste Ini. (€)	Rend. (km/h)
1	Dozer	1	2.0	175	300	0.36
2	Tractor	0	2.5	150	500	0.45
3	Grupo I	1	0.5	125	500	0.20
4	Grupo II	2	1.0	175	600	0.25
5	Máquina I	1	1.5	75	400	0.09
6	Máquina II	1	1.5	100	900	0.10
7	Máquina III	1	1.0	125	600	0.15

Tabla 4.6: Nuevas características de los recursos disponibles.

Introduciendo en la función `seleccion_recursos_L2`, el problema sin ninguna restricción, con la restricción $B_1 = 1200$ y con la restricción $B_2 = 1800$, respectivamente, obtenemos las distintas selecciones óptimas para contener el incendio. Éstas se muestran a continuación:

Presupuesto	Pre	Cost	NVC	C+NVC	Recurso	Cantidad	Iteraciones
Sin restricción	2000	1845	1164	5009	1,3,4	1,1,2	220
B1=1200	1200	1965	2630	5795	1,2,5	1,1,1	4
B2=1800	2300	1695	1164	5159	3,4,7	1,2,1	164

Tabla 4.7: Efecto de las restricciones presupuestarias.

En la Tabla 4.7 observamos cómo al marcar una cota para los costes fijos de los recursos, éstos se reducen a 1200 €. Sin embargo, este ahorro supondrá sobre todo un sacrificio en las hectáreas de terreno no calcinado. Mientras que en el caso sin restricciones, el incendio producía unos costes por terreno quemado de 1164 €, ahora supone un coste de 2630 €.

Por otro lado, al fijar la cota de los costes variables de los recursos, $B_2 = 1800$, vemos que se toman los recursos que menos tardan en llegar al incendio. Además, puesto que estos recursos intentan apagar el incendio lo antes posible (se intenta minimizar el tiempo de actuación), el coste producido por las hectáreas quemadas de terreno se mantiene.

Además, cabe mencionar dos observaciones interesantes. La primera es que al imponer restricciones presupuestarias, el tiempo de compilación se reduce considerablemente. Inicialmente, para llegar a la selección óptima se necesitaban 220 iteraciones, mientras que con las restricciones presupuestarias se reducen a 4 y 164, respectivamente. La primera restricción (asociada al valor B_1) es la que minimiza de forma muy significativa el número de iteraciones, esto es debido a que la restricción se le impone al problema máster, lo cual limita de forma considerable el número de posibles soluciones que se consideran en los subproblemas.

La segunda observación hace referencia a la segunda restricción presupuestaria (asociada al valor B_2), y es que, como se puede observar en la Tabla 4.7, al imponer la condición $B_2 = 1800$, ésta se impone a cada uno de los escenarios, o lo que es lo mismo, la restricción se aplica en cada uno de los subproblemas, y no al coste estimado (Cost). Por ello si fijamos $B_2 = 1695$, obtenemos:

```
> seleccion_recursos_L2(incendio,recursos8,coste_area,B2=1695)

Iniciando resolucion del problema
El problema no tiene solución
Escenarios Infactibles: 2 5
```

Esto nos indica que es imposible contener el incendio, si se produce el escenario 2 o 5, con un presupuesto para gastos variables menor a 1695 €. Lo cual no implica, como comentábamos antes, que el coste variable estimado por emplear los recursos no pueda ser 1695 €, tal y como se muestra al fijar la segunda restricción presupuestaria con $B_2 = 1800$.

Capítulo 5

Modelo determinista incluyendo asignaciones horarias

Con el fin de atender posibles requisitos alternativos que atañen a las aeronaves encargadas de la extinción del incendio, se modifica el modelo del Capítulo 3 con el fin de que éste tenga en consideración aspectos tales como:

- La asignación de los recursos a lo largo de diferentes periodos de tiempo, de forma que no todos los recursos seleccionados han de operar desde el instante inicial, sino que pueden incorporarse al incendio en cualquier instante en el cual se precisen.
- El algoritmo debe de poder ejecutarse en cualquier instante de tiempo, no únicamente en el momento de detección del incendio. Por tanto, se deben de tener en cuenta, no sólo las aeronaves presentes en las bases, sino también aquellas que están actuando sobre el incendio en cuestión. Hasta ahora, se suponía que al ejecutar la función creada para resolver el problema de la selección de recursos, el incendio estaba desatendido, no había ninguna aeronave en él. En el nuevo modelo, se debe considerar la posibilidad de que existan aeronaves trabajando en el momento de ejecución de la función, así como aeronaves que aunque no estén trabajando en el incendio, hayan realizado otros trabajos en la extinción de otros incendios, y que por tanto el tiempo de uso de estas en la tarea de extinción sea menor (por tener consumido tiempo de actividad diaria). Por tanto será de gran importancia tener constancia del tiempo de vuelo diario de cada aeronave.
- Si el incendio no está controlado, debe de haber al menos una aeronave en cada frente.
- Han de considerarse los tiempos de vuelo diario (tiempo total de actividad diaria), tiempos de vuelo (sin descansar) y tiempos de descanso. Estos datos se establecen en la *Circular Operativa 16-B* [15], donde se fija que el tiempo de vuelo diario en incendios forestales ha de ser inferior a 8 horas, el tiempo de vuelo (sin descansos) inferior a 2 horas y el tiempo de descanso entre vuelo y vuelo ha de ser de 40 minutos.

- Se obviarán las restricciones presupuestarias y se incorporarán restricciones en el tiempo de contención del incendio.

Debido a la complejidad adicional del nuevo problema, el modelo sólo se propone de forma determinista. Si la versión estocástica del modelo anterior ya era lenta cuando se consideraba un gran número de aeronaves, el nuevo modelo, que incorpora un número considerablemente mayor de variables binarias y restricciones, hace pensar en una versión estocástica de gran complejidad, que no se abordará en este momento.

5.1. El nuevo modelo

Haremos unas observaciones iniciales sobre los datos que se introducen en la función creada en R, y en el pre-procesado que vamos a realizar. Los periodos de tiempo se construirán tomando el máximo común divisor entre los tiempos de llegada al incendio con los que vamos a trabajar, los tiempos de vuelo de las aeronaves (sin contar descansos), los tiempos de descanso de éstas y el tiempo de vuelo que llevan realizando. A partir de estos datos, se obtienen los periodos de tiempo H_j con los que vamos a trabajar, que irán desde un periodo temporal anterior al actual (H_0) hasta el último momento del que se tiene una estimación de la evolución del incendio (H_m). En caso de que no se tenga estimación de la evolución del incendio en algún periodo H_j, se tomarán para ese periodo, los datos del siguiente periodo temporal del cual se tenga una estimación.

Introducimos la notación necesaria, para la formulación del problema:

Conjuntos:

$\mathcal{I} = \{1, \dots, n\}$:	Conjunto de índices de los recursos, indexado por i.
$\mathcal{J} = \{1, \dots, m\}$:	Conjunto de índices de los periodos de tiempo, indexado por j.

Parámetros referentes a los recursos:

C_i :	Coste por hora del recurso i.
PR_i :	Rendimiento del recurso i (km/h).
A_i :	Tiempo que tarda el recurso i en llegar al incendio.
TV_i :	Tiempo de vuelo (máximo) que debe realizar la aeronave i.
TA_i :	Tiempo de vuelo de la aeronave i, desde su último descanso.
TD_i :	Tiempo de descanso que debe realizar la aeronave i.
TDM :	Número de periodos de tiempo que comprende un descanso.
$TDMI_i$:	Número de periodos de tiempo que lleva descansados la aeronave i cuando entra en el incendio.
TVD_i :	Tiempo de vuelo diario (máximo) que debe realizar la aeronave i.
TVT_i :	Tiempo de vuelo que lleva realizado la aeronave i.

Parámetros referentes al incendio:

PER_j :	Incremento del perímetro del incendio en el periodo j.
NVC_j :	Incremento del NVC (hectáreas quemadas por coste de la hectárea) en el periodo j.
SP_j :	Perímetro del incendio en el periodo j.
H_j :	Periodo de tiempo j.
$frentes$:	Número de frentes que posee el incendio.

Variables de decisión:

D_{ij} :	Variable binaria que toma el valor 1 cuando el recurso i finaliza su actuación en el periodo j y 0 en otro caso.
DES_{ij} :	Variable binaria que toma el valor 1 si el recurso i se encuentra descansando en el periodo j y 0 en otro caso.
FD_{ij} :	Variable binaria que toma el valor 1 si el recurso i se encuentra finalizando el descansando en el periodo j y 0 en otro caso.
B_{ij} :	Variable binaria que toma el valor 1 cuando el recurso i se selecciona para actuar desde el periodo j y 0 en otro caso.
Y_j :	Variable binaria que toma el valor 1 cuando el incendio no está contenido en el instante j y 0 en otro caso.
Z_i :	Variable binaria que toma el valor 1 si el recurso i se emplea y 0 en otro caso.

Describimos paso a paso la formulación matemática que modela la función objetivo y las restricciones que se quieren cumplir:

Función Objetivo:

$$\begin{aligned} min \quad & \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} C_iH_{j+1}D_{ij} - \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} H_jC_iBij - \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} (H_{j+1}-H_j)C_iDES_{ij} + \\ & \sum_{j\in\mathcal{J}} NVC_jY_{j-1} + Y_m + \sum_{i\in\mathcal{I}} Z_i + \sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} 0,001H_{j+1}DES_{ij} + \sum_{j\in\mathcal{J}} \frac{Hj_{j+1}}{H_{m+1}}\, Y_{j-1} \end{aligned} \tag{14.1}$$

La función objetivo, (14.1), la podemos descomponer en 3 partes. La primera, formada por los cuatro primeros términos, pretende minimizar el tiempo de utilización de los recursos y el tiempo de contención del incendio forestal. Para establecer una comparación equilibrada, se ponderan ambos aspectos mediante los costes de empleo de los recursos y los costes por hectárea quemada de terreno. Además, debemos mencionar que en el objetivo se descuentan los tiempos en los que las aeronaves están en reposo, pues se asume que en estos periodos no están generando gastos.

La segunda parte está constituida por los 2 siguientes términos. Con esto, se pretende evitar errores del tipo, que Y_m o los Z_i valgan indiferentemente 0 o 1. Si no se incluyesen estos sumandos en el objetivo, la función podría devolvernos que sólo se emplea 1 recurso de los n disponibles y sin embargo que Z_i fuese 1 para todo $i \in \mathcal{I}$, lo cuál sería incorrecto.

Los dos último términos, tienen un significado un poco diferente. El primero, se pondera por 0,001 para restarle importancia frente al tercer término de la función objetivo, que es el que tiene un verdadero significado. Éste se introduce en el objetivo, con el fin de "compactificar" los intervalos de descanso. Al presentarse las variables DES_{ij} multiplicadas por el parámetro H_{j+1}, el objetivo prefiere tomar los descansos lo antes posible, lo cuál produce dicha compactificación en los periodos de descanso. Si este término no se añadiese al objetivo, la solución podría devolver resultados no deseados. El último pretende que, en caso de que en algún periodo, el incremento del área de terreno quemada sea 0 (para algún $j \in \mathcal{J}$, $NVC_j = 0$), siempre se tome el menor tiempo posible de contención del incendio. Este último caso se puede producir si no se tiene estimación del incendio en algún periodo H_j, pues en dicho caso, la función toma la predicción del periodo siguiente.

Restricciones:

$$\sum_{j \in \mathcal{J}} Y_j \leq B_1 - 2 \tag{14.2}$$

La primera restricción es opcional. Ésta permite fijar una cota para el tiempo de contención del incendio. Si se desease contener el incendio antes del periodo j, se establecerá $B_1 = j$. En caso de que se quisiese deshabilitar esta restricción, bastaría tomar B_1 igual al número de periodos de tiempo que posee H_j, $m+1$.

$$\begin{aligned} \sum_{j \in \mathcal{J}} PER_j Y_{j-1} \leq & \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (H_{m+1} - H_j - A_i) PR_i B_{ij} - \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (H_{m+1} - H_{j+1}) PR_i D_{ij} \\ & - \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} (H_{j+1} - H_j) PR_i DES_{ij} \end{aligned} \tag{14.3}$$

La restricción (14.3), tiene el mismo significado que (1.2). Nos indica que en algún periodo de tiempo el perímetro construido por los recursos tiene que cubrir el perímetro del incendio. Además, impone la condición de que el lado derecho de la desigualdad sea no negativo.

$$\forall j \in \mathcal{J}', \ \sum_{i \in \mathcal{I}} \sum_{k=1}^{j-l_i} B_{ik} - \sum_{i \in \mathcal{I}} \sum_{k=1}^{j-1} D_{ik} - \sum_{i \in \mathcal{A}_j} DES_{ij} \geq frentes \ Y_j \tag{14.4}$$

La restricción (14.4), fija que, mientras el incendio no esté contenido, el número mínimo de aeronaves actuando en el incendio en cada periodo de tiempo ha de ser mayor o igual que el número de frentes, salvando alguna excepción como se muestra a continuación. Esta restricción hay que tratarla con cierto cuidado. Puede ocurrir que no exista ninguna aeronave situada en el incendio en el instante inicial, y por tanto, el incendio estará desatendido durante un cierto tiempo. Este hecho inevitable supone un problema en el planteamiento del modelo, pues en el instante inicial será imposible que haya alguna aeronave trabajando. Para solventar el problema, se toman en la restricción únicamente los periodos de tiempo mayores al menor tiempo de llegada de una aeronave al incendio. Además, mediante este planteamiento, se impondrá la condición de que el incendio quede atendido lo antes posible por al menos una aeronave. Del planteamiento anterior, surge la definición del conjunto $\mathcal{J}'$,

$$\mathcal{J}' = \{j \in \mathcal{J} : \ H_{j+1} > H_2 + min_{i \in \mathcal{I}} A_i\}.$$

Además, se debe considerar que mientras la aeronave se encuentra viajando al incendio, está no está actuando. Por tanto se ha de definir el valor l_i, como el número de periodos de tiempo que tarda la aeronave en llegar al incendio,

$$l_i = \{j \in \mathcal{J} : \ H_{j+1} = H_2 + A_i\} + \textit{Número periodos de descanso durante el viaje}.$$

Por último, tampoco se han de tener en consideración los descansos realizados durante el tiempo de viaje, pues estos periodos ya se consideran de no actuación, por lo que procede definir el conjunto,

$$\mathcal{A}_j = \{i \in \mathcal{I} : \ H_{j+1} > H_2 + A_i\}.$$

$$\forall i \in \mathcal{I}, \ \sum_{j \in \mathcal{J}} D_{ij} \leq Z_i \tag{14.5}$$

La desigualdad (14.5) tiene el mismo significado que (1.3), impone que si un recurso i se usa hasta algún periodo j, Z_i tomará el valor 1. Además, si se emplea el recurso i, éste sólo podrá operar hasta un periodo determinado, pues la suma de los D_{ij} no podrá ser mayor que 1.

$$\forall i \in \mathcal{I}, \ \begin{cases} B_{i1} + \displaystyle\sum_{j=2}^{m} (m+1) B_{ij} \leq m Z_i & , \text{ si } \ \mathrm{A_i = 0} \\ \\ \displaystyle\sum_{j \in \mathcal{J}} B_{ij} \leq Z_i & , \text{ en otro caso} \end{cases} \tag{14.6}$$

La desigualdad (14.6) establece algo análogo a la restricción (14.5), pero aplicándose a los periodos de comienzo de actividad. Cada recurso se podrá seleccionar únicamente una vez, y si un recurso i se selecciona en algún periodo j, Z_i tomará el valor 1. Además, se hace una distinción, si el recurso está en el incendio ($A_i = 0$), en caso de emplearse, se hará desde el instante inicial.

$$\begin{aligned} \forall i \in \mathcal{I}, \ \forall j \in \mathcal{J}, \quad TV_i - TA_i \ \geq \ & \sum_{k=1}^{j} (H_{j+1} - H_k) B_{ik} - \sum_{k=1}^{j} (H_{j+1} - H_{k+1}) D_{ik} \\ & - \sum_{k=1}^{j} \frac{TD_i}{TDM} DES_{ik} - \sum_{k=1}^{j} TV_i FD_{ik} \end{aligned} \tag{14.7}$$

$$
\begin{aligned}
\forall i \in \mathcal{I},\ \forall j \in \mathcal{J}, \quad -TA_i \ \geq \ & \sum_{k=1}^{j}(H_{j+1}-H_k)B_{ik} - \sum_{k=1}^{j}(H_{j+1}-H_{k+1})D_{ik} \\
& - \sum_{k=1}^{j}\frac{TD_i}{TDM}DES_{ik} - \sum_{k=1}^{j}TV_iFD_{ik}
\end{aligned}
\tag{14.8}
$$

$$
\forall i \in \mathcal{I},\ \forall j \in \mathcal{J}, \quad
\begin{cases}
\displaystyle\sum_{k=j-TDM+1}^{j} DES_{ij} \geq TDM\ FD_{ij} & ,\ \text{si}\ \ \text{j} - \text{TDM} \geq 0 \\
\displaystyle\sum_{k=1}^{j} DES_{ij} \geq (TDM - TDMI_i)\ FD_{ij} & ,\ \text{en otro caso}
\end{cases}
\tag{14.9}
$$

Las restricciones (14.7), (14.8) y (14.9), no tienen sentido por separado. Se establecen para obligar a las aeronaves a realizar los correspondientes descansos durante el tiempo establecido. Es aquí donde juega un papel muy importante la variable DF_{ij}, pues esta es la que "proporciona" a la aeronave tiempo de actuación tras la realización correspondiente del descanso.

Las dos primeras restricciones (14.7) y (14.8), obligan a realizar descansos una vez las aeronaves realizan un vuelo de TV_i horas. Y obliga a las aeronaves a volver a actuar una vez transcurrido el tiempo de descanso, TD_i.

La restricción (14.9), por otro lado, obliga a que la variable DF_{ij} tome el valor 1 cuando la aeronave i se encuentra finalizando el descansando en el periodo j, y por tanto, le permite volver a volar durante TV_i horas.

$$
\begin{aligned}
\forall j \in \mathcal{J}, \quad M\ Y_j \ \geq \ & -\sum_{i\in\mathcal{I}}\sum_{k=1}^{j}(H_{j+1}-H_k-A_i)PR_iB_{ik} + \sum_{i\in\mathcal{I}}\sum_{k=1}^{j}(H_{j+1}-H_{k+1})PR_iD_{ik} \\
& + \sum_{i\in\mathcal{I}}\sum_{k=1}^{j}(H_{k+1}-H_k)PR_iDES_{ik} + SP_jY_{j-1}
\end{aligned}
\tag{14.10}
$$

La restricción (14.10), establece lo mismo que (1.4), si el incendio está contenido o no en el periodo j. Si el perímetro construido por los recursos es menor que el perímetro del incendio, entonces no está contenido, y por tanto Y_j valdrá 1. Si por el contrario el perímetro construido por los recursos es mayor que el del incendio, Y_j podrá valer 0 (la función objetivo, que busca minimizarse, prefiere que los valores de los Y_j sean 0). El valor de M lo fijamos en este caso como $M = \text{máx}_{j\in\mathcal{J}} SP_j + \sum_{i\in\mathcal{I}}(H_{m+1} - H_2)PR_i$ pues este valor será una cota superior para el lado derecho de la restricción (14.10), entendiendo por lado derecho, no al termino independiente de la restricción, sino a la expresión a la derecha del signo de desigualdad.

$$\forall i \in \mathcal{I}, \quad \sum_{j \in \mathcal{J}} H_{j+1} D_{ij} \;\geq\; \sum_{j \in \mathcal{J}} H_{j+1} B_{ij} \tag{14.11}$$

La restricción (14.11) establece que el periodo de selección de un recurso para que actúe en el incendio ha de ser anterior al periodo de salida del mismo.

$$\forall i \in \mathcal{I}, \quad \sum_{j \in \mathcal{J}} (H_{m+1} - H_j) B_{ij} - \sum_{j \in \mathcal{J}} (H_{m+1} - H_{j+1}) D_{ij} \;\leq\; TVD_i - TVT_i \tag{14.12}$$

La restricción (14.12) establece que el tiempo de vuelo de una aeronave no puede ser superior al tiempo máximo de vuelo diario. Por tanto, también se ha de contemplar el tiempo que lleva trabajando la aeronave antes de haber sido seleccionada.

$$\forall i \in \mathcal{I}, \quad \sum_{j \in \mathcal{J}} (H_{m+1} - H_j) B_{ij} - \sum_{j \in \mathcal{J}} (H_{m+1} - H_{j+1}) D_{ij} - \sum_{j \in \mathcal{J}} (H_{j+1} - H_j) DES_{ij} \;\geq\; A_i Z_i \tag{14.13}$$

La restricción (14.13) establece que toda aeronave seleccionada para trabajar ha de contribuir en la contención del incendio. De este modo se evita la posibilidad de que una aeronave se seleccione para cubrir el incendio, pero se le indique abandonarlo antes de que ésta llegue al mismo.

$$\forall i \in \mathcal{I},\ \forall j \in \mathcal{J}, \quad DES_{ij} \;\leq\; Y_j \tag{14.14}$$

Por último, la restricción (14.14) establece que sólo se puede descansar en los periodos en los que el incendio no está contenido. Por tanto, ninguna aeronave realizará un descanso en el periodo de contención del incendio, y si una aeronave necesitase descansar en el periodo de la contención, a ésta se le indicará que puede abandonar el incendio en el periodo de iniciar el descanso.

5.2. Función `asignacion_recursos`

Para el cálculo de la selección y asignación óptima de recursos a un incendio forestal, se ha implementado en `R`, la función `asignacion_recursos` (A.3).

Para el manejo correcto de la función implementada, realizaremos una descripción detallada de los argumentos de entrada y de los valores de salida que ésta proporciona.

```
> asignacion_recursos(incendio,recursos,coste_area,frentes=1,InstCont=F,imp=T,aprox=T)
```

Argumentos de entrada:

El primer argumento de entrada, `incendio`, es un `data.frame` que contiene la información estimada de la evolución del incendio forestal. Los elementos de `incendio` contienen la siguiente información:

`incendio$Hj`: vector formado por los periodos de tiempo en los que se realiza la predicción del incendio, tomando como primer valor la hora en la que se ejecuta la función, y los sucesivos las horas correspondientes a las predicciones de los parámetros del incendio. Por ejemplo, si se ejecuta la función a las 9 de la mañana, y se tiene la predicción de la evolución del incendio hasta las 12 de la mañana, en intervalos de media hora, `Hj` será de la forma, `c(9,9.5,10,10.5,11,11.5,12)`.

`incendio$SPj`: vector de la misma longitud que `Hj`, con el perímetro total del incendio en cada uno de los periodos (medido en km).

`incendio$area`: vector de la misma longitud que `Hj`, con el área total del incendio en cada uno de los periodos (medido en hectáreas).

El segundo argumento de entrada, `recursos`, también es un `data.frame` que contiene la información de los recursos disponibles. Los elementos de `recursos` contienen la siguiente información:

`recursos$i`: vector que contiene el índice de cada tipo de recursos.

`recursos$DESi`: vector de la misma longitud que `i`, con una breve descripción o siglas de cada tipo de recurso.

`recursos$NUMi`: vector de la misma longitud que `i`, con el número de recursos de cada tipo.

`recursos$Ai`: vector de la misma longitud que `i`, con el tiempo que tardan en llegar los diferentes recursos al incendio (en horas).

`recursos$Ci`: vector de la misma longitud que `i`, con el coste por hora (variable) que supone emplear cada tipo de recurso (medido en €/hora).

`recursos$PRi`: vector de la misma longitud que `i`, con el rendimiento que tiene cada tipo de recurso (en km/hora).

`recursos$TVi`: vector de la misma longitud que `i`, con el tiempo de vuelo (sin contar descansos) máximo que debe realizar cada tipo de recurso (en horas).

`recursos$TDi`: vector de la misma longitud que `i`, con el tiempo de descanso mínimo que debe realizar cada tipo de recurso (en horas).

`recursos$TVDi`: vector de la misma longitud que `i`, con el tiempo de vuelo diario máximo que debe realizar cada tipo de recurso (en horas).

`recursos$TVTi`: vector de la misma longitud que `i`, con el tiempo de vuelo diario que ha realizado cada recurso (en horas).

`recursos$DESCi`: vector lógico de la misma longitud que `i`, donde se indica si las aeronaves seleccionadas están o no descansadas.

Otros argumentos de entrada son:

`coste_area`: valor numérico que establece el coste que supone la quema de una hectárea de terreno (en €/hectárea).

`frentes`: Opcional. Valor numérico que establece el número mínimo de aeronaves que ha de haber en cada instante en el incendio, hasta su contención. Se supone que como mínimo ha de haber tantos como frentes tenga el incendio. Por defecto, se fija el número mínimo de aeronaves en 1 (frentes=1).

`InstCont`: Opcional. Valor numérico que establece una cota inferior para el tiempo de contención del incendio. Por ejemplo si se quiere contener el incendio antes de las 12:30, se fijaría `InstCont=12+30/60`.

`imp`: Opcional. Parámetro lógico que define el tipo de salida que queremos que nos devuelva la función. Si `imp=T`, la función hará una impresión de pantalla con una breve explicación del resultado óptimo. En caso contrario, `imp=F`, se obtendrán las variables de decisión del modelo matemático.

`aprox`: Opcional. Parámetro lógico que indica si se quiere realizar una aproximación de los tiempos de llegada. En caso de realizarse la aproximación (aprox=T), se aproximan a la alza a su inmediato periodo, estableciéndose los periodos mediante intervalos de 10 minutos. Por ejemplo, si una aeronave tardase 34 minutos en llegar al incendio, el tiempo se aproximaría a 40 minutos. En caso contrario, `aprox=F`, tomará el tiempo real, lo cual puede ocasionar tiempos de compilación muy elevados, pues los periodos podrían ir en intervalos de hasta 1 minuto.

Mediante estos parámetros se pueden establecer situaciones reales, como por ejemplo, considerar una aeronave que lleva operando 3 horas en el incendio, en cuyo caso, se asignará a `Ai` el valor 0, a `TVTi` 3 y a `DESCi` el valor lógico `F`. Otro ejemplo sería considerar una aeronave que está en una base, pero que estuvo operado durante 4 horas en otro incendio; en este caso, se fijaría `TVTi` 4 y se indicaría que la aeronave está descansada, por tanto, `DESCi=T`. También se podría considerar una aeronave que viene de estar operando en otro incendio, en cuyo caso sería la misma situación que la anterior pero se indicaría que la aeronave no ha descansado, `DESCi=F`.

Valores de salida:

Inicialmente la función imprime el proceso de obtención de la solución óptima. Posteriormente, podremos obtener dos tipos de salida, una con `imp=T` u otra con `imp=F`. Si `imp=T`, la salida es en forma de texto y en ella se detalla: el tiempo de resolución del problema; el número de variables y restricciones del problema; en forma de matriz, la selección y asignación adecuada de los recursos para la extinción (se añade una nota a modo de leyenda para interpretarla de forma adecuada), junto con el porcentaje de contención del incendio en cada periodo. Además se muestra el coste (desglosado) que supone la operación de extinción, indicando los gastos por utilización de los recursos y los costes producidos por las hectáreas de terreno afectadas por el incendio.

En caso contrario, si `imp=F`, la función devuelve una lista, `result` con la siguiente información:

`result$Bij`: una matriz binaria con número de filas igual a la longitud de `i` y número de columnas igual a la longitud de `Hj`. En ella se detalla el periodo en el que entra cada uno de los recursos, para realizar una contención óptima.

`result$DESij`: una matriz binaria con número de filas igual a la longitud de `i` y número de columnas igual a la longitud de `Hj`. En ella se detallan los periodos en los que las aeronaves han de realizar los descansos, para realizar una contención óptima.

`result$Dij`: una matriz binaria con número de filas igual a la longitud de `i` y número de columnas igual a la longitud de `Hj`. En ella se detalla hasta que periodo ha de actuar cada uno de los recursos, para realizar una contención óptima.

`result$Yj`: vector binario, de la misma longitud que `Hj`. Si el valor j-ésimo es 1, indica que el incendio no está contenido en el periodo j.

`result$Zi`: vector binario, de la misma longitud que `i`. En él se detallan las aeronaves que se seleccionan para contener el incendio de forma óptima.

5.3. Asignación de recursos en un escenario simulado

Para comprobar el buen funcionamiento de la función, tomaremos el incendio forestal con el que se trabajó en el modelo determinista, Tabla 3.1, y el mismo coste por hectárea de terreno quemado, 100 €/hectárea. Por otro lado, suponemos que se dispone de dos aviones grandes, dos aviones pequeños y dos helicópteros. La información acerca de las características y estado de cada tipo de recurso se recoge en la Tabla 5.1.

Rec.	Descr.	Núm.	Lleg.	Coste	Rend.	TVi	TDi	TVDi	TVTi	DESCi
1	AvionG	2	2	200	0.30	2	40/60	8	0	T
2	AvionP	2	1	150	0.20	2	40/60	8	0	T
3	Helicop	2	1	125	0.15	2	40/60	8	0	T

Tabla 5.1: Parámetros de los recursos.

Con estos datos tendremos información suficiente para ejecutar la función `asignacion_recursos`. Esta información se introducirá en la función implementada en R, de la siguiente forma:

```
> # Tabla crecimiento del fuego:
> (incendio<-data.frame(Hj=seq(1,6),SPj=c(0.3,1,1.3,1.8,2,2.2),
+                       area=c(0.7,5.6,9.6,15.9,20.3,24.3)))
>
> # Tabla de recursos contra el fuego:
> (recursos<-data.frame(i=seq(1,3), DESi=c("AvionG","AvionP","Helicop"),
+                       NUMi=c(2,2,2), Ai=c(2,1,1),
+                       Ci=c(200,150,125), PRi=c(0.3,0.2,0.15),
+                       TVi=rep(2,3), TDi=rep(40/60,3), TVDi=rep(8,3),
+                       TVTi=c(0,0,0), DESCi=c(T,T,T)))
>
> # Coste hectarea
> coste_area<-100
```

Una vez que los datos son leídos y guardados, se introducen en la función `asignacion_recursos`. Inicialmente, tendremos una salida referente al proceso de obtención de la solución óptima. Esta salida la produce la función `gurobi`, que hace uso del solver así llamado y que se emplea para la resolución del problema de programación lineal entera, debido a la rapidez de este frente a otros. A continuación, exponemos la estructura de la salida que nos proporciona la función, fijando como parámetro de salida `imp=T`, para obtener dicha salida gráfica.

```
> asignacion_recursos(incendio,recursos,coste_area)
Optimize a model with 474 rows, 431 columns and 204294 nonzeros
```

```
Presolve removed 127 rows and 100 columns
Presolve time: 0.11s
Presolved: 347 rows, 331 columns, 8979 nonzeros
Variable types: 0 continuous, 331 integer (331 binary)

Root relaxation: objective -1.635249e+03, 196 iterations, 0.01 seconds

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 -1635.2489    0   79          - -1635.2489     -      -    0s
H    0     0                    5350.6962105 -1635.2489   131%     -    0s
.
.
.
 2375502 61661     cutoff   57      4421.00253 4418.48410  0.06%   4.1  370s
 2403642 33521     cutoff   65      4421.00253 4418.48453  0.06%   4.2  375s
 2426786 10423 4420.20867   32    7 4421.00253 4419.61356  0.03%   4.3  380s

Cutting planes:
  Gomory: 8
  Cover: 28
  Clique: 9
  MIR: 13
  GUB cover: 2
  Zero half: 11

Explored 2437161 nodes (10322742 simplex iterations) in 381.95 seconds
Thread count was 4 (of 4 available processors)

Optimal solution found (tolerance 1.00e-04)
Best objective 4.421002533333e+03, best bound 4.420861580804e+03, gap 0.0032%

==============================================================
SOLUCION OPTIMA
=====================================================
Time difference of 6 mins
Numero Variables: 431
Numero Restricciones: 474
-----------------------------------------------------
Seleccion de Recursos:
```

```
(Notacion:0=NoActua,1=Actua,2=Descansa,3=VueloIncendio,4==DescansoVueloIncendio)

            1:00 1:20 1:40 2:00 2:20 2:40 3:00 3:20 3:40 4:00 4:20 4:40 5:00 5:20
AvionG         0    0    0    0    0    0    0    0    0    0    0    0    0    0
AvionG         3    3    3    3    3    3    4    4    1    1    1    1    1    0
AvionP         3    3    3    1    1    1    2    2    1    1    1    1    0    0
AvionP         3    3    3    1    1    1    2    2    1    1    1    1    1    0
Helicop        3    3    3    1    1    1    2    2    1    1    1    1    0    0
Helicop        0    0    3    3    3    1    1    1    0    0    0    0    0    0
PORCENTAJE     0    0    0   18   28   46   50   39   55   70   77   92  100  100

El incendio se contiene entre las 5:00 y las 5:20
-----------------------------------------------------
Coste asignacion optima:  4480
Cost:  2450
NVC:  2030
```

Por tanto, para contener el incendio tendremos que emplear un avión grande (AvionG) desde el inicio (1:00) hasta las 5:20; dos aviones pequeños (AvionP), uno desde la 1:00 hasta las 5:00 y otro desde la 1:00 hasta las 5:20, y 2 helicópteros (Helicop), uno desde la 1:00 hasta las 5:00 y el otro desde la 1:40 hasta las 3:40. Dentro de la matriz, también podemos observar que el avión grande no empezará a realizar operaciones de extinción hasta las 3:40 (después de realizar el viaje de 2 horas de ida hasta el incendio y los 40 minutos de descanso), que tanto los dos aviones pequeños, como un helicóptero, deberán realizar un descanso desde las 3:00 hasta las 3:40 o que el incendio se contendrá entre las 5:00 y las 5:20 (en el momento en el que los aviones dejan de operar y el porcentaje de contención alcanza el 100 %). Además, la función devolverá el coste estimado de la operación, diferenciando entre el coste por la utilización de las aeronaves (consumo de combustible, etc.) y el coste por las hectáreas de terreno quemado.

5.4. Análisis de sensibilidad

En esta sección, observaremos en primer lugar, cómo varían las soluciones que nos devuelve la función `asignacion_recursos` según la disponibilidad de un mayor o menor número de recursos. Posteriormente, cómo varían los tiempos de compilación de la función según el número de recursos disponibles y necesarios para la contención del incendio. Por último, estudiaremos el buen comportamiento de la función ante diferentes situaciones que se pueden producir en la vida real, donde se estudia también el comportamiento del modelo ante la restricción (opcional) que se impone sobre el periodo de contención del incendio, la cuál puede ser de gran importancia si el valor del territorio es elevado, o existe un gran riesgo de que se ocasione alguna situación peligrosa (tener que evacuar una ciudad por la cercanía del incendio, proximidad del incendio a zonas industrializadas, etc.).

Comencemos por estudiar cómo afecta la disponibilidad de un mayor o menor número de recursos a las posibles soluciones del problema.

Escenario	N° Var	N° Restr	Tiempo	Cost	NVC	C+NVC	Recurso	Periodo
Original	431	474	6 min	2450.00	2030	4480.00	1 AvG	1:00-5:20
							1 AvP	1:00-5:00
							1 AvP	1:00-5:20
							1 Heli	1:00-5:00
							1 Heli	1:40-3:40
3 de cada	638	693	16 seg	2608.33	1590	4198.33	1 AvG	1:00-4:20
							3 AvP	1:00-4:20
							2 Heli	1:00-4:20
							1 Heli	2:20-4:00
10 de cada	2087	2226	1 min	2000.00	960	2960.00	5 AvP	1:00-3:00
							2 Heli	1:00-3:00
10 del primero	379	430	3 seg	2933.33	2030	4963.33	2 AvG	1:00-5:40
							2 AvG	1:00-5:00

Tabla 5.2: Resultados cuando se varía el número de recursos.

En la Tabla 5.2 realizamos un análisis de sensibilidad sobre el número de recursos disponibles. En la primera situación disponemos de tres unidades de cada recurso; en este caso, al realizar la comparación con el escenario original, observamos que el gasto por utilización de los recursos aumenta, pero sin embargo, el coste ocasionado por las hectáreas de terreno quemadas disminuye, con lo que el coste total de la operación disminuye como es lógico, debido a que el conjunto de posibles soluciones se ha incrementado. En el siguiente escenario, con diez unidades de cada recurso, pasa algo similar pues vemos como el coste total se reduce drásticamente; esto es debido fundamentalmente a que conseguimos contener el incendio mucho antes.

En el último caso, tenemos únicamente diez unidades disponibles del primer recurso. Ante esta situación, para contener el incendio de forma óptima, necesitaremos emplear 4 aviones grandes desde el instante inicial, dos hasta las 5:40 y otros dos hasta las 5:00.

Otro aspecto importante es la comparación conjunta del tiempo de compilación, y el número de variables y restricciones en cada uno de los casos. El escenario original, pese a ser uno de los que menos variables y restricciones tiene, es el que más tiempo precisa para la obtención de la solución óptima. De hecho, tomando un número concreto de recursos y variando el perímetro del incendio (el perímetro del incendio es el parámetro a partir del cual establecemos la contención o no del incendio), observamos que el tiempo de compilación es mucho mayor en aquellos casos donde el número de recursos necesarios es prácticamente igual al número de recursos disponibles, independientemente de que el número de variables y restricciones sea el mismo. Realizando varios experimentos a partir del escenario considerado en la tabla anterior, 10 recursos de cada tipo, obtenemos:

Escenario	N° Var	N° Restr	Tiempo	N° Rec. Selec.	N° Rec. Total
Original	2087	2226	1 min	7	30
2 veces el perímetro original	2087	2226	34 seg	14	30
3 veces el perímetro original	2087	2226	3 min	12	30
4 veces el perímetro original	2087	2226	2 min	17	30
5 veces el perímetro original	2087	2226	>1 h	-	30
6 veces el perímetro original	2087	2226	>1 h	-	30
7 veces el perímetro original	2087	2226	>1 h	-	30
8 veces el perímetro original	2087	2226	>1 h	-	30
9 veces el perímetro original	2087	2226	>1 h	-	30
10 veces el perímetro original	2087	2226	6 seg	No Solución	30

Tabla 5.3: Variación del perímetro manteniendo fijas 10 unidades de cada de recurso.

Podemos observar que cuando el perímetro del incendio es 5 veces más grande que el del incendio original, el tiempo de compilación aumenta considerablemente. Esto sucede hasta que seleccionamos un perímetro del incendio 10 veces mayor que el original, en este caso, el perímetro es lo suficientemente grande como para que el número de aeronaves necesarias para contener el incendio sea insuficiente.

Supongamos ahora que se detecta un incendio a las 9:00 de la mañana, y se estima que el incendio forestal evoluciona según los parámetros que se recogen en la Tabla 5.4. Además, se dispone de 5 unidades de cada tipo de recurso (avión grande, avión pequeño y helicóptero), que supondremos en diferentes situaciones.

Horas	Perímetro (km)	Área (ha)
9:00	1.0	0.7
9:30	1.3	5.6
10:00	1.8	9.6
10:30	2.0	15.9
11:00	2.2	20.3
11:30	2.3	24.3
12:00	3.1	26.0
12:30	3.4	27.0
13:00	3.7	28.0

Tabla 5.4: Nuevos parámetros del incendio forestal.

A continuación, analizamos distintos casos por separado:

- Suponemos que hay un avión grande y dos pequeños en el incendio, el grande lleva operando 2 horas y le debería tocar realizar un descanso, por otro lado, los aviones pequeños llevan 6 horas trabajadas en el incendio, con lo que deben realizar un descanso en $120\ min - 40\ min = 80\ min$ ($120\ min + 40\ min + 120\ min + 40\ min + 40\ min = 360\ min = 6\ horas$). Por otro lado, se dispone de 4 aviones grandes situados en una base a 80 minutos del incendio (tomando la velocidad media de dicha aeronave), 3 aviones pequeños a 40 minutos, 2 helicópteros a 20 minutos, otros dos a 40 minutos y otro a 80 minutos. Esta situación se recoge en la siguiente tabla:

Rec.	Descr.	Núm.	Lleg.	Coste	Rend.	TVi	TDi	TVDi	TVTi	DESCi
1	AvionG	1	0	200	0.30	2	40/60	8	2	F
1	AvionG	4	80/60	200	0.30	2	40/60	8	0	F
2	AvionP	2	0	150	0.20	2	40/60	8	6	F
2	AvionP	3	40/60	150	0.20	2	40/60	8	0	F
3	Helicop	2	20/60	125	0.15	2	40/60	8	0	F
3	Helicop	2	40/60	125	0.15	2	40/60	8	0	F
3	Helicop	1	80/60	125	0.15	2	40/60	8	0	F

Tabla 5.5: **Caso 1**. Recursos disponibles.

Al introducir los parámetros del incendio, junto con los parámetros referentes a los recursos y el coste por hectárea de terreno quemado obtenemos:

```
==============================================================
SOLUCION OPTIMA
=====================================================
Time difference of 10 secs
Numero Variables: 808
Numero Restricciones: 883
-----------------------------------------------------
Seleccion de Recursos:
(Notacion: 0=NoActua,1=Actua,2=Descansa,3=VueloIncendio,4==DescVueloInc)

             9:00  9:20  9:40 10:00 10:20 10:40 11:00 11:20 11:40 12:00 12:20
AvionG1         2     2     1     1     1     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionP1         1     1     1     0     0     0     0     0     0     0     0
AvionP1         1     1     1     0     0     0     0     0     0     0     0
AvionP2         3     3     1     1     1     0     0     0     0     0     0
```

AvionP2	3	3	1	1	1	0	0	0	0	0	0
AvionP2	3	3	1	1	1	0	0	0	0	0	0
Helicop1	3	1	1	1	1	0	0	0	0	0	0
Helicop1	3	1	1	1	1	0	0	0	0	0	0
Helicop2	3	3	1	1	1	0	0	0	0	0	0
Helicop2	3	3	1	1	1	0	0	0	0	0	0
Helicop3	0	0	0	0	0	0	0	0	0	0	0
PORCENTAJE	13	28	56	83	100	100	100	100	100	100	100

```
El incendio se contiene entre las  10:20  y las  10:40

----------------------------------------------------
Coste asignacion optima:  3673.33
Cost:  2083.33
NVC:  1590
```

Por tanto, observamos que se prioriza la selección de los recursos presentes en el incendio frente a los de las bases. Además, puesto que dichos recursos son insuficientes, se seleccionan los tres aviones pequeños que estaban en la base (AvionP2) y los cuatro helicópteros más cercanos al incendio (Helicop1 y Helicop2).

También se puede observar como el avión grande seleccionado, entra en el incendio realizando un descanso y como las aeronaves seleccionadas de las bases no producen rendimiento hasta no llegar al incendio.

Con la solución obtenida, el incendio se contendrá entre las 10:20 y las 10:40. Esta operación conlleva un coste estimado de 3656.67 €, de los cuales 2066.67 son debidos a costes ocasionados por el uso de las aeronaves.

- Suponemos ahora que hay un avión grande y dos pequeños en el incendio, el grande lleva operando 2 horas y le debería tocar realizar un descanso, por otro lado, los aviones pequeños llevan 6 horas trabajadas en el incendio con lo que deben realizar un descanso en $120\ min - 40\ min = 80\ min$ ($120\ min + 40\ min + 120\ min + 40\ min + 40\ min = 360\ min = 6\ horas$). Por otro lado, se dispone de 4 aviones grandes situados en una base a 80 minutos del incendio (tomando la velocidad media de dicha aeronave), 3 aviones pequeños a 40 minutos, 2 helicópteros a 20 minutos, otros dos a 40 minutos y otro a 80 minutos. Esta situación se recoge en la siguiente tabla:

Rec.	Descr.	Núm.	Lleg.	Coste	Rend.	TVi	TDi	TVDi	TVTi	DESCi
1	AvionG1	1	0	200	0.30	2	40/60	8	2	F
1	AvionG2	4	80/60	200	0.30	2	40/60	8	0	F
2	AvionP1	2	0	150	0.20	2	40/60	8	6	F
2	AvionP2	3	40/60	150	0.20	2	40/60	8	0	F
3	Helicop1	2	20/60	125	0.15	2	40/60	8	0	F
3	Helicop2	2	40/60	125	0.15	2	40/60	8	0	F
3	Helicop3	1	80/60	125	0.15	2	40/60	8	0	F

Tabla 5.6: **Caso 2**. Recursos disponibles.

Al introducir los parámetros del incendio, junto con el coste por hectárea de terreno quemado y los parámetros referentes a los recursos, obtenemos:

```
================================================================
SOLUCION OPTIMA
================================================================
Time difference of 5 secs
Numero Variables: 808
Numero Restricciones: 883
-------------------------------------------------------
Seleccion de Recursos:
(Notacion: 0=NoActua,1=Actua,2=Descansa,3=VueloIncendio,4==DescVueloInc)

             9:00  9:20  9:40 10:00 10:20 10:40 11:00 11:20 11:40 12:00 12:20
AvionG1         2     1     1     1     1     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionG2         0     0     0     0     0     0     0     0     0     0     0
AvionP1         1     1     1     0     0     0     0     0     0     0     0
AvionP1         1     1     1     0     0     0     0     0     0     0     0
AvionP2         3     3     1     1     1     0     0     0     0     0     0
AvionP2         3     3     1     1     1     0     0     0     0     0     0
AvionP2         3     3     1     1     1     0     0     0     0     0     0
Helicop1        0     0     3     1     1     0     0     0     0     0     0
Helicop1        3     1     1     1     1     0     0     0     0     0     0
Helicop2        3     3     1     1     1     0     0     0     0     0     0
Helicop2        3     3     1     1     1     0     0     0     0     0     0
Helicop3        0     0     0     0     0     0     0     0     0     0     0
PORCENTAJE     13    32    56    83   100   100   100   100   100   100   100
```

```
El incendio se contiene entre las  10:20  y las  10:40

----------------------------------------------------
Coste asignacion optima:  3656.67
Cost:  2066.67
NVC:  1590
```

Por tanto, observamos que nuevamente se prioriza la selección de los recursos presentes en el incendio frente a los de las bases. Además, puesto que estos recursos son insuficientes, se seleccionan los tres aviones pequeños que estaban en la base (AvionP2) y los cuatro helicópteros más cercanos al incendio (Helicop1 y Helicop2).

También se puede observar como el avión grande seleccionado, entra en el incendio realizando un descanso y como las aeronaves seleccionadas de las bases no producen rendimiento hasta llegar al incendio.

Con la solución obtenida del modelo de programación lineal, el incendio se contiene entre las 10:20 y las 10:40. Esta operación conllevará un coste estimado de 3656.67 €, de los cuales 2066.67 son debidos a costes ocasionados por el uso de las aeronaves y los restantes producidos por las hectáreas de terreno calcinadas.

- Suponemos que hay un avión grande, dos pequeños y dos helicópteros en el incendio, los cuales llevan 6 horas y 40 minutos trabajando en el incendio. Por otro lado, se dispone de 4 aviones grandes situadas en una base a 1 hora y 20 minutos del incendio; 3 aviones pequeños y un helicóptero a 40 minutos del incendio, que han estado trabajando 2 horas y sus pilotos están descansados, y 2 helicópteros en una base que está a 1 hora y 20 minutos del incendio. Esta situación se recoge en la siguiente tabla:

Rec.	Descr.	Núm.	Lleg.	Coste	Rend.	TVi	TDi	TVDi	TVTi	DESCi
1	AvionG1	1	0	200	0.30	2	40/60	8	6+40/60	F
1	AvionG2	4	80/60	200	0.30	2	40/60	8	0	T
2	AvionP1	2	0	150	0.20	2	40/60	8	6+40/60	F
2	AvionP2	3	40/60	150	0.20	2	40/60	8	2	T
3	Helicop1	2	0	125	0.15	2	40/60	8	6+40/60	F
3	Helicop2	1	40/60	125	0.15	2	40/60	8	2	T
3	Helicop3	2	80/60	125	0.15	2	40/60	8	0	T

Tabla 5.7: **Caso 3**. Recursos disponibles.

Al introducir los parámetros del incendio, junto con el coste por hectárea de terreno quemado y los parámetros referentes a los recursos, obtenemos:

```
==============================================================
SOLUCION OPTIMA
=====================================================
Time difference of 8 secs
Numero Variables: 442
Numero Restricciones: 511
-----------------------------------------------------
Seleccion de Recursos:
(Notacion: 0=NoActua,1=Actua,2=Descansa,3=VueloIncendio,4==DescVueloInc)

            9:00  9:40 10:20 11:00 11:40 12:20 13:00
AvionG1        1     0     0     0     0     0     0
AvionG2        0     0     0     0     0     0     0
AvionG2        3     3     1     0     0     0     0
AvionG2        0     0     0     0     0     0     0
AvionG2        3     3     1     0     0     0     0
AvionP1        1     0     0     0     0     0     0
AvionP1        1     0     0     0     0     0     0
AvionP2        3     1     1     0     0     0     0
AvionP2        3     1     1     0     0     0     0
AvionP2        3     1     1     0     0     0     0
Helicop1       1     0     0     0     0     0     0
Helicop1       1     0     0     0     0     0     0
Helicop2       3     1     1     0     0     0     0
Helicop3       0     0     0     0     0     0     0
Helicop3       0     0     0     0     0     0     0
PORCENTAJE    67    65   100   100   100   100   100

El incendio se contiene entre las  10:20  y las  11:00

-----------------------------------------------------
Coste asignacion optima:  4040
Cost:  2450
NVC:  1590
```

Por tanto, observamos que nuevamente se prioriza la selección de los recursos presentes en el incendio frente a los de las bases. Además, puesto que a estos recursos únicamente les quedan 80 minutos de tiempo de vuelo útil (40 minutos trabajando en el incendio y otros 40 descansando), es necesario que entren 2 de los aviones grandes situados a 80 minutos del incendio y las 4 aeronaves situadas a 40 minutos del incendio (3 aviones pequeños y 1 helicóptero). Con esta selección y asignación de recursos, el incendio se consigue contener entre las 10:20 y las 11:00, ocasionando un coste estimado de 4040 €, de los cuales 2450 son debidos a costes ocasionados por el uso de las aeronaves y los restantes producidos por las hectáreas de terreno quemado.

Por último podemos ver cómo influye la restricción (14.2) (restricción sobre el tiempo de contención del incendio) en la selección de recursos. Supongamos que se produce el incendio inicial, datos de la Tabla 3.1, y que disponemos de 10 recurso de cada uno de los tipos tal y como se recoge en la Tabla 5.1. Estudiemos los diferentes resultados obtenidos mediante la función `asignacion_recursos` dependiendo de las distintas restricciones de tiempo. Los resultados obtenidos se resumen en la siguiente tabla:

Escenario	Tiempo	Cost	NVC	C+NVC	Recurso	Periodo
Sin restr.	2 min	2000.00	960	2960.00	5 AvP	1:00-3:00
					2 Heli	1:00-3:00
InstCont=2	1 seg	-	-	-	-	-
InstCont=2+20/60	5 seg	3133.33	560	3693.33	9 AvP	1:00-2:20
					8 Heli	1:00-2:20
InstCont=2+40/60	17 seg	2458.33	960	3418.33	9 AvP	1:00-2:40
					1 Heli	1:00-2:40
InstCont=3	55 seg	2000.00	960	2960.00	5 AvP	1:00-3:00
					2 Heli	1:00-3:00

Tabla 5.8: Diferentes restricciones sobre el periodo de contención del incendio.

En la Tabla 5.8 observamos cómo afecta restringir de una forma u otra el tiempo de contención del incendio. En primer lugar, se observa cómo al imponer la contención del incendio antes de las 2, la función nos devuelve que no existe solución. Esto es debido a que el incendio se detecta a la 1, y las aeronaves tardan como mínimo una hora en llegar hasta él, por lo que resulta imposible contener el incendio antes de este instante. Si le damos más margen al tiempo de contención, obtenemos soluciones factibles, pero con un coste mayor hasta llegar a la restricción `InstCont=3`. Si se fija un instante de contención igual o superior a las 3:00, entonces la solución que obtenemos será la del modelo sin la restricción en el tiempo de contención, pues se sabe que la solución óptima sin esta restricción contiene el incendio a las 3:00.

Capítulo 6

Conclusiones

A lo largo del presente trabajo se trabajó con diferentes modelos de programación lineal. Inicialmente, se realizó una búsqueda de artículos y libros en la literatura sobre la selección de los recursos necesarios para contener un incendio forestal. Una vez seleccionados los artículos de mayor interés, se comenzó a trabajar con más detalle con ellos, en concreto, los más utilizados fueron el artículo de Donovan y Rideout [6] y posteriormente el trabajo de tesis de Lee [13].

El artículo de Donovan y Rideout [6], no sólo expone un modelo de programación lineal entera, sino que además sirve de base en la tesis de Lee [13]. De este artículo se obtiene la idea de cómo contener un incendio desde un punto de vista matemático, al comparar la línea construida por los recursos seleccionados con el perímetro del incendio.

La tesis de Lee [13] amplia la idea de Donovan y Rideout. En el artículo de Donovan y Rideout, los parámetros del incendio se suponen conocidos en cada instante, pero esta hipótesis puede no ser correcta, pues tanto el perímetro, como el área del incendio en cada periodo de tiempo se obtienen en ocasiones mediante una estimación en la evolución del incendio. Para obtener una asignación válida ante dicha incertidumbre, Lee propone un modelo de programación lineal estocástico. Con este modelo se asegura que cualquiera de las posibles evoluciones que pueda tener el incendio se van a poder contener con la selección de recursos propuesta. Este modelo no aporta ningún nuevo discurso al problema de la selección de recursos con respecto al de Donovan y Rideout, pero sin embargo aborda el problema desde una perspectiva más realista, en cuanto a la obtención de una solución más adecuada.

Pese a los buenos resultados que pueda aportar el planteamiento estocástico del modelo, la resolución del mismo es una tarea muy costosa computacionalmente. Por ese mismo motivo, al disponer de un tiempo limitado para la toma de decisiones en el incendio, se decidió descartar la incorporación de estocasticidad en el modelo.

Con el modelo determinista que incluye las asignaciones horarias, se pretendía dar un enfoque ligeramente distinto al problema. Los principales cambios que se tuvieron en cuenta fueron la obtención no únicamente de la selección óptima, sino también la asignación en cada periodo de tiempo; la posibilidad de ejecutar la función en cualquier instante (sin tener que ser el inicial), lo cual implicaba poder considerar aeronaves que ya estuviesen en el incendio o en otro tipo de situación, y fijar restricciones en el tiempo de uso de cada tipo de recurso.

Al construir el modelo surgieron pequeños problemas, los cuales se solucionaron de forma satisfactoria. En primer lugar se tuvieron que redefinir los periodos de tiempo H_j, pues fue necesario cuadrar los periodos con los tiempos de descanso, tiempos de vuelo hasta el incendio, tiempos de actividad realizada y tiempos de vuelo sin descanso. Una vez redefinidos los periodos, se tuvo que solventar otro gran problema, y es que los tiempos de descanso pueden estar formados por varios periodos H_j, y el modelo no realizaba una correcta compactificación de los mismo, lo cual producía soluciones no deseadas. Para solventar el problema, se introdujo una penalización en el objetivo, de forma que la función objetivo prefiriese tomar los descansos lo antes posible. Una vez solucionados estos dos problemas, que se consideraron los de mayor gravedad, se solventaron otros de menor índole, como la correcta incorporación al modelo de la restricción (14.4), donde no se fija únicamente que el número mínimo de aeronaves a de ser mayor o igual que el número de frentes, sino que además obliga a seleccionar en un instante inicial a las aeronaves que lleguen lo antes posible al incendio.

En resumen, se han estudiado nuevos tipos de modelos y algoritmos de programación matemática, que además se han programado con el fin de aplicarlos a problemas reales haciendo así efectiva la transferencia a la industria de una metodología concreta de optimización.

Además de lo expuesto en el presente trabajo, en el proyecto LUMES, en el que estoy contratado, se están realizando otro tipo de tareas. Algunas ya completadas, como la creación de un algoritmo que proporcione tanto a los pilotos como a los coordinadores aéreos alertas en caso de que exista un posible riesgo de colisión, tanto contra otra aeronave como contra algún obstáculo. El algoritmo se basa en la idea de crear una burbuja, formada por un la mitad de una elipse y la mitad de una esfera, de modo que cuando dos burbujas, o una burbuja y un obstáculo se intersequen, salte una alarme de peligro.

Otro objetivo que se está llevando a cabo, y es complementario al expuesto en este trabajo, es la creación de un modelo de programación lineal entera que en cada periodo de tiempo asigne a cada aeronave una noria (nos referimos por noria, al circuito elíptico que realiza un avión entre el punto de agua que se le asigna y el frente del incendio sobre el cual ha de actuar). Mediante este modelo, se pretende que una vez establecidas las aeronaves en el incendio, se busque la combinación óptima de las aeronaves en las distintas norias para que el rendimiento de éstas sea el mayor posible.

Por último indicar otro tipo de tareas que atañen al proyecto, como es la identificación de zonas seguras para los brigadistas y cálculo de la ruta más corta (mínimo tiempo), donde diferenciamos entre rutas de escape o avituallamiento de las brigadas y rutas de escape civil (por dónde evacuar a una población en caso de que se encuentre en peligro), o la obtención del número máximo de aeronaves por noria.

Con todos estos objetivos, el proyecto LUMES pretende realizar extinciones de incendios forestales más eficiente, optimizando la utilización de sus recursos, para realizar extinciones más efectivas en un menor periodo de tiempo. Y lo que es más importante, aumentar las medidas de seguridad en el ataque a un incendio forestal, tanto en las brigadas o núcleos urbanos en peligro, con el diseño de rutas de escape seguras, como advirtiendo a las aeronaves de alguna situación de peligro, como puede ser una posible colisión entre dos aeronaves, o la proximidad de un obstáculo peligroso, muchos de los cuales no se pueden detectar a simple vista, como las redes eléctricas.

Apéndice A

Implementación en R de los algoritmos

A.1. Modelo determinista

```
seleccion_recursos<-function(incendio,recursos,coste_area,B1=F,B2=F,imp=T){
  #----------------------------------------------------------------------
  # Se instalan los paquetes faltantes y se cargan

  paquetes.instalados<-installed.packages()[,1] # paquetes instalados

  #comprobamos si los paquetes estan instalados. En caso de no estar se instalan.
  if(sum("lpSolveAPI"==paquetes.instalados)==0 |
     sum("R.utils"==paquetes.instalados)==0){
    cat("-----------------------------------------------------\n")
    if(sum("lpSolveAPI"==paquetes.instalados)==0){
      cat("Instalacion de la libreria: lpSolveAPI\n")
      install.packages("lpSolveAPI",repos="http://cran.es.r-project.org")
    }
    if(sum("R.utils"==paquetes.instalados)==0){
      cat("Instalacion de la libreria: R.utils\n")
      install.packages("R.utils",repos="http://cran.es.r-project.org")
    }
    cat("-----------------------------------------------------\n")
  }
```

```
# Se cargan
library(lpSolveAPI)
library(R.utils)
#----------------------------------------------------------------------

time<-proc.time()   #Fijamos tiempo inicial

Hj<-incendio$Hj # periodo j
SPj<-incendio$SPj # perimetro del fuego en j
Area<-incendio$area # area del incendio en j
IndRec<-recursos$i # indices de los recursos
Ai<-recursos$Ai # tiempo de vuelo hasta el incendio
Ci<-recursos$Ci # coste de alquiler por hora
Pi<-recursos$Pi # coste de alquiler inicial
PRi<-recursos$PRi # produccion del recurso i por hora
NUMi<-recursos$NUMi # numero de recursos del tipo i
Orden<-order(recursos$DESi)
DESi<-c()
for(i in 1:length(Orden))DESi[i]<-levels(recursos$DESi)[which(Orden==i)]


# Si hay 0 unidades de algun recurso se elimina
if(sum(NUMi==0)!=0){
  NO_r<-which(NUMi==0)
  IndRec<-IndRec[-NO_r]
  Ai<-Ai[-NO_r]
  Ci<-Ci[-NO_r]
  Pi<-Pi[-NO_r]
  PRi<-PRi[-NO_r]
  NUMi<-NUMi[-NO_r]
  DESi<-DESi[-NO_r]
}

# creamos cada recurso (de los NUMi) como una variable binaria
for(i in length(DESi):1){
  NUM<-NUMi[i]
  while(NUM>1){
    IndRec<-insert(IndRec,i,IndRec[i])
    Ai<-insert(Ai,i,Ai[i])
    Ci<-insert(Ci,i,Ci[i])
    Pi<-insert(Pi,i,Pi[i])
```

```
    PRi<-insert(PRi,i,PRi[i])
    DESi<-insert(DESi,i,DESi[i])
    NUM<-NUM-1
  }
}


# Orden de las variables de decision:
# Dij Orden de los Dij (D11,D12,...,D21,D22,...)
# Lj
# Yj
# Nj
# Zi

n<-length(DESi) # numero de recursos
I<-1:n # conjunto de indices para los recursos
m<-length(Hj) # numero de periodos
J<-1:m # conjunto de indices para los periodos
n_var<-(n*m)+(m)+(m)+(m)+(n) # numero de variables
n_restr<-1+n+m+m+m+2 # numero de restricciones

M<-max(SPj) # Valor de M, (restriccion 3)

# Incremente del perimetro del incendio en el periodo j
PERj<-numeric(m)
PERj[1]<-SPj[1]
for(j in 2:m){
  PERj[j]<-SPj[j]-SPj[j-1]
}

# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#------------------------------------------------
# Si no se fijan restricciones en el presupuesto:
if (B1==F){
```

```
  B1<-sum(Pi)
}
if (B2==F){
  B2<-sum(Ci*max(Hj))
}

#-------------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
# Coeficientes segundas restricciones:
restr2<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes terceras restricciones:
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
restr4<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes quintas restricciones:
restr5<-matrix(0,nrow=m,ncol=n_var)
#----------------------------------------------------------------------------
# Coeficientes sexta restriccion:
restr6<-numeric(n_var)
# Coeficientes septima restriccion:
restr7<-numeric(n_var)
#----------------------------------------------------------------------------
# Funcion objetivo:
obj<-numeric(n_var)

for(i in I){
  restr2[i,(i-1)*m+(1:m)]<-1 # Coeficientes Dij
  restr2[i,(n*m)+(m)+(m)+(m)+i]<--1 # Coeficientes Zi

  for(j in J){
    restr1[j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # Coeficientes Dij
    restr1[(n*m)+(m)+(m)+j]<--PERj[j] # Coeficientes Nj

    restr3[j,(n*m)+j]<--1 # Coeficientes Lj
    restr3[j,(n*m)+(m)+j]<--M # Coeficientes Yj
    restr3[j,(n*m)+(m)+(m)+j]<-SPj[j] # Coeficientes Nj

    restr4[j,(i-1)*m+j]<-(Hj[j]-Ai[i])*PRi[i] # Coeficientes Dij
    restr4[j,m*n+j]<--1 # Coeficientes Lj
```

```
    restr5[j,(n*m)+(m)+(m)+j]<-1 # Coeficientes Nj
    if(j!=1){
      restr5[j,(n*m)+(m)+j-1]<--1 # Coeficientes Yj-1
    }

    #-----------------------------------------------------------------------
    restr6[(n*m)+(m)+(m)+(m)+i]<-Pi[i] # Coeficientes Zi

    restr7[j+(i-1)*m]<-Ci[i]*Hj[j] # Coeficientes Dij
    #-----------------------------------------------------------------------

    obj[j+(i-1)*m]<-Ci[i]*Hj[j] # Coeficientes Dij
    obj[(n*m)+(m)+(m)+(m)+i]<-Pi[i] # Coeficientes Zi
    obj[(n*m)+(m)+(m)+j]<-NVCj[j] # Coeficientes Nj
  }
}

################################################################################
# El modelo

nrecursos<-make.lp(n_restr,n_var)

set.row(nrecursos,1,restr1)
for(i in I)     set.row(nrecursos,1+i,restr2[i,])
for(j in J)     set.row(nrecursos,1+n+j,restr3[j,])
for(j in J)     set.row(nrecursos,1+n+m+j,restr4[j,])
for(j in J)     set.row(nrecursos,1+n+2*m+j,restr5[j,])
#-----------------------------------------------------------------------
set.row(nrecursos,2+n+3*m,restr6)
set.row(nrecursos,3+n+3*m,restr7)
#-----------------------------------------------------------------------


set.objfn(nrecursos,obj)
set.rhs(nrecursos,c(rep(0,1+n),rep(0,2*m),1,rep(0,m-1),B1,B2))
set.constr.type(nrecursos,c(">=",rep("<=",n+m),rep("=",2*m),rep("<=",2)))
set.type(nrecursos, c(1:(n*m),((n*m)+(m)+1):((n*m)+(m)+(m)+(m)+(n))), "binary")
set.type(nrecursos, (n*m+1):((n*m)+(m)), "real")

if(imp==T) cat("Iniciando resolucion del problema\n")
```

```
resolver<-solve(nrecursos)
if(imp==T) cat("Resolucion del problema terminada\n")

if(resolver!=0){
  if(imp==T) cat("Error al resolver el problema")
}else{
  # Valores optimos de las variables:
  Dij<-matrix(get.variables(nrecursos)[1:(n*m)],byrow=T,nrow=n,ncol=m)
  Lj<-get.variables(nrecursos)[(n*m+1):(n*m+m)]
  Yj<-get.variables(nrecursos)[(n*m+m+1):(n*m+m+m)]
  Nj<-get.variables(nrecursos)[(n*m+m+m+1):(n_var-n)]
  Zi<-get.variables(nrecursos)[(n_var-n+1):(n_var)]

  dij<-c()
  desi<-c()
  indi<-c()
  INDICE<-IndRec
  while(length(INDICE)!=0){
    dij<-rbind(dij,colSums(matrix(Dij[which(INDICE[1]==IndRec),],ncol=m)))
    desi<-c(desi,DESi[which(INDICE[1]==IndRec)][1])
    indi<-c(indi,INDICE[1])
    INDICE<-INDICE[-which(INDICE[1]==INDICE)]
  }

  # El incendio se controla en el periodo:
  if(sum(Yj==0)!=0){
    contr<-min(which(Yj==0))
  }
  if(sum(Yj==0)==0){
    if(sum(Lj)>sum(PERj)){
      contr<-length(Yj)
    }
    if(sum(Lj)<sum(PERj)){
      contr<-FALSE
    }
  }

  # Funcion objetivo (total y desglosada):
  # Cost:
  cost<-0
  for(i in I){
```

```
  for(j in J){
    cost<-cost+Ci[i]*Hj[j]*Dij[i,j]
  }
}

# Pre:
pre<-0
for(i in 1:n){
  pre<-pre+Pi[i]*Zi[i]
}

# NVC:
NVC<-0
for(j in 1:m){
  NVC<-NVC+NVCj[j]*Nj[j]
}

finaltime<-(proc.time()-time)[1]/60

if(imp==T){
  cat(paste("Tiempo compilacion:",finaltime,"\n"))
  cat(paste("Seleccion de Recursos:\n"))
  Indice=Descripcion=Cantidad=Periodo<-c()
  for(i in 1:sum(dij!=0)){
    Indice<-c(Indice,indi[which(dij!=0,arr.ind=T)[,1][i]])
    Descripcion<-c(Descripcion,desi[which(dij!=0,arr.ind=T)[,1][i]])
    Cantidad<-c(Cantidad,dij[which(dij!=0)][i])
    Periodo<-c(Periodo,Hj[which(dij!=0,arr.ind=T)[,2][i]])
  }
  print((data.frame("Indice"=Indice,
                    "Descripcion"=Descripcion,
                    "Cantidad"=Cantidad,
                    "Periodo"=Periodo)))
  cat(paste("Perimetro cubierto: ", sum(Lj),"\n"))
  cat(paste("El incendio se contiene en el periodo: ",contr,"\n"))
  cat("---------------------------------------------------\n")
  cat(paste("Coste asignacion optima: ", get.objective(nrecursos),"\n"))
  cat(paste("Pre:", pre,"\n"))
  cat(paste("Cost:", cost,"\n"))
  cat(paste("NVC:", NVC),"\n")
}else{
```

```
      result<-list(Dij=dij,Yj=Yj,Zi=rowSums(dij))
      return(result)
    }
  }
}
```

A.2. Modelo estocástico

A.2.1. Cortes de factibilidad

```
Feasible_L<-function(x0,incendiow,Ai,Ci,Pi,PRi,coste_area,B2){

  Hj<-incendiow$Hj # periodo j
  SPj<-incendiow$SPj # perimetro del incendio en j
  Area<-incendiow$area # area del incendio en j

  # Orden de las variables de decision:
  # variables de holgura
  # Dij: (D11,D12,...,D21,D22,...)
  # Yj
  Zi<-x0 # Si se emplea un recurso i

  n<-length(Ai) # numero de recursos
  I<-1:n # conjunto de indices para los recursos
  m<-length(Hj) # numero de periodos
  J<-1:m # conjunto de indices para los periodos
  n_restr<-1+1+m+n # numero de restricciones
  n_var<-2*n_restr+(n*m)+(m) # numero de variables

  M<-max(SPj) # Valor de M (para la restricción 3)

  # Incremento del perimetro del incendio en el periodo j
  PERj<-numeric(m)
  PERj[1]<-SPj[1]
  for(j in 2:m){
    PERj[j]<-SPj[j]-SPj[j-1]
  }

  # Incremento del area del incendio en el periodo j
```

```
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#-----------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
restr1[1]<-1
restr1[2]<--1
# Coeficientes de la segunda restriccion:
restr2<-numeric(n_var)
restr2[3]<-1
restr2[4]<--1
# Coeficientes terceras restricciones:
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
restr4<-matrix(0,nrow=n,ncol=n_var)



# Funcion objetivo:
obj<-numeric(n_var)
obj[1:(2*n_restr)]<-1

for(i in I){
  restr4[i,2*n_restr+(i-1)*m+(1:m)]<-1 # Coeficientes Dij
  restr4[i,2*(2+m+i)-1]<-1 # holguras
  restr4[i,2*(2+m+i)]<--1 # holguras

  for(j in J){
    restr2[2*n_restr+j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # Coeficientes Dij
    if(j>1){
      restr2[2*n_restr+(n*m)+j-1]<--PERj[j] # Coeficientes Yj-1
    }
    restr3[j,2*n_restr+(i-1)*m+j]<--(Hj[j]-Ai[i])*PRi[i] # Coeficientes Dij
    restr3[j,2*n_restr+(n*m)+j]<--M # Coeficientes Yj
    if(j>1){
      restr3[j,2*n_restr+(n*m)+j-1]<-SPj[j] # Coeficientes Yj-1
```

```
      }
      restr3[j,2*(2+j)-1]<-1 # holguras
      restr3[j,2*(2+j)]<--1 # holguras

      restr1[2*n_restr+j+(i-1)*m]<-Ci[i]*Hj[j] # Coeficientes Dij
    }
  }


  ################################################################################
  # El modelo

  F_L<-make.lp(n_restr,n_var)

  set.row(F_L,1,restr1)
  set.row(F_L,2,restr2)
  for(j in J)     set.row(F_L,2+j,restr3[j,])
  for(i in I)     set.row(F_L,2+m+i,restr4[i,])

  set.objfn(F_L,obj)
  set.rhs(F_L,c(B2,PERj[1],-SPj[1],rep(0,m-1),Zi))
  set.constr.type(F_L,c("<=",">=",rep("<=",m),rep("<=",n)))
  set.type(F_L, 1:(2*n_restr), "real") # v+, v-
  set.type(F_L, (2*n_restr+1):(2*n_restr+(n*m)), "real") # Dij
  set.type(F_L, (2*n_restr+(n*m)+1):(2*n_restr+(n*m)+(m)), "real") # Yj

  resolver<-solve(F_L)

  Tec<-matrix(c(rep(0,n),rep(0,n),rep(0,n*m),-diag(n)),ncol=n,nrow=n_restr,byrow=T)

  return(list(resolver=resolver,dual=get.dual.solution(F_L)[1+1:n_restr],
            rhs=get.rhs(F_L),Tec=Tec,Sol=get.objective(F_L)))
}
```

A.2.2. Cortes de optimalidad

```
Optimal_L<-function(x0,incendiow,Ai,Ci,Pi,PRi,coste_area,B2){

  Hj<-incendiow$Hj # periodo j
  SPj<-incendiow$SPj # perimetro del incendio en j
```

```
Area<-incendiow$area # area del incendio en j
IndRec<-recursos$i # indices de los recursos

# Orden de las variables de decision:
# Dij: (D11,D12,...,D21,D22,...)
# Yj
Zi<-x0

n<-length(Ai) # numero de recursos
I<-1:n # conjunto de indices para los recursos
m<-length(Hj) # numero de periodos
J<-1:m # conjunto de indices para los periodos
n_var<-(n*m)+(m) # numero de variables
n_restr<-1+1+m+n  #numero de restricciones

M<-max(SPj) # valor de M (restriccion 3)

# Incremento del perimetro del incendio en el periodo j
PERj<-numeric(m)
PERj[1]<-SPj[1]
for(j in 2:m){
  PERj[j]<-SPj[j]-SPj[j-1]
}

# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#------------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
# Coeficientes segunda restriccion:
restr2<-numeric(n_var)
# Coeficientes terceras restricciones:
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
```

```
restr4<-matrix(0,nrow=n,ncol=n_var)

# Funcion objetivo:
obj<-numeric(n_var)

for(i in I){
  restr4[i,(i-1)*m+(1:m)]<-1 # coeficientes Dij

  for(j in J){
    restr1[j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij
    #-------------------------------------------------------------------------
    restr2[j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    if(j>1){
      restr2[(n*m)+j-1]<--PERj[j] # coeficientes Yj-1
    }
    #-------------------------------------------------------------------------
    restr3[j,(i-1)*m+j]<--(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    restr3[j,(n*m)+j]<--M # Yj
    if(j>1){
      restr3[j,(n*m)+j-1]<-SPj[j] # coeficientes Yj-1
    }
    #-------------------------------------------------------------------------

    obj[j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij
    if(j>1){
      obj[(n*m)+j-1]<-NVCj[j] # coeficientes Yj-1
    }
  }
}

###############################################################################
# El modelo

O_L<-make.lp(n_restr,n_var)

set.row(O_L,1,restr1)
set.row(O_L,2,restr2)
for(j in J)     set.row(O_L,2+j,restr3[j,])
for(i in I)     set.row(O_L,2+m+i,restr4[i,])

set.objfn(O_L,obj)
```

```
  set.rhs(O_L,c(B2,PERj[1],-SPj[1],rep(0,m-1),Zi))
  set.constr.type(O_L,c("<=",">=",rep("<=",m+n)))
  set.type(O_L, 1:(n*m), "real") #Dij
  set.type(O_L, (n*m+1):n_var, "real") # Yj

  resolver<-solve(O_L)

  Tec<-matrix(c(rep(0,n),rep(0,n),rep(0,n*m),-diag(n)),ncol=n,nrow=n_restr,byrow=T)

  return(list(resolver=resolver,dual=get.dual.solution(O_L)[1+1:n_restr],
              rhs=get.rhs(O_L),Tec=Tec,Sol=get.objective(O_L)+NVCj[1],
              Var=get.variables(O_L)))
}
```

A.2.3. Método L-shaped

```
L_Shaped<-function(incendio,Pw,recursos,coste_area,B1,B2){

  # Datos de los recursos:
  IndRec<-recursos$i # indices de los recursos
  Ai<-recursos$Ai # tiempo de vuelo hasta el incendio
  Ci<-recursos$Ci # coste de alquiler por hora
  Pi<-recursos$Pi # coste de alquiler inicial
  PRi<-recursos$PRi # produccion del recurso i por hora
  NUMi<-recursos$NUMi # numero de recursos del tipo i
  Orden<-order(recursos$DESi)
  DESi<-c()
  for(i in 1:length(Orden))DESi[i]<-levels(recursos$DESi)[which(Orden==i)]

  # Si algun recurso tiene NUMi=0 se elimina
  if(sum(NUMi==0)!=0){
    NO_r<-which(NUMi==0)
    IndRec<-IndRec[-NO_r]
    Ai<-Ai[-NO_r]
    Ci<-Ci[-NO_r]
    Pi<-Pi[-NO_r]
    PRi<-PRi[-NO_r]
    NUMi<-NUMi[-NO_r]
    DESi<-DESi[-NO_r]
  }
```

```
# creamos cada recurso (de los NUMi) como una variable binaria
  for(i in length(DESi):1){
    NUM<-NUMi[i]
    while(NUM>1){
      IndRec<-insert(IndRec,i,IndRec[i])
      Ai<-insert(Ai,i,Ai[i])
      Ci<-insert(Ci,i,Ci[i])
      Pi<-insert(Pi,i,Pi[i])
      PRi<-insert(PRi,i,PRi[i])
      DESi<-insert(DESi,i,DESi[i])
      NUM<-NUM-1
    }
  }

  n<-length(DESi) # numero de recursos
  m<-length(incendio[[1]]$Hj) # numero de periodos

  #---------------------------------------------------------------------------
  # Step 0: Inicializacion

  # Fijamos LB, UB, eps y fijamos k (iterante):
  r=s<-0 # Contadores numero de restricciones de factibilidad y optimalidad
  v<-0 # Contador numero de iteraciones
  theta<--Inf
  lambda<-Inf
  eps<-0.001
  #--------------------------------------------------------------------------

  # Definimos variables:
  xv<-list() # Variables Zi
  K<-length(incendio) # Numero de escenarios

  # Coeficientes para Feasibility Cuts:
  Dl<-c()
  dl<-c()
  # Coeficientes para Optimality Cuts:
  El<-c()
  el<-c()
  # Restricciones del Problema Master:
  A<-c(Pi,0)
```

```
b<-B1


while(lambda-theta>eps){
  #--------------------------------------------------------------------------
  # Paso 1: Resolvemos Problema Master.

  v<-v+1 # contador numero iteraciones

  Master<-make.lp(1+r+s,length(A))

  set.row(Master,1,A)
  if(r>0)  for(l in 1:r) set.row(Master,1+l,c(Dl[l,],0))
  if(s>0)  for(l in 1:s) set.row(Master,1+r+l,c(El[l,],1))

  set.objfn(Master,c(Pi,1))
  set.rhs(Master,c(b,dl,el))
  set.constr.type(Master,c("<=",rep(">=",r+s)))
  set.type(Master, 1:(length(A)-1), "real") # Zi
  set.type(Master, length(A), "real") # Mu (holgura)

  solve(Master)

  Master_Obj<-get.objective(Master)
  xv[[v]]<-get.variables(Master)[1:n]
  theta<-get.variables(Master)[n+1]


  #-------------------------------------------------------------------------
  # Paso 2: miramos si xv pertenece a la region factible de los subproblemas

  subprob<-list()
  Subprob_Op<-list()
  feasible<-c()

  # Subproblemas:
  for(k in 1:K){
    subprob[[k]]<-Feasible_L(xv[[v]],incendio[[k]],Ai,Ci,Pi,PRi,coste_area,B2)
    feasible[k]<-subprob[[k]]$Sol # indica si algun subproblema es infactible
  }
  feasible
```

```
    if(sum(feasible!=0)!=0){ # Si hay algun escenario infactible

      # Tomamos todos los escenarios infactibles para realizar los cortes
      infeasible<-which(feasible!=0) # Escenarios infactibles para xv

      for(k in infeasible){
        r<-r+1
        dl[r]<-subprob[[k]]$dual%*%subprob[[k]]$rhs
        Dl<-rbind(Dl,subprob[[k]]$dual%*%subprob[[k]]$Tec)
      }

      #---------------------------------------------------------------------------
      # Paso 3: Creamos los cortes de optimalidad en caso de ser necesarios

    }else{ # Si todos los escenarios son factibles

      s<-s+1
      Els<-0
      els<-0
      for(k in 1:K){
        Subprob_Op[[k]]<-Optimal_L(xv[[v]],incendio[[k]],Ai,Ci,Pi,PRi,coste_area,B2)
        Els<-Els+Pw[k]*Subprob_Op[[k]]$dual%*%Subprob_Op[[k]]$Tec
        els<-els+Pw[k]*Subprob_Op[[k]]$dual%*%Subprob_Op[[k]]$rhs
      }
      El<-rbind(El,Els)
      el[s]<-els

      lambda<-el[s]-El[s,]%*%xv[[v]]
    }
  }
  return(theta+Pi%*%xv[[v]])
}
```

A.2.4. Cortes para obtener una solución inicial aceptable

```
Restr_Master<-function(incendiow,Ai,Ci,Pi,PRi,coste_area,B1,B2){
  Hj<-incendiow$Hj # periodo j
  SPj<-incendiow$SPj # perimetro del incendio en j
```

```
Area<-incendiow$area # area del incendio en j
IndRec<-recursos$i # indices de los recursos

# Orden de las variables de decision:
# Dij: (D11,D12,...,D21,D22,...)
# Yj
# Zi

n<-length(Ai) # numero de recursos
I<-1:n # conjunto de indices para los recursos
m<-length(Hj) # numero de periodos
J<-1:m # conjunto de indices para los periodos
n_var<-(n*m)+(m)+(n) # numero de variables
n_restr<-1+1+1+m+n # numero de restricciones

M<-max(SPj) # valor de M (restriccion 3)

# Incremento del perimetro del incendio en el periodo j
PERj<-numeric(m)
PERj[1]<-SPj[1]
for(j in 2:m){
  PERj[j]<-SPj[j]-SPj[j-1]
}

# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#-------------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
# Coeficientes de la segunda restriccion:
restr2<-numeric(n_var)
# Coeficientes terceras restricciones:
restr3<-numeric(n_var)
# Coeficientes cuartas restricciones:
```

```
restr4<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes quintas restricciones:
restr5<-matrix(0,nrow=n,ncol=n_var)

# Funcion objetivo:
obj<-numeric(n_var)

for(i in I){
  restr5[i,(i-1)*m+(1:m)]<-1 # coeficientes Dij
  restr5[i,(n*m)+m+i]<--1 # coeficientes Zi

  obj[(m*n)+m+i]<-1 # coeficientes Zi
  for(j in J){
    restr1[(m*n)+m+i]<-Pi[i] # coeficientes Zi
    #---------------------------------------------------------------------------

    restr2[j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij
    #---------------------------------------------------------------------------

    restr3[j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    if(j>1){
      restr3[(n*m)+j-1]<--PERj[j] # coeficientes Yj-1
    }
    #---------------------------------------------------------------------------

    restr4[j,(i-1)*m+j]<--(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    restr4[j,(n*m)+j]<--M # coeficientes Yj
    if(j>1){
      restr4[j,(n*m)+j-1]<-SPj[j] # coeficientes Yj-1
    }
    #---------------------------------------------------------------------------
  }
}
##########################################################
# El modelo

O_L<-make.lp(n_restr,n_var)

set.row(O_L,1,restr1)
set.row(O_L,2,restr2)
set.row(O_L,3,restr3)
```

```
  for(j in J)     set.row(O_L,3+j,restr4[j,])
  for(i in I)     set.row(O_L,3+m+i,restr5[i,])

  set.objfn(O_L,obj)
  set.rhs(O_L,c(B1,B2,PERj[1],-SPj[1],rep(0,m-1+n)))
  set.constr.type(O_L,c(rep("<=",2),">=",rep("<=",m+n)))
  set.type(O_L, 1:n_var, "binary") #Dij, Yj, Zi

  resolver<-solve(O_L)
  return(list(get.objective(O_L),resolver))
}
```

A.2.5. Cortes de factibilidad L^2

```
Feasible_L2<-function(x0,incendiow,Ai,Ci,Pi,PRi,coste_area,B2){

  Hj<-incendiow$Hj # periodo j
  SPj<-incendiow$SPj # perimetro del incendio en j
  Area<-incendiow$area # area del incendio en j

  # Orden de las variables de decision:
  # Dij: (D11,D12,...,D21,D22,...)
  # Yj
  Zi<-x0

  n<-length(Ai) # numero de recursos
  I<-1:n # conjunto de indices para los recursos
  m<-length(Hj) # numero de periodos
  J<-1:m # conjunto de indices para los periodos
  n_restr<-1+1+m+n # numero de restricciones
  n_var<-2*n_restr+(n*m)+(m) # numero de variables

  M<-max(SPj) # valor de M (restriccion 3)

  # Incremento del perimetro del incendio en el periodo j
  PERj<-numeric(m)
  PERj[1]<-SPj[1]
  for(j in 2:m){
    PERj[j]<-SPj[j]-SPj[j-1]
  }
```

```
# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#-------------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
restr1[1]<-1 # holguras
restr1[2]<--1 # holguras
# Coeficientes de la segunda restriccion:
restr2<-numeric(n_var)
restr2[3]<-1 # holguras
restr2[4]<--1 # holguras
# Coeficientes terceras restricciones:
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
restr4<-matrix(0,nrow=n,ncol=n_var)


# Funcion objetivo:
obj<-numeric(n_var)
obj[1:(2*n_restr)]<-1

for(i in I){
  restr4[i,2*n_restr+(i-1)*m+(1:m)]<-1 # coeficientes Dij
  restr4[i,2*(2+m+i)-1]<-1 # holguras
  restr4[i,2*(2+m+i)]<--1 # holguras

  for(j in J){
    restr2[2*n_restr+j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    if(j>1){
      restr2[2*n_restr+(n*m)+j-1]<--PERj[j] # coeficientes Yj-1
    }
    restr3[j,2*n_restr+(i-1)*m+j]<--(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    restr3[j,2*n_restr+(n*m)+j]<--M # coeficientes Yj
```

```
      if(j>1){
        restr3[j,2*n_restr+(n*m)+j-1]<-SPj[j] # coeficientes Yj-1
      }
      restr3[j,2*(2+j)-1]<-1 # holguras
      restr3[j,2*(2+j)]<--1 # holguras

      restr1[2*n_restr+j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij
    }
  }


  ##################################################################################
  # El modelo

  F_L2<-make.lp(n_restr,n_var)

  set.row(F_L2,1,restr1)
  set.row(F_L2,2,restr2)
  for(j in J)     set.row(F_L2,2+j,restr3[j,])
  for(i in I)     set.row(F_L2,2+m+i,restr4[i,])

  set.objfn(F_L2,obj)
  set.rhs(F_L2,c(B2,PERj[1],-SPj[1],rep(0,m-1),Zi))
  set.constr.type(F_L2,c("<=",">=",rep("<=",m),rep("<=",n)))
  set.type(F_L2, 1:(2*n_restr), "real") # v+, v-
  set.type(F_L2, (2*n_restr+1):(2*n_restr+(n*m)+(m)), "binary") # Dij, Yj

  resolver<-solve(F_L2)

  Tec<-matrix(c(rep(0,n),rep(0,n),rep(0,n*m),-diag(n)),ncol=n,nrow=n_restr,byrow=T)

  return(list(resolver=resolver,dual=get.dual.solution(F_L2)[1+1:n_restr],
              rhs=get.rhs(F_L2),Tec=Tec,Sol=get.objective(F_L2),
              Var=get.variables(F_L2)))
}
```

A.2.6. Cortes de optimalidad L^2

```
Optimal_L2<-function(x0,incendiow,Ai,Ci,Pi,PRi,coste_area,B2){
```

```
Hj<-incendiow$Hj # periodo j
SPj<-incendiow$SPj # perimetro del incendio en j
Area<-incendiow$area # area del incendio en j
IndRec<-recursos$i # indices de los recursos

# Orden de las variables de decision:
# Dij: (D11,D12,...,D21,D22,...)
# Yj
Zi<-x0

n<-length(Ai) # numero de recursos
I<-1:n # conjunto de indices para los recursos
m<-length(Hj) # numero de periodos
J<-1:m # conjunto de indices para los periodos
n_var<-(n*m)+(m) # numero de variables
n_restr<-1+1+m+n # numero de restricciones

M<-max(SPj) # valor de M (restriccion 3)

# Incremento del perimetro del incendio en el periodo j
PERj<-numeric(m)
PERj[1]<-SPj[1]
for(j in 2:m){
  PERj[j]<-SPj[j]-SPj[j-1]
}

# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#------------------------------------------------

# Coeficientes de la primera restriccion:
restr1<-numeric(n_var)
# Coeficientes de la segunda restriccion:
restr2<-numeric(n_var)
# Coeficientes terceras restricciones:
```

```
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
restr4<-matrix(0,nrow=n,ncol=n_var)

# Funcion objetivo:
obj<-numeric(n_var)
for(i in I){
  restr4[i,(i-1)*m+(1:m)]<-1 # coeficientes Dij

  for(j in J){

    restr1[j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij

    restr2[j+(i-1)*m]<-(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    if(j>1){
      restr2[(n*m)+j-1]<--PERj[j] # coeficientes Yj-1

    }

    restr3[j,(i-1)*m+j]<--(Hj[j]-Ai[i])*PRi[i] # coeficientes Dij
    restr3[j,(n*m)+j]<--M # Yj
    if(j>1){
      restr3[j,(n*m)+j-1]<-SPj[j] # coeficientes Yj-1
    }

    obj[j+(i-1)*m]<-Ci[i]*Hj[j] # coeficientes Dij
    if(j>1){
      obj[(n*m)+j-1]<-NVCj[j] # coeficientes Yj-1
    }
  }
}

##############################################################################
# El modelo

O_L2<-make.lp(n_restr,n_var)

set.row(O_L2,1,restr1)
set.row(O_L2,2,restr2)
for(j in J)     set.row(O_L2,2+j,restr3[j,])
for(i in I)     set.row(O_L2,2+m+i,restr4[i,])
```

```
  set.objfn(O_L2,obj)
  set.rhs(O_L2,c(B2,PERj[1],-SPj[1],rep(0,m-1),Zi))
  set.constr.type(O_L2,c("<=",">=",rep("<=",m+n)))
  set.type(O_L2, 1:n_var, "binary") #Dij, Yj

  resolver<-solve(O_L2)

  Tec<-matrix(c(rep(0,n),rep(0,n),rep(0,n*m),-diag(n)),
              ncol=n,nrow=n_restr,byrow=T)

  return(list(resolver=resolver,Sol=get.objective(O_L2)+NVCj[1],
              var=get.variables(O_L2),rhs=get.constraints(O_L2)))
}
```

A.2.7. L^2 con los cortes de Benders

```
seleccion_recursos_L2<-function(incendio,recursos,coste_area,
                       Pw=rep(1/length(incendio),length(incendio)),
                       B1=F,B2=F,imp=T){

  #---------------------------------------------------------------------
  # Se instalan los paquetes faltantes y se cargan

  paquetes.instalados<-installed.packages()[,1] # paquetes instalados

  # Comprobamos si los paquetes están instalados. En caso de no estar se instalan.
  if(sum("lpSolveAPI"==paquetes.instalados)==0 |
     sum("R.utils"==paquetes.instalados)==0){
    cat("-----------------------------------------------------\n")
    if(sum("lpSolveAPI"==paquetes.instalados)==0){
      cat("Instalacion de la librería: lpSolveAPI\n")
      install.packages("lpSolveAPI",repos="http://cran.es.r-project.org")
    }
    if(sum("R.utils"==paquetes.instalados)==0){
      cat("Instalacion de la librería: R.utils\n")
      install.packages("R.utils",repos="http://cran.es.r-project.org")
    }
    cat("-----------------------------------------------------\n")
  }
```

```
# Se cargan
library(lpSolveAPI) # Para resolver problemas de programacion lineal
library(R.utils) # Para la función insert
#----------------------------------------------------------------------

#Fijamos tiempo inicial
time<-proc.time()

Hj<-incendio[[1]]$Hj

# Datos de los recursos:
IndRec<-recursos$i # indices de los recursos
Ai<-recursos$Ai # tiempo de vuelo hasta el incendio
Ci<-recursos$Ci # coste de alquiler por hora
Pi<-recursos$Pi # coste de alquiler inicial
PRi<-recursos$PRi # produccion del recurso i por hora
NUMi<-recursos$NUMi # numero de recursos del tipo i
Orden<-order(recursos$DESi)
DESi<-c()
for(i in 1:length(Orden))DESi[i]<-levels(recursos$DESi)[which(Orden==i)]

# Si NUMi=0 para algún i, se elimina
if(sum(NUMi==0)!=0){
  NO_r<-which(NUMi==0)
  IndRec<-IndRec[-NO_r]
  Ai<-Ai[-NO_r]
  Ci<-Ci[-NO_r]
  Pi<-Pi[-NO_r]
  PRi<-PRi[-NO_r]
  NUMi<-NUMi[-NO_r]
  DESi<-DESi[-NO_r]
}

# Creamos cada recurso (de los NUMi) como una variable binaria
for(i in length(DESi):1){
  NUM<-NUMi[i]
  while(NUM>1){
    IndRec<-insert(IndRec,i,IndRec[i])
    Ai<-insert(Ai,i,Ai[i])
    Ci<-insert(Ci,i,Ci[i])
```

```
    Pi<-insert(Pi,i,Pi[i])
    PRi<-insert(PRi,i,PRi[i])
    DESi<-insert(DESi,i,DESi[i])
    NUM<-NUM-1
  }
}

n<-length(DESi) # numero de recursos
m<-length(Hj) # numero de periodos

#-------------------------------------------------
# Si no se fijan restricciones en el presupuesto:
if (B1==F) B1<-sum(Pi)
if (B2==F) B2<-max(Hj)*sum(Ci)
#---------------------------------------------------------------------------

# Paso 0: Inicializacion

# Fijamos LB, UB, eps y fijamos k (iterante):
r=s=h=z<-0 # Contadores numero de restricciones de fact, opt, factL2 y optL2
v<-0 # Contador numero de iteraciones
#--------------------------------------------------------------------------

# Definimos variables:
xv<-list() # Variables Zi
theta<--Inf # en tesis Lee v_k+1
lambda<-Inf # en tesis Lee V_k+1
eps<-0.001
K<-length(incendio) # Numero de escenarios

# Esperanza de los subproblemas
Exw<-0
# Coeficientes para Feasibility Cuts:
Dl<-c()
dl<-c()
# Coeficientes para Optimality Cuts:
El<-c()
el<-c()
# Coeficientes para L2 Feasibility Cut:
Gl<-c()
gl<-c()
```

```
# Coeficientes para L2 Optimality Cut:
Fl<-c()
fl<-c()
# Restricciones del Problema Master:
A<-c(Pi,0)
b<-B1

# Restriccion para factibilidad de los subproblemas binarios
A<-rbind(A,c(rep(1,n),0))

restr_A<-dim(A)[1]

b[2]<-0
res<-c()
for(k in 1:K){
  res[k]<-Restr_Master(incendio[[k]],Ai,Ci,Pi,PRi,coste_area,B1,B2)[[2]]
  b[2]<-max(b[2],Restr_Master(incendio[[k]],Ai,Ci,Pi,PRi,coste_area,
                            B1,B2)[[1]])
}

if(imp==T) cat("Iniciando resolucion del problema\n")

if(sum(res!=0)!=0){# Si el problema NO tiene solución
  if(imp==T){
    cat("El problema no tiene solución \n")
    cat("Escenarios Infactibles: ")
    cat(paste(which(res!=0)))
  }
}else{ # Si el problema SI tiene solución

  #------------------------------------------------
  # Fijamos la cota inferior (inicial) L:
  L<-L_Shaped(incendio,Pw,recursos,coste_area,B1,B2)
  #------------------------------------------------

  while(lambda-theta>eps){

    v<-v+1 # contador numero iteraciones

    #------------------------------------------------------------------------
    # Paso 1: Resolvemos Problema Master.
```

```
Master<-make.lp(restr_A+r+s+h+z,n+1)

for(l in 1:restr_A)  set.row(Master,l,A[l,])
if(r>0)  for(l in 1:r) set.row(Master,restr_A+l,c(Dl[l,],0))
if(s>0)  for(l in 1:s) set.row(Master,restr_A+r+l,c(El[l,],1))
if(h>0)  for(l in 1:h) set.row(Master,restr_A+r+s+l,c(Gl[l,],0))
if(z>0)  for(l in 1:z) set.row(Master,restr_A+r+s+h+l,c(Fl[l,],1))

set.objfn(Master,c(Pi,1))
set.rhs(Master,c(b,dl,el,gl,fl))
set.constr.type(Master,c("<=",rep(">=",restr_A-1+r+s+h+z)))
set.type(Master, 1:n, "binary") # Zi
set.type(Master, n+1, "real") # Mu

Resolucion<-solve(Master)

Master_Obj<-get.objective(Master)
xv[[v]]<-get.variables(Master)[1:n]
theta<-get.variables(Master)[n+1]


#--------------------------------------------------------------------------
# Paso 2: Resolvemos los subproblemas relajados

subprob<-list()
Subprob_Op<-list()
feasible<-c()

# Subproblemas:
for(k in 1:K){
  subprob[[k]]<-Feasible_L(xv[[v]],incendio[[k]],Ai,Ci,Pi,PRi,coste_area,B2)
  feasible[k]<-subprob[[k]]$Sol # indica si algun subproblema es infactible
}

if(sum(feasible!=0)!=0){ # Si hay algun escenario infactible

  # Tomamos todos los escenarios infactibles para realizar los cortes
  infeasible<-which(feasible!=0) # Escenarios infactibles para xv

  for(k in infeasible){
```

```
    r<-r+1
    dl[r]<-subprob[[k]]$dual%*%subprob[[k]]$rhs
    Dl<-rbind(Dl,-subprob[[k]]$dual%*%subprob[[k]]$Tec)
  }

  #----------------------------------------------------------------------
  # Paso 3: Creamos los cortes de optimalidad en caso de ser necesarios

}else{ # Si todos los escenarios son factibles

  s<-s+1
  Els<-0
  els<-0
  for(k in 1:K){
    Subprob_Op[[k]]<-Optimal_L(xv[[v]],incendio[[k]],Ai,Ci,
                               Pi,PRi,coste_area,B2)
    Els<-Els+Pw[k]*Subprob_Op[[k]]$dual%*%Subprob_Op[[k]]$Tec
    els<-els+Pw[k]*Subprob_Op[[k]]$dual%*%Subprob_Op[[k]]$rhs
  }
  El<-rbind(El,Els)
  el[s]<-els
}

#-----------------------------------------------------------------------
# Paso 4: Creamos los cortes de fact. L2 si algún subprobl. es infact.

subprobL2<-list()
feasibleL2<-c()

Exw[v]<-0
for(k in 1:K){
  subprobL2[[k]]<-Optimal_L2(xv[[v]],incendio[[k]],Ai,Ci,Pi,
                             PRi,coste_area,B2)
  feasibleL2[k]<-subprobL2[[k]]$resolver # indica subprobl. es infact
  Exw[v]<-Exw[v]+Pw[k]*subprobL2[[k]]$Sol
}

if(sum(feasibleL2!=0)!=0){
  h<-h+1
  Gl<-rbind(Gl,c(-2*xv[[v]]+1))
  gl[h]<--sum(xv[[v]])+1
```

```
  }else{
    #-------------------------------------------------------------------
    # Step 5: Creamos los cortes de opt. L2 si los subprobl. son fact.

    lambda<-min(Exw[v],lambda)
    z<-z+1
    Fl<-rbind(Fl,c((-Exw[v]+L)*(2*xv[[v]]-1)))
    fl[z]<-(Exw[v]-L)*(-sum(xv[[v]])+1)+L
  }
}

# Salida de la función:
if(imp==T) cat("Resolucion del problema terminada\n")

Variables<-list()
Subproblemas<-list()
EspSub<-0

Cost<-0
Pre<-get.constraints(Master)[1]

Dij<-list()
x<-list()
Yj<-list()
Cont_Period<-list()

for(k in 1:K){
  Subproblemas[[k]]<-Optimal_L2(xv[[v]],incendio[[k]],Ai,Ci,
                                Pi,PRi,coste_area,B2)
  Variables[[k]]<-Subproblemas[[k]]$var
  EspSub<-EspSub+Pw[k]*Subproblemas[[k]]$Sol
  Cost<-Cost+Pw[k]*Subproblemas[[k]]$rhs[1]
  Dij[[k]]<-matrix(Variables[[k]][1:(m*n)],ncol=m,nrow=n,byrow=T)
  Yj[[k]]<-c(Variables[[k]][(m*n+1):(m*n+m-1)],0)
  Cont_Period[[k]]<-sum(Yj[[k]])+1
}

x<-c()
desi<-c()
indi<-c()
```

```
INDICE<-IndRec
while(length(INDICE)!=0){
  x<-rbind(x,sum(xv[[v]][which(INDICE[1]==IndRec)]))
  desi<-c(desi,DESi[which(INDICE[1]==IndRec)][1])
  indi<-c(indi,INDICE[1])
  INDICE<-INDICE[-which(INDICE[1]==INDICE)]
}

Escenario<-list()
for(k in 1:K){
  for(i in 1:sum(Dij[[k]]!=0)){
    if(i==1){
      Escenario[[k]]<-Hj[which(Dij[[k]]!=0,arr.ind=T)[,2][i]]
    }else{
      Escenario[[k]]<-c(Escenario[[k]],
                        Hj[which(Dij[[k]]!=0,arr.ind=T)[,2][i]])
    }
  }
}

dij<-list()
for(k in 1:K){
  for(i in 1:length(indi)){
    if(i==1){
      dij[[k]]<-colSums(matrix(Dij[[k]][which(indi[i]==IndRec),],
                        ncol=m))
    }else{
      dij[[k]]<-rbind(dij[[k]],colSums(matrix(Dij[[k]][which(indi[i]==IndRec),],
                       ncol=m)))
    }
  }
}
if(imp==T){
  cat(paste("Tiempo compilacion:",round((proc.time()-time)[1]),"s \t(",
            round((proc.time()-time)[1]/60,digits=2),"min) \n"))
  cat(paste("Numero iteraciones: ",v,"\n"))
  cat("=========================================================\n")
  cat("Seleccion de Recursos: \n")
  for(i in which(x!=0)){
    cat(paste("Recurso:",indi[i],"(", desi[i],"), Unidades:",x[i],"\n"))
  }
```

```
      cat("===================================================\n")
      Indice=Descripcion=Cantidad<-c()
      for(k in 1:K){
        if(k!=1){
          cat("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
          cat("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n")
        }
        cat(paste("Escenario", k,": Periodo", Cont_Period[[k]] ,"\n"))
        cat(paste("--------------------------------------------------- \n"))
        for(i in which(dij[[k]]!=0)){
          if(is.null(dim(dij[[k]]))){
            cat(paste("Recurso:",indi,"(", desi,"),"))
            cat(paste("\tUnidades:",dij[[k]][i],",\t Periodo:",Hj[i],"\n"))
          }else{
            cat(paste("Recurso:",indi[(i-1)%%dim(dij[[k]])[1]+1],"(",
                      desi[(i-1)%%dim(dij[[k]])[1]+1],"),"))
            cat(paste("\tUnidades:",dij[[k]][i],",\t Periodo:",
                      Hj[(i-1)%/%dim(dij[[k]])[1]+1],"\n"))
          }
        }
      }
      cat("======================================================= \n")
      cat(paste("Coste asignacion optima: ", round(Master_Obj),"\n"))
      cat(paste("Pre: ", round(Pre),"\n"))
      cat(paste("Cost: ", round(Cost),"\n"))
      cat(paste("NVC: ", round(Master_Obj-Cost-Pre),"\n"))
    }else{
      result<-list(Dij=Dij,Yj=Yj,Zi=x,Obj=Master_Obj)
      return(result)
    }
  }
}
```

A.3. Modelo determinista incluyendo asignaciones horarias

```
asignacion_recursos<-function(incendio,recursos,coste_area,frentes=1,InstCont=F,
                              imp=T,aprox=T){

  # Cargamos los paquetes necesarios
```

```
library(slam)
library(gurobi)

#Fijamos tiempo inicial
time<-Sys.time()

Hj2<-incendio$Hj # periodo con estimacion del incendio
SPj2<-incendio$SPj # perimetro del incendio en j
Area2<-incendio$area # area del incendio en j
IndRec<-recursos$i # indices de los recursos
Ai<-recursos$Ai # tiempo de vuelo de i hasta el incendio
Ci<-recursos$Ci # coste de alquiler por hora
PRi<-recursos$PRi # produccion del recurso i por hora
NUMi<-recursos$NUMi # numero de recursos del tipo i
TVi<-recursos$TVi # tiempo de actividad maximo del recurso i
TVTi<-recursos$TVTi # tiempo actividad que lleva el recurso i
TDi<-recursos$TDi # Tiempo de descanso minimo del recurso i
TVDi<-recursos$TVDi # Tiempo de actividad diaria maxima del recurso i
DESCi<-recursos$DESCi # Si la aeronave i esta descansada o no
Orden<-order(recursos$DESi)
DESi<-c()
for(i in 1:length(NUMi))DESi[i]<-levels(recursos$DESi)[which(Orden==i)]

# Si NUMi=0, se elimina ese recurso
if(sum(NUMi==0)!=0){
  NO_r<-which(NUMi==0)
  IndRec<-IndRec[-NO_r]
  Ai<-Ai[-NO_r]
  Ci<-Ci[-NO_r]
  PRi<-PRi[-NO_r]
  NUMi<-NUMi[-NO_r]
  DESi<-DESi[-NO_r]
  TVi<-TVi[-NO_r]
  TVTi<-TVTi[-NO_r]
  TDi<-TDi[-NO_r]
  TVDi<-TVDi[-NO_r]
  DESCi<-DESCi[-NO_r]
}

# Aproximamos los valores del tiempo (de 10 min en 10 min)
if(aprox==T){
```

```
    for(i in 1:length(NUMi)){
      if(Ai[i]%%(1/6)!=0){
        for(h in 1:6){
          if(floor(Ai[i])<Ai[i]+(h-1)/6 &&
             floor(Ai[i])+h/6>Ai[i] && Ai[i]%%(1/6)!=0){
            Ai[i]<-floor(Ai[i])+h/6
          }
        }
      }
    }
  }

  TAi<-numeric(length(NUMi))
  if(sum(!DESCi)!=0){
    i<-which(DESCi==F)
    TAi[i]<-TVTi[i]%%(TVi[i]+TDi[i])
  }

  if(aprox==T){
    for(i in 1:length(NUMi)){
      if(round(TAi[i]%%(1/6),digits=4)!=0){
        for(h in 1:6){
          if(floor(TAi[i])<TAi[i]+(h-1)/6 &&
             floor(TAi[i])+h/6>TAi[i] && TAi[i]%%(1/6)!=0){
            TAi[i]<-floor(TAi[i])+h/6
          }
        }
      }
    }
  }

  # Los intervalos de asignacion se construyen a partir de los tiempos
  # de actividad sin descansos, tiempos de llegada al incendio, tiempos
  # de actividad total realizada y tiempos de descanso:

  TAM<-round(TVi[1]/TDi[1],digits=4)
  TAMA<-round(Ai/TDi,digits=4)
  TA<-round(TAi/TDi,digits=4)
  TDM<-1

  b<-1
```

```
while((TAM*b)%%1!=0){
  if((TAM*(b+1))%%1!=0){
    b<-b+1
  }else{
    TAM<-TAM*(b+1)
    TDM<-TDM*(b+1)
    TAMA<-TAMA*(b+1)
    TA<-TA*(b+1)
  }
}

for(i in 1:length(DESi)){
  b<-1
  while((TAMA[i]*b)%%1!=0){
    if((TAMA[i]*(b+1))%%1!=0){
      b<-b+1
    }else{
      TAMA<-TAMA*(b+1)
      TAM<-TAM*(b+1)
      TDM<-TDM*(b+1)
      TA<-TA*(b+1)
    }
  }
}

for(i in 1:length(DESi)){
  b<-1
  while((TA[i]*b)%%1!=0){
    if((TA[i]*(b+1))%%1!=0){
      b<-b+1
    }else{
      b<-b+1
      TAMA<-TAMA*(b)
      TAM<-TAM*(b)
      TDM<-TDM*(b)
      TA<-TA*(b)
    }
  }
}

# Se construyen los periodos Hj y se les asigna sus
```

```
# correspondientes parametros.
Intervalos<-ceiling((max(Hj2)-min(Hj2))/TDi[1])*TDM+2

Hj<-numeric(Intervalos)
SPj<-numeric(Intervalos-1)
Area<-numeric(Intervalos-1)

for(t in 1:Intervalos){
  Hj[t]<-min(Hj2)+(t-2)*(TDi[1]/TDM)
}

# Tiempos de descanso que realiza en el momento inicial
TDMIi<-numeric(length(DESi))
for(i in 1:length(DESi)){
  while((Hj[2]-Hj[1])*TDMIi[i]<TAi[i]-TVi[i]){
    TDMIi[i]<-TDMIi[i]+1
  }
}

for(t in 1:(Intervalos-1)){
  if(sum(Hj[t+1]<=Hj2)!=0){
    j<-min(which(Hj[t+1]<=Hj2))

    SPj[t]<-SPj2[j]
    Area[t]<-Area2[j]
  }else{
    SPj[t]<-SPj2[length(SPj2)]
    Area[t]<-Area2[length(Area2)]
  }
}

# Creamos cada recurso (de los NUMi) como una variable binaria
for(i in length(DESi):1){
  NUM<-NUMi[i]
  while(NUM>1){
    IndRec<-insert(IndRec,i,IndRec[i])
    Ai<-insert(Ai,i,Ai[i])
    Ci<-insert(Ci,i,Ci[i])
    PRi<-insert(PRi,i,PRi[i])
    DESi<-insert(DESi,i,DESi[i])
    TVi<-insert(TVi,i,TVi[i])
```

```
    TVTi<-insert(TVTi,i,TVTi[i])
    TDi<-insert(TDi,i,TDi[i])
    TDMIi<-insert(TDMIi,i,TDMIi[i])
    TVDi<-insert(TVDi,i,TVDi[i])
    TAi<-insert(TAi,i,TAi[i])
    DESCi<-insert(DESCi,i,DESCi[i])
    NUM<-NUM-1
  }
}

# Orden de las variables de decision:
# Dij Orden (D11,D12,...,D21,D22,...)
# DESij Orden (DES11,DES12,...,DES21,DES22,...)
# Bij Orden (B11,B12,...,B21,B22,...)
# Yj
# Zi
# FDij Orden (FD11,FD12,...,FD21,FD22,...)

n<-length(IndRec) # numero de recursos
m<-length(Hj)-1 # numero de periodos
n_var<-(n*m)+(n*m)+(n*m)+(m)+(n)+(n*m) # numero de variables
n_restr<-1+1+m+n+n+n*m+n*m+m+n+n+(n*m)+n+n*m # numero de restricciones

M<-max(SPj)+sum((Hj[m+1]-Hj[2])*PRi) # Valor de M (restriccion 9)

# Incremento del perimetro del incendio en el periodo j
PERj<-numeric(m)
PERj[1]<-SPj[1]
for(j in 2:m){
  PERj[j]<-SPj[j]-SPj[j-1]
}

# Incremento del area del incendio en el periodo j
InArea<-numeric(m)
InArea[1]<-Area[1]
for(j in 2:m){
  InArea[j]<-Area[j]-Area[j-1]
}
NVCj<-InArea*coste_area

#----------------------------------------
```

```
# Si se desea omitir o no la restriccion 1
if(InstCont==F){ # se omite
  B1<-length(Hj)
}else{ # a partir de la hora fijada se establece B1
  B1<-length(which(Hj[-1]<=InstCont))
}
#----------------------------------------

li<-numeric(n) #numero de periodos que forman el tiempo de vuelo
for(i in 1:n){
  li[i]<-which(Hj[2:(m+1)]==Hj[2]+Ai[i])-1
  A<-Ai[i]
  while(A>TVi[i]-TAi[i]){
    li[i]<-li[i]+which(Hj[2:(m+1)]==Hj[2]+TDi[i])-1
    A<-A-TVi[i]
  }
}

#------------------------------------------------
sense<-c("<=",rep(">=",1+m),rep("<=",n+n+n*m),rep(">=",n*m+n*m),rep("<=",m),
         rep(">=",n),rep("<=",n),rep(">=",n+n*m))

# LADOS DERECHOS
rhs<-numeric(n_restr)
rhs[1]<-B1-2 # restr1
rhs[2]<-PERj[1] # restr2
rhs[2+m+n+n+n*m+n*m+n*m+1]<--SPj[1] # restr9
rhs[2+m+n+n+n*m+n*m+n*m+m+n+1:n]<-TVDi-TVTi # restr11

# LADOS IZQUIERDOS
# Coeficientes primera restriccion:
restr1<-numeric(n_var)
# Coeficientes segunda restriccion:
restr2<-numeric(n_var)
# Coeficientes terceras restricciones:
restr3<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes cuartas restricciones:
restr4<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes quintas restricciones:
restr5<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes sextas restricciones:
```

```
restr6<-matrix(0,nrow=n*m,ncol=n_var)
# Coeficientes septimas restricciones:
restr7<-matrix(0,nrow=n*m,ncol=n_var)
# Coeficientes octavas restricciones:
restr8<-matrix(0,nrow=n*m,ncol=n_var)
# Coeficientes novenas restricciones:
restr9<-matrix(0,nrow=m,ncol=n_var)
# Coeficientes decimas restricciones:
restr10<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes onceavas restricciones:
restr11<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes decimosegundas restricciones:
restr12<-matrix(0,nrow=n,ncol=n_var)
# Coeficientes decimoterceras restricciones:
restr13<-matrix(0,nrow=n*m,ncol=n_var)


# Funcion objetivo:
obj<-numeric(n_var)

for(i in 1:n){

  restr4[i,1:m+(i-1)*m]<-1 # Coeficientes Dij
  restr4[i,3*(n*m)+m+i]<--1 # Coeficientes Zi

  if(Ai[i]==0){
    restr5[i,2*(n*m)+1+(i-1)*m]<-1 # Coeficiente Bi1
    restr5[i,2*(n*m)+2:m+(i-1)*m]<-m+1 # Coeficientes Bij
    restr5[i,3*(n*m)+m+i]<--m # Coeficientes de Zi
  }else{
    restr5[i,2*(n*m)+1:m+(i-1)*m]<-1 # Coeficientes Bij
    restr5[i,3*(n*m)+m+i]<--1 # Coeficientes de Zi
  }

  for(j in 1:m){
    rhs[2+m+n+n+j+(i-1)*m]<-TVi[i]-TAi[i] # restr6
    rhs[2+m+n+n+n*m+j+(i-1)*m]<--TAi[i] # restr7

    restr1[3*(n*m)+j]<-1 # Coeficientes Yj

    restr2[j+(i-1)*m]<--(Hj[m+1]-Hj[j+1])*PRi[i] # Coeficientes Dij
```

```
restr2[(m*n)+j+(i-1)*m]<--(Hj[j+1]-Hj[j])*PRi[i] # Coeficientes DESij
restr2[2*(m*n)+j+(i-1)*m]<-(Hj[m+1]-Hj[j]-Ai[i])*PRi[i] # Coeficientes Bij

if(j>min(li)){
  restr3[j,3*(n*m)+j]<--frentes # Coeficientes Yj
}

if(j>li[i]){
  restr3[j,2*(n*m)+(i-1)*m+1:(j-li[i])]<-1 # Coeficientes Bij
  restr3[j,(n*m)+j+(i-1)*m]<--1 # Coeficientes DESij
  for(k in 1:j){
    if(j>1){
      restr3[j,k+(i-1)*m-1]<--1 # Coeficientes Dij-1
    }
  }
}


restr9[j,3*(n*m)+j]<--M # Coeficientes Yj

restr10[i,j+(i-1)*m]<-Hj[j+1] # Coeficientes Dij
restr10[i,2*(n*m)+j+(i-1)*m]<--Hj[j+1] # Coeficientes Bij

restr11[i,j+(i-1)*m]<--(Hj[m+1]-Hj[j+1]) # Coeficientes Dij
restr11[i,2*(m*n)+j+(i-1)*m]<-Hj[m+1]-Hj[j] # Coeficientes Bij

restr12[i,j+(i-1)*m]<--(Hj[m+1]-Hj[j]) # Coeficientes Dij
restr12[i,2*(m*n)+j+(i-1)*m]<-Hj[m+1]-Hj[j] # Coeficientes Bij
restr12[i,(m*n)+j+(i-1)*m]<--(Hj[j+1]-Hj[j]) # Coeficientes DESij
restr12[i,3*(m*n)+m+i]<--Ai[i] # Coeficientes Zi

restr13[j+(i-1)*m,(m*n)+j+(i-1)*m]<--1 # Coeficientes DESij
restr13[j+(i-1)*m,3*(n*m)+j]<-1 # Coeficientes Yj

if(j>1){
  restr2[3*(n*m)+j-1]<--PERj[j] # Coeficientes Yj-1

  restr9[j,3*(n*m)+j-1]<-SPj[j] # Coeficientes Yj-1
}

if(j-TDM>=0){
```

```
      restr8[j+(i-1)*m,(n*m)+((j-TDM+1):j)+(i-1)*m]<-1 # Coeficientes DESij
      restr8[j+(i-1)*m,3*(n*m)+n+m+j+(i-1)*m]<--TDM # Coeficientes FDij
    }else{
      restr8[j+(i-1)*m,(n*m)+(1:j)+(i-1)*m]<-1 # Coeficientes DESij
      restr8[j+(i-1)*m,3*(n*m)+n+m+j+(i-1)*m]<--TDM+TDMIi[i] # Coeficientes FDij
    }



    for(k in 1:j){
      restr6[j+(i-1)*m,(i-1)*m+k]<--(Hj[j+1]-Hj[k+1]) # Coeficientes Dij
      restr6[j+(i-1)*m,2*(n*m)+(i-1)*m+k]<-Hj[j+1]-Hj[k] # Coeficientes Bij
      restr6[j+(i-1)*m,n*m+(i-1)*m+k]<--TDi[i]/TDM # Coeficientes DESij
      restr6[j+(i-1)*m,3*(n*m)+n+m+k+(i-1)*m]<--TVi[i] # Coeficientes FDij

      restr7[j+(i-1)*m,(i-1)*m+k]<--(Hj[j+1]-Hj[k+1]) # Coeficientes Dij
      restr7[j+(i-1)*m,2*(n*m)+(i-1)*m+k]<-Hj[j+1]-Hj[k] # Coeficientes Bij
      restr7[j+(i-1)*m,n*m+(i-1)*m+k]<--TDi[i]/TDM # Coeficientes DESij
      restr7[j+(i-1)*m,3*(n*m)+n+m+k+(i-1)*m]<--TVi[i] # Coeficientes FDij

      restr9[j,(i-1)*m+k]<-(Hj[j+1]-Hj[k+1])*PRi[i] # Coeficientes Dij
      restr9[j,(n*m)+(i-1)*m+k]<-(Hj[j+1]-Hj[j])*PRi[i] # Coeficientes DESij
      restr9[j,2*(n*m)+(i-1)*m+k]<--(Hj[j+1]-Hj[k]-Ai[i])*PRi[i] # Coef. Bij
    }

    obj[j+(i-1)*m]<-Hj[j+1]*Ci[i] # Coeficientes Dij
    obj[2*(n*m)+j+(i-1)*m]<--Hj[j]*Ci[i] # Coeficientes Bij
    obj[(n*m)+j+(i-1)*m]<--(Hj[j+1]-Hj[j])*Ci[i]+0.0001*Hj[j+1] # Coef. DESij
    if(j>1){
      obj[3*(n*m)+j-1]<-(Hj[j+1]/Hj[m+1])+NVCj[j] # Coeficientes Yj-1
    }
    obj[3*(n*m)+m]<-1 # Coeficiente Ym
    obj[3*(n*m)+m+i]<-1 # Coeficientes Zi
  }
}

restr<-round(rbind(restr1,restr2,restr3,restr4,restr5,restr6,restr7,restr8,
                   restr9,restr10,restr11,restr12,restr13)*60,digits=2)

################################################################################
# El modelo
```

```
S_rec<-list()
S_rec$A<-restr
S_rec$obj<-obj
S_rec$sense<-sense
S_rec$rhs<-rhs*60
S_rec$vtypes<-c(rep("B",n_var))
S_rec$modelsense<-"min"

sol<-gurobi(S_rec)
#=================================================================

if(sol$status=="OPTIMAL"){ # Si la solucion es optima

  Var<-round(sol$x)

  # Valores optimos de las variables:
  Dij<-matrix(Var[1:(n*m)],ncol=m,byrow=T)
  DESij<-matrix(Var[(n*m+1):(2*n*m)],ncol=m,byrow=T)
  Bij<-matrix(Var[(2*n*m+1):(3*n*m)],ncol=m,byrow=T)
  Yj<-Var[(3*n*m+1):(3*n*m+m)]
  Zi<-Var[(3*n*m+m+1):(3*n*m+m+n)]
  FDij<-matrix(Var[(3*n*m+m+n+1):(n_var)],ncol=m,byrow=T)

  # Periodo de actividad:
  Act<-matrix(0,nrow=n,ncol=m)
  for(i in 1:n){
    if(sum(Bij[i,]==1)!=0){
      Act[i,Bij[i,]==1]<-1
      j<-which(Bij[i,]==1)
      if(Dij[i,j]!=Bij[i,j]){
        while(Dij[i,j]==0){
          j<-j+1
          Act[i,j]<-1
        }
      }
    }
  }

  Tiempos<-format(floor(Hj[-1])+round(Hj[-1]%%1*60)/100,nsmall=2,decimal.mark=":")

  Act<-round(Act-DESij)
```

```
Act_Rec<-Act+2*DESij
rownames(Act_Rec)<-DESi
colnames(Act_Rec)<-Tiempos

# Cost:
cost<-round((Hj[2]-Hj[1])*sum(Ci%*%Act),digits=2)

# NVC:
NVC<-round(c(1,Yj[-m])%*%NVCj,digits=2)

# El incendio se controla en el periodo:
if(sum(Yj==0)!=0){
  contr1<-Tiempos[min(which(Yj==0))]
  contr2<-Tiempos[1+min(which(Yj==0))]
}else{
  contr1<-format(floor(Hj[length(Hj)])+round(Hj[length(Hj)]%%1*60)/100,
               nsmall=2,decimal.mark=":")
  contr2<-format(floor(2*Hj[length(Hj)]-Hj[length(Hj)-1])+
                     round((2*Hj[length(Hj)]-Hj[length(Hj)-1])%%1*60)/100,
               nsmall=2,decimal.mark=":")
}

# Periodo en el que empiezan a actuar las aeronaves
momento_act_i<-numeric(n)
descansos_llegada_i<-numeric(n)

for(i in which(Zi==1)){
  Vuelo_Ini<-0
  j<-1
  while(round(Vuelo_Ini,digits=4)<=round(Ai[i],digits=4)){
    Vuelo_Ini<-sum(Act[i,1:j]*(Hj[2]-Hj[1]))
    j<-j+1
  }

  momento_act_i[i]<-j-1
  descansos_llegada_i[i]<-length(which(Act_Rec[i,1:momento_act_i[i]]==2))

  if(Ai[i]!=0){
    Act_Rec[i,which(Act_Rec[i,1:(momento_act_i[i]-1)]==1)]<-3
    Act_Rec[i,which(Act_Rec[i,1:(momento_act_i[i]-1)]==2)]<-4
```

```
    Act[i,which(Act[i,1:(momento_act_i[i]-1)]==1)]<-0
  }
}

# Linea construida:
Lj<-numeric(m)
for(j in 1:m){
  for(i in which(Zi==1)){
    Lj[j]<-Lj[j]+(Hj[j+1]-Hj[j])*PRi[i]*Act[i,j]
  }
}
for(j in 1:m){
  Lj[j]<-sum(Lj[(j-1):j])
}

# Porcentaje de contencion del incendio
PorInc<-Lj/SPj*100
if(sum(PorInc<0)!=0){
  PorInc[PorInc<0]<-0
}
if(sum(round(Lj,digits=4)>=round(SPj,digits=4))!=0){
  PorInc[which(round(Lj,digits=4)>=round(SPj,digits=4))[1]:m]<-100
}

Act_Rec<-rbind(Act_Rec,round(PorInc))
rownames(Act_Rec)[dim(Act_Rec)[1]]<-"PORCENTAJE"


finaltime<-(Sys.time()-time)

if(imp==T){
  cat("\n")
  cat("============================================================== \n")
  cat("SOLUCION OPTIMA \n")
  cat("=====================================================\n")
  print(round(finaltime))
  cat(paste("Numero Variables:",n_var,"\n"))
  cat(paste("Numero Restricciones:",n_restr,"\n"))
  cat("-----------------------------------------------------\n")
  cat(paste("Seleccion de Recursos:\n"))
  cat(paste("(Notacion: 0=NoActua, 1=Actua, 2=Descansa, 3=VueloIncendio,
```

```
                4==DescansoVueloIncendio)\n\n"))
      print(Act_Rec)
      cat("\n")
      cat(paste("El incendio se contiene entre las ",contr1," y las ",
                contr2, "\n"))
      cat("\n")
      cat("------------------------------------------------------ \n")
      cat(paste("Coste asignacion optima: ", cost+NVC,"\n"))
      cat(paste("Cost: ", cost,"\n"))
      cat(paste("NVC: ", NVC,"\n"))

    }else{
      result<-list(Bij=Bij,DESij=DESij,Dij=Dij,Yj=Yj,Zi=Zi)
      return(result)
    }
  }else{
      cat("============================================================== \n")
      cat("EL INCENDIO NO SE PUEDE CONTROLAR EN EL PERIODO ESTIMADO \n")
      cat("============================================================== \n\n")
  }
}
```

Bibliografía

[1] Benders, J. F. (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik* **4** (15), 238-252.

[2] Birge, J. R. and Louveaux, L. (1997) Introduction to Stochastic Programming. New York, New York: Springer.

[3] Butry, D. T., Mercer, D. E., Prestemon, J. P., Pye, J. M. and Holmes, T. P. (2001) What is the price of catastrophic wildfire? *Journal of Forestry*, **99** (11), 9-17.

[4] Caroe, C. C. and Schultz, R. (1996) Dual decomposition in stochastic integer programming. *Zentrum für Informationstechnik Berlin, ZIB*. Publicado en Operations Letters (1999), **24** (1-2), 37-45.

[5] Caroe, C. C. and Tind, J. (1998) L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming* **83**, 451-464.

[6] Donovan, G. and Rideout, D. (2003) An integer programming model to optimize resource allocation for wildfire containment. *Forest Science* **49**(2), 331-335.

[7] Finney, M. A. (1998) Fire Area Simulator-Model Development and Evaluation. *United States Departament of Agriculture, Forest Service*, Research Paper RMRS-RP-4, 47 p.

[8] Gorte, J. K. and Gorte, R. W. (1979) Application of Economic Techniques to Fire Management-A status Review and Evaluation. *USDA Forest Service General Technical Report* INT-53.

[9] Greulich, F. E. (2003) Airtanker initial attack: a spreadsheet-based modeling procedure. *Canadian Journal of Forest Research*, **33** (2), 232-242.

[10] Headley, R. (1916) Fire Suppression, District 5. *United States Departament of Agriculture, Forest Service*, Washington, 57 p.

[11] Hu, X. and Ntaimo, L. (2009) Integrated simulation and optimization for wildfire containment. *ACM Transactions on Modeling and Computer Simulation* **19**, 1-29.

[12] Laporte, G. and Louveaux, F. V. (1993) The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters*, **13** (3), 133-142.

[13] Lee, W. J. (2006) *A Stochastic Mixed Integer Programming Approach to Wildfire Management Systems*. Tesis doctoral. Department of Industrial and Systems Engineering. Texas A&M University.

[14] Minas, J. P., Hearne, J. W. and Handmer, J. W. (2012) A review of operations research methods applicable to wildfire management. *International Journal of Wildland Fire*, **21** (3), 189-196.

[15] Ministerio de Fomento, Dirección General de Aviación Civil, España (2001). *Anexo nº. 1 a Circular Operativa 16-B*. Madrid. `http://www.seguridadaerea.gob.es/media/Migracion/PDF/71765/1456.pdf`

[16] Morton, P. (2000) Wildland economics: theory and practice. *USDA Forest Service Proceedings* RMRS-P-15-Vol-2, 238-250.

[17] National Wildfire Coordinating Group Fireline Handbook (1998) *NWCG Handbook* 3. PMS 410-1. NFES 0065.

[18] Slyke, R. M. and Wets, R. (1969) L-Shape linear programs with applications to optimal control and stochastic programming. *SIAM Journal of Applied Mathematics*, **17** (4), 638-663.

[19] Sparhawk, W. N. (1925) The use of liability ratings in planning forest fire protection. *Journal of Agricultural Research* **30** (8), 693-792.

[20] Winston, W. L. (1994) Operations Research Applications and Algorithms. *Duxbury Press*, Belmont, California.