
El calendario de la NBA: Modelos de optimización y metaheurísticas



TRABAJO DE FIN DE GRADO

Doble Grado en Ingeniería Informática y Matemáticas

Autor:
Javier Amado Lázaro

Director:
Francisco Javier Martín Campo

Universidad Complutense de Madrid

1 de julio de 2025

Agradecimientos

A mi familia, en especial a mis padres, por educarme con valores, por su apoyo constante, por enseñarme a no rendirme nunca, por su cariño y por hacer posible que nunca me faltara de nada.

A Lucía, por motivarme, por enseñarme, por ser una gran fuente de inspiración y un ejemplo a seguir, por su apoyo incondicional y por creer en mí siempre.

A mis amigos de Leganés y del tenis, por confiar en mí y animarme en cada paso.

A los amigos que he conocido en esta maravillosa etapa, en especial a los *escape roomers*, por todas las experiencias y momentos que hemos vivido juntos, tanto buenos como malos.

A Javier Martín Campo, por dedicarme su tiempo, por su atención en todo momento y por todo lo que he aprendido durante estos meses.

Resumen

La planificación de partidos en el deporte profesional es una tarea compleja que impacta tanto en la salud de los jugadores como en aspectos económicos, logísticos y mediáticos de la competición. Este Trabajo de Fin de Grado se centra en el diseño de un calendario optimizado para la temporada regular de la NBA, con el objetivo principal de minimizar las distancias recorridas por los equipos a lo largo del curso competitivo. En este contexto se prioriza el bienestar de los deportistas, buscando una reducción significativa en los desplazamientos para disminuir la fatiga acumulada.

En primer lugar, se realiza un análisis del funcionamiento del calendario oficial de la NBA, identificando las restricciones y reglas clave mediante una revisión bibliográfica. Posteriormente, se desarrolla un modelo matemático exacto capaz de representar dichas restricciones, cuya resolución se aborda mediante el optimizador Gurobi.

Sin embargo, debido a la alta complejidad del problema, se opta por implementar algoritmos heurísticos (intercambio de jornadas, cambio de partidos local-visitante, etc.) y metaheurísticos (Temple Simulado, VNS y Búsqueda Tabú) que permiten obtener soluciones aproximadas en tiempos de cómputo razonables, explorando el espacio de soluciones de forma eficiente.

Como resultado, se ha conseguido una herramienta que combina generación exacta con optimización heurística, obteniendo soluciones que presentan una mejora sustancial en los desplazamientos totales frente al calendario oficial. Todos ello se ha realizado con un enfoque modular, lo que permite futuras ampliaciones y adaptaciones a otros contextos.

Palabras clave: NBA, optimización lineal entera, heurística, metaheurística, distancia.

Abstract

Planning matches in professional sports is a complex task that impacts both the health of players and the economic, logistical, and media aspects of the competition. This Final Degree Project focuses on designing an optimised schedule for the NBA regular season, with the main objective of minimising the distances travelled by teams throughout the competitive season. In this context, the well-being of athletes is prioritised, seeking a significant reduction in travel to decrease accumulated fatigue.

First, an analysis of the functioning of the official NBA calendar is carried out, identifying key restrictions and rules through a literature review. Subsequently, an exact mathematical model capable of representing these restrictions is developed, which is solved using the Gurobi optimiser.

However, due to the high complexity of the problem, heuristic algorithms (exchange of days, change of home-away games, etc.) and metaheuristic algorithms (Simulated Annealing, VNS and Tabu Search) are implemented to obtain approximate solutions in reasonable computing times, efficiently exploring the solution space.

As a result, a tool has been developed that combines exact generation with heuristic optimisation, obtaining solutions that represent a substantial improvement in total travel compared to the official schedule. All of this has been done with a modular approach, allowing for future expansion and adaptation to other contexts.

Keywords: NBA, integer linear optimisation, heuristics, metaheuristics, distance.

Índice general

1. Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos	1
1.3. Plan de trabajo	2
1.4. Estructura de la memoria	2
2. Descripción del problema	4
2.1. Estado del arte	4
2.2. Introducción a la NBA	5
2.3. La estructura del calendario NBA	6
3. Modelo de programación matemática	8
3.1. Conjuntos de índices	8
3.2. Parámetros	8
3.3. Modelos	9
3.3.1. Modelo 1: Generación de un calendario factible	9
3.3.1.1. Variables de decisión	9
3.3.1.2. Función objetivo	9
3.3.1.3. Restricciones	9
3.3.2. Modelo 2: Minimización de distancias	10
3.3.2.1. Variables de decisión	11
3.3.2.2. Función Objetivo	11
3.3.2.3. Restricciones	11
4. Heurísticas	14
4.1. Heurísticas basadas en calendarios estructurados	14
4.1.1. Heurística 1: Generación de múltiples calendarios	14
4.1.2. Heurística 2: Intercambios de jornadas	15
4.1.3. Heurística 3: Intercambios local-visitante	16
4.2. Heurísticas basadas en calendarios no estructurados	17
4.2.1. Heurística 4: Inserción	17
5. Metaheurísticas	20
5.1. Temple simulado	20
5.2. Búsqueda por Entornos Variables	22
5.3. Búsqueda Tabú	24
6. Resultados	26
6.1. Entorno de experimentación	26
6.1.1. Especificaciones del equipo	26
6.1.2. Herramientas y lenguajes utilizados	26
6.2. Casos de estudio y análisis de resultados	27
6.2.1. Especificaciones en las metaheurísticas	27

6.2.2.	Calendarios estructurados	28
6.2.2.1.	Liga de 4 equipos	28
6.2.2.2.	Liga de 8 equipos	30
6.2.2.3.	Liga regular	33
6.2.3.	Calendarios no estructurados	36
6.2.3.1.	Enfoque 1	36
6.2.3.2.	Enfoque 2	37
6.3.	Discusiones finales	37
7.	Conclusiones y trabajo futuro	39
7.1.	Trabajo futuro	39
A.	Archivos desarrollados	41
A.1.	Datos	41
A.2.	Calendarios iniciales	41
A.3.	Código fuente	41
A.3.1.	Modelos	42
A.3.2.	Heurísticas	42
A.3.3.	Metaheurísticas	43
A.3.3.1.	Temple simulado	43
A.3.3.2.	VNS	43
A.3.3.3.	Búsqueda Tabú	44
A.4.	Resultados	44
A.4.1.	Modelos	44
A.4.2.	Heurísticas	44
A.4.3.	Metaheurísticas	45
A.4.3.1.	Temple Simulado	45
A.4.3.2.	VNS	46
A.4.3.3.	Búsqueda Tabú	47
B.	Detalles de implementación	49
B.1.	Estructuras de datos	49
B.1.1.	Información de los equipos	49
B.1.2.	Matriz de distancias	49
B.1.3.	Recorridos y rivales	50
B.1.4.	Balance del calendario	50
Bibliografía		51

Capítulo 1

Introducción

1.1. Contexto y motivación

La planificación en el deporte profesional es un aspecto fundamental que impacta directamente en la calidad de la competición, el interés de los aficionados, la salud de los deportistas y el negocio alrededor del espectáculo. El calendario oficial de partidos de la NBA (*National Basketball Association*) no solo estructura el desarrollo de la temporada, sino que también se convierte en un eje estratégico para maximizar el rendimiento de los equipos y la experiencia de los seguidores.

Para los aficionados, el calendario es la guía esencial que les permite organizar su asistencia a los partidos, seguir a sus equipos o jugadores favoritos y mantener viva la pasión por el deporte. Para los jugadores y entrenadores, la planificación del calendario influye en la gestión mental y física. El diseño de un mal calendario puede derivar en una carga de viajes excesiva, fatiga acumulada y mayor riesgo de lesiones, por lo que la salud y el rendimiento de los deportistas dependen en gran medida de la confección de un calendario equilibrado que contemple descansos adecuados y reduzca la fatiga provocada por desplazamientos largos y frecuentes.

Por otra parte, la organización de la NBA y las emisoras de televisión tienen un interés económico crucial en el calendario ya que la programación de los partidos influye en la audiencia, los ingresos de publicidad y derechos de retransmisión, así como en la promoción internacional de la liga.

Equilibrar estos múltiples intereses para que todas las partes se sientan beneficiadas convierte la planificación del calendario de la NBA en un problema muy complejo de resolver.

En este trabajo se opta por centrar la atención en el bienestar de los jugadores, abordando el problema de optimizar el calendario de la NBA con el objetivo de minimizar las distancias recorridas por los equipos a lo largo de la temporada regular. Se busca reducir la carga de desplazamientos que sufren los jugadores y el *staff* de los equipos con el objetivo de mejorar sus condiciones físicas, disminuir el riesgo de lesiones y de esta manera poder elevar la calidad y competitividad del campeonato.

1.2. Objetivos

El objetivo principal de este trabajo se centra en diseñar un calendario para la temporada regular de la NBA, atendiendo a la necesidad de minimizar las distancias que recorren los equipos durante la temporada regular, contribuyendo así a una planificación más eficiente desde el punto de vista logístico y económico, además de reducir la fatiga asociada a los desplazamientos.

Cabe destacar que la planificación del calendario realizada durante este proyecto se limita exclusivamente a la asignación de partidos por jornadas o días, sin entrar en ningún momento

en la asignación específica de horarios diarios. El objetivo es estructurar la secuencia temporal de los enfrentamientos, sin interferir en aspectos logísticos propios de cada organización local o de la NBA en relación con franjas horarias concretas.

Para alcanzar este objetivo general, se han definido una serie de objetivos específicos que estructuran el desarrollo del proyecto y guían las distintas fases de trabajo:

- Formular modelos matemáticos exactos que representen fielmente las restricciones del calendario oficial de la NBA.
- Diseñar e implementar algoritmos heurísticos que permitan obtener soluciones aproximadas de buena calidad en tiempos de cómputo razonables sin necesidad de resolver el problema exacto.
- Investigar y aplicar técnicas metaheurísticas con el fin de mejorar los resultados obtenidos por las heurísticas básicas, explorando de manera más efectiva el espacio de soluciones y escapando de óptimos locales.
- Comparar y evaluar el rendimiento de los diferentes algoritmos implementados, tanto en términos de calidad de la solución como de eficiencia computacional.

1.3. Plan de trabajo

En primer lugar, ante la necesidad de comprender en profundidad el funcionamiento del calendario oficial de la NBA, se realizó una fase de documentación, durante los meses de septiembre y octubre, en la que se consultaron diversos artículos científicos [3, 7, 16, 25] para extraer las reglas clave que rigen la organización de partidos a lo largo de la temporada regular.

Una vez recopilada esta información, durante los meses de noviembre y diciembre se procedió al diseño y formulación de un modelo matemático exacto que permitiera representar todas las restricciones del calendario real de la NBA.

Sin embargo, en el mes de febrero, debido a la elevada complejidad computacional del problema y a las dificultades que presentó el modelo exacto para generar soluciones en un tiempo razonable, se optó por un enfoque más práctico, implementando diversas heurísticas diseñadas para generar soluciones aproximadas de forma eficiente.

Con el fin de mejorar aún más la calidad de las soluciones heurísticas, en marzo se aplicaron diferentes técnicas metaheurísticas que permitieron combinar las heurísticas implementadas y explorar de forma más amplia el espacio de soluciones, incrementando así la probabilidad de obtener resultados más cercanos al óptimo.

Posteriormente, en el mes de abril se planteó una reformulación del problema con un enfoque más flexible. Además de distribuir el calendario en 82 jornadas fijas, se consideró la distribución por días, permitiendo extender la temporada a 163 días.

Una vez obtenidos todos los resultados, durante los meses de mayo y junio se inició la evaluación y análisis de los mismos para extraer conclusiones sólidas. Paralelamente, se inició la redacción de la memoria del proyecto con el fin de documentar adecuadamente el trabajo realizado.

1.4. Estructura de la memoria

Antes de abordar en detalle el desarrollo del proyecto, se presenta una descripción formal del problema en el Capítulo 2, donde se exponen los tipos de problemas de programación de calendarios y se explican las reglas que rigen la elaboración del calendario de la NBA. A continuación, en el Capítulo 3 se introducen los modelos matemáticos basados en restricciones que

representan dichas reglas, constituyendo el primer enfoque utilizado para abordar su resolución exacta.

Posteriormente, en los capítulos 4 y 5, se presentan las distintas heurísticas y metaheurísticas desarrolladas a lo largo del proyecto con el objetivo de mejorar la calidad de las soluciones obtenidas, que en este caso serán aproximadas.

Finalmente, se exponen y analizan los resultados experimentales en el Capítulo 6 junto con una serie de reflexiones finales y propuestas de trabajo futuro en el Capítulo 7.

En el Apéndice A se describen los archivos desarrollados a lo largo del proyecto, mientras que en el Apéndice B se explican en detalle las principales estructuras de datos empleadas en la implementación de los diversos algoritmos.

Capítulo 2

Descripción del problema

2.1. Estado del arte

El diseño de calendarios deportivos ha sido objeto de estudio dentro de la investigación operativa y la optimización combinatoria en las últimas décadas, debido a la complejidad que implica equilibrar un gran número de variables y restricciones. A lo largo del tiempo, este problema ha generado un creciente interés académico, particularmente en lo que se refiere a la automatización del proceso y la mejora de su eficiencia.

Dentro de este ámbito, se distinguen dos grandes enfoques de programación [20]:

- **Programación con tiempo restringido:** El número de fechas disponibles es exactamente el necesario para completar todos los partidos requeridos, como sucede en la mayoría de ligas europeas de fútbol. Se puede decir que los calendarios se organizan por jornadas estrictas.
- **Programación con tiempo relajado:** El calendario cuenta con más jornadas de las estrictamente necesarias. Esta flexibilidad adicional permite distribuir mejor la carga de partidos, introducir descansos estratégicos y ajustar el calendario a condicionantes externos. Este es el modelo seguido por ligas como la NBA o la NHL (*National Hockey League*).

Diversos estudios [2, 5, 6, 8, 10, 11, 15, 23, 24] han abordado problemas de programación y optimización de calendarios deportivos en disciplinas como hockey, fútbol americano, cricket, baloncesto, etc. Estos trabajos han empleado una gran variedad de enfoques metodológicos, entre los que destacan principalmente la programación con restricciones y el uso de heurísticas y metaheurísticas.

Más concretamente, en el ámbito de la NBA se han consultado varios estudios [3, 4, 7, 18, 25] que amplían el enfoque aplicado en otros deportes. Sin embargo, existe un número relativamente escaso de artículos específicos que aborden en profundidad la problemática del calendario en la NBA, lo que sugiere la gran complejidad que rodea a este tema. Los artículos encontrados atienden a diversos criterios de optimización. Además de minimizar distancias, también consideran aspectos económicos o de equidad deportiva.

En este proyecto se tratarán ambos enfoques de programación temporal. Por un lado, el enfoque con tiempo restringido, al que se hará referencia como *calendario estructurado*, delimitando la NBA como si fuese una liga basada en 82 jornadas que se juegan a la vez. Por otro lado, se considera el enfoque con tiempo relajado, denominado *calendario no estructurado*, que se adapta más a la realidad, distribuyendo la competición a lo largo de días.

Antes de entrar en detalles, es útil introducir la NBA para conocer un poco mejor cómo se estructura su calendario.

2.2. Introducción a la NBA

La NBA (*National Basketball Association*) [9] es la liga profesional de baloncesto más importante y prestigiosa a nivel mundial. Fundada en 1946 a partir de varias ligas regionales en Estados Unidos, la NBA ha crecido hasta convertirse en un fenómeno global que atrae a millones de aficionados, generando miles de millones de dólares en ingresos y albergando a los mejores jugadores del mundo.

Actualmente, la NBA está compuesta por 30 franquicias, distribuidas principalmente en Estados Unidos, con una sola franquicia en Canadá, tal y como se puede observar en la Figura 2.1. Las franquicias son entidades comerciales que operan como marcas deportivas, propiedad de inversores que deben cumplir con una serie de requisitos estrictos establecidos por la liga para garantizar la viabilidad económica y organizativa del equipo.



Figura 2.1: Distribución de las franquicias NBA en EEUU y Canadá [19].

Estas 30 franquicias están organizadas en dos conferencias: la Conferencia Este y la Conferencia Oeste, cada una formada por 15 equipos. A su vez, cada conferencia se subdivide en tres divisiones, compuestas por 5 equipos cada una. La división se puede observar en detalle en la Figura 2.2.

WESTERN OESTE		EASTERN ESTE	
Noroeste	Pacífico	Atlántico	Central
Suroeste	Sureste		

Figura 2.2: Conferencias y divisiones NBA.

2.3. La estructura del calendario NBA

Uno de los elementos más complejos y críticos en la organización de la NBA es la planificación del calendario de la temporada regular. Cada equipo (franquicia) debe disputar un total de 82 partidos distribuidos a lo largo de aproximadamente 163 días, jugando 41 partidos como local y otros 41 como visitante.

El desglose del calendario es el siguiente:

- **Enfrentamientos dentro de la división:** Cada equipo juega 4 partidos (2 en casa y 2 fuera de casa) contra cada uno de sus 4 oponentes de la división, sumando un total de 16 partidos.
- **Enfrentamientos fuera de la división:** Dentro de la misma conferencia, cada equipo se enfrenta a 6 oponentes de otras divisiones en 4 partidos cada uno (2 en casa y 2 fuera de casa), lo que agrega 24 partidos. Además, juega 3 partidos contra los 4 oponentes restantes (contra dos de ellos jugará 2 partidos en casa y 1 fuera y contra los otros dos jugará 1 partido en casa y 2 fuera), sumando 12 partidos. Los cuatro oponentes fuera de la división a los que se enfrentan tres veces cada uno son determinados por un sistema de rotación de cinco años.
- **Enfrentamientos interconferencia:** Cada equipo juega 2 partidos (1 en casa y 1 fuera de casa) contra los 15 equipos de la conferencia opuesta, completando así 30 partidos.

Esta distribución responde a criterios competitivos, geográficos y de equidad, pero genera una complejidad logística notable.

Con el fin de evaluar la eficacia de los algoritmos desarrollados en este proyecto, se ha tomado como referencia el calendario oficial de la temporada 2022-2023 de la NBA [1]. En concreto, en los calendarios generados a lo largo del proyecto, la clasificación de los rivales de cada equipo según los tipos definidos anteriormente sigue el mismo esquema que en la temporada oficial 2022-2023, replicando así la estructura original de enfrentamientos. Esta temporada ha sido seleccionada por ser la última que se disputó íntegramente bajo el formato tradicional antes de la introducción del nuevo torneo de mitad de temporada, el *Play-In Tournament*, celebrado en Las Vegas durante las últimas dos temporadas. Dado que este torneo introduce una modificación sustancial en la estructura y dinámica del calendario, con partidos que no cuentan para la clasificación de la liga, se ha optado por excluirlo del presente estudio para mantener la coherencia en el modelo de comparación.

Para poder analizar y comparar el rendimiento de los calendarios generados por los algoritmos con respecto al calendario oficial, se ha utilizado como métrica principal la distancia total recorrida por los equipos durante la temporada regular. Esta métrica se genera gracias a un estudio previo [17] que analiza las distancias entre las ciudades de los equipos NBA. A partir de este análisis, se construyó una matriz de constantes denominada **distanciasNBA** que representa las distancias entre las distintas ciudades. Esta matriz se define como:

$$\text{distanciasNBA}[j][j'] \quad \forall j, j' \in N$$

donde **distanciasNBA**[j][j'] indica la distancia (en millas) entre la sede del equipo j y la sede del equipo j' . Esta matriz es simétrica, es decir, cumple:

$$\text{distanciasNBA}[j][j'] = \text{distanciasNBA}[j'][j] \quad \forall j, j' \in N$$

La matriz **distanciasNBA** será empleada en los distintos algoritmos con el objetivo de calcular y minimizar la suma total de millas recorridas por todos los equipos a lo largo de la temporada.

Tras analizar el calendario de la temporada 2022-2023 utilizando esta métrica, se han obtenido varios resultados relevantes que se pueden observar en la Figura 2.3 y que sirven como referencia para este proyecto. En primer lugar, la suma total de millas recorridas por los 30 equipos de la NBA durante toda la temporada regular asciende a 1.465.250.

Al desglosar estos datos por conferencias, se observa que los equipos de la Conferencia Este recorrieron 44.780 millas de media cada uno, lo que resulta en un acumulado total de 671.700 millas. Por otro lado, los equipos de la Conferencia Oeste presentaron una media de desplazamiento significativamente superior, con 52.903 millas de media por equipo, alcanzando un total de 793.549 millas totales. Esta diferencia se debe a la situación geográfica de cada una de estas franquicias. Tal y como se puede observar en la Figura 2.1, las franquicias de la Conferencia Este se encuentran más próximas entre sí que las de la Conferencia Oeste, provocando que los desplazamientos de los equipos de la Conferencia Este sean más cortos que los de la Conferencia Oeste.

A nivel individual, el equipo que recorrió la menor distancia durante la temporada fue Cleveland Cavaliers, con un total de 39.725,9 millas. En el extremo opuesto, los Denver Nuggets fueron el equipo con mayor carga de desplazamientos, acumulando 59.803,9 millas a lo largo de la campaña.

Estas cifras sirven como referencia para comparar de forma objetiva los calendarios que se generen con los algoritmos desarrollados en este proyecto y de esta manera poder observar cuánto se puede mejorar la eficiencia en los desplazamientos.

OESTE		ESTE	
	Equipo	Equipo	Millas recorridas
Nordeste			52194.0
			49054.4
			55411.3
			43363.5
			54931.5
Suroeste			42881.6
			59803.9
			46909.0
			57706.9
			40878.3
Central			49579.8
			41371.1
			52244.0
			39725.9
			44075.6
Suroeste			42217.9
			48013.4
			46970.1
			52017.4
			49479.4
Sudeste			43459.5
			51948.0
			41250.9
			56292.7
			49912.2
Pacífico			45782.8
			55726.9
Millas totales: 793549		Millas totales: 671700	
Millas totales recorridas: 1.465.250			

Figura 2.3: Millas recorridas durante la temporada 2022-2023.

Capítulo 3

Modelo de programación matemática

Con el objetivo de abordar de forma rigurosa la creación de un calendario para la temporada regular de la NBA que cumpliese con las restricciones reales de la competición, el primer enfoque fue el desarrollo de dos modelos de programación matemática.

Los modelos planteados comparten los siguientes conjuntos, fundamentales para representar las reglas específicas del calendario NBA.

3.1. Conjuntos de índices

- N : conjunto de equipos NBA, $|N| = 30$.
- K : conjunto de jornadas del calendario, $|K| = 82$.

Es importante señalar que los conjuntos que se presentan a continuación son subconjuntos del conjunto N y están definidos en función del equipo i considerado:

- $D(i)$: rivales del equipo i dentro de su misma división.
- $C_4(i)$: rivales de la misma conferencia pero distinta división con los que el equipo i juega 4 partidos (2 como local y 2 como visitante).
- $C_{3a}(i)$: rivales de la misma conferencia pero distinta división con los que el equipo i juega 3 partidos, 2 de ellos como local y 1 como visitante.
- $C_{3b}(i)$: rivales de la misma conferencia pero distinta división con los que el equipo i juega 3 partidos, 1 de ellos como local y 2 como visitante.
- $I(i)$: rivales de la conferencia opuesta con los que el equipo i juega 2 partidos (uno como local y otro como visitante).

3.2. Parámetros

En la construcción de un calendario equilibrado, no solo es importante asegurar que cada equipo juegue el mismo número total de partidos como local y como visitante, sino también que dicha distribución sea razonablemente uniforme a lo largo del tiempo. Para ello se introduce el siguiente parámetro:

- Δ_{\max} : diferencia absoluta máxima permitida entre la cantidad acumulada de partidos jugados como local y como visitante por un equipo en cualquier jornada o momento de la temporada.

Este parámetro busca evitar secuencias descompensadas que puedan generar ventajas o desventajas competitivas, como largas rachas de partidos consecutivos como local o como visitante.

3.3. Modelos

Cabe destacar que ambos modelos representan la misma realidad: la programación de los enfrentamientos entre los 30 equipos de la NBA a lo largo de una temporada regular de 82 jornadas. Sin embargo, cada modelo responde a un propósito diferente.

3.3.1. Modelo 1: Generación de un calendario factible

El primer modelo tiene como objetivo principal construir una solución factible, generando un calendario que cumpla todas las reglas impuestas en el Capítulo 2. Este modelo incluye una función objetivo constante, simplemente para asegurar que todas las condiciones del calendario se cumplan, aunque no se pretenda optimizar una métrica concreta.

3.3.1.1. Variables de decisión

Se ha optado por utilizar la variable binaria x_{ijk} que se define de la siguiente manera:

$$x_{ijk} = \begin{cases} 1, & \text{si el equipo } i \text{ juega como local contra el equipo } j \text{ en la jornada } k \\ 0, & \text{en caso contrario} \end{cases}$$

donde:

- $i \in N$: equipo local
- $j \in N$, con $j \neq i$: equipo visitante
- $k \in K$: jornada del calendario

3.3.1.2. Función objetivo

Tal como se ha mencionado anteriormente, este modelo no persigue la optimización de una métrica concreta, sino que tiene como objetivo principal encontrar una solución factible que satisfaga todas las restricciones estructurales del calendario NBA.

Por esta razón, se incorpora una función objetivo constante, que permite formalizar el modelo en el marco de la programación entera, sin afectar al proceso de resolución:

$$\min c \tag{3.1}$$

donde c representa una constante sin significado práctico, utilizada únicamente para completar la formulación matemática del modelo.

3.3.1.3. Restricciones

- Cada equipo i juega un único partido por jornada, que puede ser o bien como local, o bien como visitante:

$$\sum_{j \in N \setminus \{i\}} (x_{ijk} + x_{jik}) = 1 \quad \forall i \in N, \forall k \in K \tag{3.2}$$

- Cada equipo i juega 4 partidos contra los rivales de su misma división, 2 como local y 2 como visitante:

$$\sum_{k \in K} x_{ijk} = 2, \quad \sum_{k \in K} x_{jik} = 2 \quad \forall i \in N, \forall j \in D(i) \quad (3.3)$$

- Cada equipo i juega 3 o 4 partidos contra equipos de su misma conferencia pero pertenecientes a una división diferente. Se establecen 3 situaciones:

- El equipo i juega 4 partidos contra un equipo rival perteneciente al conjunto $C_4(i)$, 2 como local y 2 como visitante:

$$\sum_{k \in K} x_{ijk} = 2, \quad \sum_{k \in K} x_{jik} = 2 \quad \forall i \in N, \forall j \in C_4(i) \quad (3.4)$$

- El equipo i juega 3 partidos contra un equipo rival perteneciente al conjunto $C_{3a}(i)$, 2 como local y 1 como visitante:

$$\sum_{k \in K} x_{ijk} = 2, \quad \sum_{k \in K} x_{jik} = 1 \quad \forall i \in N, \forall j \in C_{3a}(i) \quad (3.5)$$

- El equipo i juega 3 partidos contra un equipo rival perteneciente al conjunto $C_{3b}(i)$, 1 como local y 2 como visitante:

$$\sum_{k \in K} x_{ijk} = 1, \quad \sum_{k \in K} x_{jik} = 2 \quad \forall i \in N, \forall j \in C_{3b}(i) \quad (3.6)$$

- Cada equipo i juega 2 partidos contra rivales de la conferencia contraria, 1 como local y 1 como visitante:

$$\sum_{k \in K} x_{ijk} = 1, \quad \sum_{k \in K} x_{jik} = 1 \quad \forall i \in N, \forall j \in I(i) \quad (3.7)$$

- Para favorecer un equilibrio en el calendario, se impone que la diferencia máxima para cada equipo i entre partidos jugados en casa y fuera de casa en una jornada determinada esté acotada por Δ_{\max} :

$$\sum_{j \in N \setminus \{i\}} \sum_{t=1}^k x_{ijt} - \sum_{j \in N \setminus \{i\}} \sum_{t=1}^k x_{jbt} \leq \Delta_{\max} \quad \forall i \in N, \forall k \in K \quad (3.8)$$

$$\sum_{j \in N \setminus \{i\}} \sum_{t=1}^k x_{jbt} - \sum_{j \in N \setminus \{i\}} \sum_{t=1}^k x_{ijt} \leq \Delta_{\max} \quad \forall i \in N, \forall k \in K \quad (3.9)$$

donde se recuerda que Δ_{\max} representa el límite máximo permitido de diferencia acumulada entre partidos como local y visitante.

Este modelo principalmente garantiza que cada equipo juegue un único partido por jornada, respetando el número de enfrentamientos específicos frente a cada tipo de rival y distribuyendo correctamente los partidos como local y visitante. Es esencial como primer paso para comprobar que existe una solución que respete todas las condiciones estructurales del calendario.

3.3.2. Modelo 2: Minimización de distancias

Una vez asegurada la viabilidad estructural mediante el primer modelo, se plantea un segundo modelo que incorpora una función objetivo orientada a minimizar la distancia total recorrida por los equipos durante la temporada. Para ello se utilizará la matriz `distanciasNBA` presentada en el capítulo anterior, que recoge la distancia en millas entre cualquier par de franquicias.

3.3.2.1. Variables de decisión

Con respecto a las variables involucradas, se sustituyen las variables x_{ijk} del primer modelo por las siguientes variables binarias:

$$y_{ijk} = \begin{cases} 1, & \text{si el equipo } i \text{ se encuentra (para jugar) en la ciudad del equipo } j \text{ en la jornada } k \\ 0, & \text{en caso contrario} \end{cases}$$

donde:

- $i \in N$: equipo en concreto
- $j \in N$: ciudad en la que se encuentra el equipo i
- $k \in K$: jornada del calendario

y

$$z_{ijj'k} = \begin{cases} 1, & \text{si el equipo } i \text{ viaja de la ciudad del equipo } j \text{ a la ciudad del equipo } j' \\ & \text{entre las jornadas } k \text{ y } k+1 \\ 0, & \text{en caso contrario} \end{cases}$$

donde:

- $i \in N$: equipo que realiza el desplazamiento
- $j, j' \in N$: ciudades que se toman como origen y destino
- $k \in \{0, \dots, |K| - 1\}$: jornada del calendario

3.3.2.2. Función Objetivo

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{j' \in N} \sum_{k=0}^{|K|-1} \text{distanciasNBA}_{jj'} \cdot z_{ijj'k} \quad (3.10)$$

El objetivo ahora es minimizar la suma total de las distancias recorridas por todos los equipos durante la temporada, basándose en las distancias entre ciudades recogidas en la matriz `distanciasNBA`.

3.3.2.3. Restricciones

- Cada equipo i solo puede estar en una ciudad por jornada:

$$\sum_{j \in N} y_{ijk} = 1 \quad \forall i \in N, \forall k \in K \quad (3.11)$$

- Solo puede haber como máximo un equipo distinto del local en esa misma ciudad j por jornada:

$$\sum_{i \in N \setminus \{j\}} y_{ijk} \leq 1 \quad \forall j \in N, \forall k \in K \quad (3.12)$$

- Las siguientes restricciones definen los viajes que realiza cada equipo:

$$z_{ijj'k} \geq y_{ijk} + y_{ij'k+1} - 1 \quad \forall i, j, j' \in N, \forall k \in \{0, \dots, |K| - 1\} \quad (3.13)$$

$$z_{ijj'k} \leq y_{ijk} \quad \forall i, j, j' \in N, \forall k \in \{0, \dots, |K| - 1\} \quad (3.14)$$

$$z_{ijj'k} \leq y_{ij'k+1} \quad \forall i, j, j' \in N, \forall k \in \{0, \dots, |K| - 1\} \quad (3.15)$$

- Cada equipo i juega 4 partidos contra los rivales de su misma división, 2 como local y 2 como visitante:

$$\sum_{k \in K} y_{jik} = 2, \quad \sum_{k \in K} y_{ijk} = 2 \quad \forall i \in N, \forall j \in D(i) \quad (3.16)$$

- Cada equipo i juega 3 o 4 partidos contra equipos de su misma conferencia pero de diferente división. Se establecen 3 situaciones:

- El equipo i juega 4 partidos contra un equipo rival perteneciente al conjunto $C_4(i)$, 2 como local y 2 como visitante:

$$\sum_{k \in K} y_{jik} = 2, \quad \sum_{k \in K} y_{ijk} = 2 \quad \forall i \in N, \forall j \in C_4(i) \quad (3.17)$$

- El equipo i juega 3 partidos contra un equipo rival perteneciente al conjunto $C_{3a}(i)$, 2 como local y 1 como visitante:

$$\sum_{k \in K} y_{jik} = 2, \quad \sum_{k \in K} y_{ijk} = 1 \quad \forall i \in N, \forall j \in C_{3a}(i) \quad (3.18)$$

- El equipo i juega 3 partidos contra un equipo rival perteneciente al conjunto $C_{3b}(i)$, 1 como local y 2 como visitante:

$$\sum_{k \in K} y_{jik} = 1, \quad \sum_{k \in K} y_{ijk} = 2 \quad \forall i \in N, \forall j \in C_{3b}(i) \quad (3.19)$$

- Cada equipo i juega 2 partidos contra rivales de la conferencia contraria, 1 como local y 1 como visitante:

$$\sum_{k \in K} y_{jik} = 1, \quad \sum_{k \in K} y_{ijk} = 1 \quad \forall i \in N, \forall j \in I(i) \quad (3.20)$$

- Para garantizar que los partidos se disputen solo si ambos equipos están presentes en la ciudad correspondiente, se imponen las siguientes restricciones de consistencia espacial:

$$y_{jik} \leq y_{iik} \quad \forall i, j \in N, \forall k \in K \quad (3.21)$$

Esta restricción asegura que si el equipo j juega como visitante en la ciudad del equipo i en la jornada k , entonces el equipo i debe estar en su propia ciudad para jugar el partido en dicha jornada k .

$$y_{ijk} \leq y_{jjk} \quad \forall i, j \in N, \forall k \in K \quad (3.22)$$

Esta restricción garantiza que si el equipo i juega como visitante en la ciudad del equipo j en la jornada k , entonces el equipo j debe estar en su ciudad en la jornada k para disputar ese partido.

- Al igual que en el Modelo 1, para favorecer un equilibrio en el calendario, se impone que la diferencia máxima entre partidos jugados en casa y fuera de casa por equipo en una jornada determinada esté acotada por Δ_{\max} :

$$\sum_{t=1}^k y_{iit} - \sum_{\substack{j \in N \\ j \neq i}} \sum_{t=1}^k y_{ijt} \leq \Delta_{\max} \quad \forall i \in N, \forall k \in K \quad (3.23)$$

$$\sum_{\substack{j \in N \\ j \neq i}} \sum_{t=1}^k y_{ijt} - \sum_{t=1}^k y_{iit} \leq \Delta_{\max} \quad \forall i \in N, \forall k \in K \quad (3.24)$$

donde se recuerda que Δ_{\max} representa el límite máximo permitido de diferencia acumulada entre partidos como local y visitante.

Además de las restricciones estructurales con respecto al número de partidos que se juegan contra cada tipo de rival como en el primer modelo, se añaden otras para controlar que cada equipo esté en una sola ciudad por jornada, limitar a un único visitante en cada ciudad y definir los desplazamientos por jornada en función de la localización en la que se encuentre cada equipo.

Observación 1. Cabe destacar que los modelos presentados están formulados para una liga compuesta por 30 equipos y un total de 82 jornadas. No obstante, también se han considerado versiones reducidas con 4 y 8 equipos, como se estudiará más adelante. Para estas ligas simplificadas, se han adaptado las condiciones correspondientes, eliminando algunas restricciones relacionadas con los tipos de partidos y ajustando los conjuntos involucrados. Además, debido al escaso número de jornadas en estas ligas reducidas, no se ha tenido en cuenta la restricción relativa a la diferencia entre partidos disputados como local y como visitante, relacionada con Δ_{\max} .

Observación 2. En la implementación de estos modelos, se ha considerado que todos los equipos comienzan y finalizan sus respectivos recorridos en su ciudad de origen. Para representar esta condición, a la hora de implementar se han incorporado dos jornadas adicionales ficticias, una al comienzo y otra al final del calendario, que actúan como jornadas fantasma, permitiendo cerrar el recorrido de cada equipo de forma coherente dentro del modelo.

Como se discutirá en el Capítulo 6, las limitaciones computacionales han condicionado la aplicabilidad práctica del segundo modelo, motivando el uso de técnicas heurísticas y meta-heurísticas que se desarrollarán en los capítulos posteriores.

Capítulo 4

Heurísticas

Debido a la elevada complejidad computacional y al extenso espacio de búsqueda que planteaba el modelo matemático exacto, este enfoque resultó inviable para obtener una solución óptima en tiempos razonables. Ante esta limitación, se optó por la implementación de diferentes heurísticas que permitieron obtener soluciones aproximadas de forma eficiente.

Se han desarrollado cuatro heurísticas distintas, cada una centrada en un tipo específico de modificación del calendario. Tres de ellas operan sobre calendarios estructurados por jornadas, lo que corresponde a una programación con tiempo restringido. La cuarta, en cambio, actúa sobre calendarios no estructurados de 163 días, lo que implica una programación con tiempo relajado. Estas heurísticas permiten realizar ajustes progresivos sobre el calendario con el objetivo de reducir la distancia total recorrida por los equipos.

En todas las heurísticas, los movimientos se aplican en cuanto se detecta una mejora en la solución. Es decir, se prioriza la mejora inmediata frente a la búsqueda de la mejor posible ya que esta estrategia tiene un coste computacional significativamente menor. Esta eficiencia es clave para su posterior integración en metaheurísticas, donde se requiere rapidez en los algoritmos.

Además, todas estas heurísticas han sido diseñadas para garantizar que, tras cada modificación realizada, se conserve el equilibrio entre partidos disputados como local y como visitante (excepto en las ligas reducidas de 4 y 8 equipos). En particular, se ha respetado la restricción que impide una diferencia superior a Δ_{\max} partidos entre ambos tipos de encuentros para cualquier equipo en cualquier momento de la temporada.

4.1. Heurísticas basadas en calendarios estructurados

En primer lugar, se presentan las tres primeras heurísticas que mantienen la distribución del calendario por jornadas.

4.1.1. Heurística 1: Generación de múltiples calendarios

Esta heurística se basa en una estrategia de generación múltiple de soluciones mediante la resolución repetida del Modelo 1 utilizando el optimizador Gurobi (herramienta que se introducirá en el Capítulo 6). En cada ejecución, Gurobi resuelve el modelo con pequeñas variaciones, lo que permite obtener calendarios factibles distintos en cada iteración. Cada una de estas soluciones representa un calendario completo que cumple con todas las restricciones impuestas por el modelo.

Para fomentar la diversidad en las soluciones generadas, se reinicia la semilla del generador de números aleatorios de Gurobi en cada ejecución, asegurando que la configuración interna del optimizador varíe en cada llamada, permitiendo explorar diferentes regiones del espacio de soluciones factibles y evitando repetir la misma solución.

Una vez generada cada solución, se evalúa su calidad en función de la distancia total recorrida por los equipos a lo largo de la temporada. El proceso se repite varias veces y se escoge el calendario que presenta la menor distancia total recorrida entre todos los calendarios generados. El pseudocódigo que resume el funcionamiento de esta heurística se presenta en el Algoritmo 1.

Algoritmo 1 Heurística 1: Generación de múltiples calendarios

```

1: Inicializar mejor calendario vacío y mejor distancia como infinito
2: for cada ejecución del optimizador do
3:   Ejecutar Modelo 1 con Gurobi (reiniciando semilla) para obtener un calendario factible
4:   Calcular distancia total del calendario generado
5:   if distancia es menor que mejor distancia then
6:     Actualizar mejor calendario y mejor distancia
7:   end if
8: end for
9: return mejor calendario y mejor distancia encontrada

```

4.1.2. Heurística 2: Intercambios de jornadas

Para comprender en profundidad el funcionamiento de esta heurística, es fundamental explicar el movimiento básico que emplea para modificar el calendario:

- **Intercambio de jornadas:** Este movimiento consiste en intercambiar dos jornadas completas entre sí, manteniendo los mismos enfrentamientos pero alterando el orden temporal en el que se disputan, tal y como se puede observar en la Figura 4.1. De este modo se pueden reducir desplazamientos innecesarios sin modificar la estructura global del calendario.



Figura 4.1: Intercambio de jornadas.

Partiendo de este movimiento, la heurística explora exhaustivamente todos los pares de jornadas para analizar si intercambiar su posición contribuye a disminuir la distancia total recorrida por los equipos a lo largo de la temporada regular y respeta el cumplimiento de la condición Δ_{\max} . Cada vez que se encuentra un intercambio que mejora la solución, el algoritmo se reinicia, repitiéndose este proceso hasta que no sea posible mejorar más el calendario. Se puede consultar el pseudocódigo de esta heurística en el Algoritmo 2.

Algoritmo 2 Heurística 2: Intercambios de jornadas

```

1: Considerar la distancia total del calendario correspondiente a la solución inicial
2: repeat
3:   for all pares de jornadas ( $k_1, k_2$ ) con  $k_1 < k_2$  do
4:     Estimar la distancia resultante si se intercambian  $k_1$  y  $k_2$ 
5:     if el intercambio reduce la distancia y respeta la condición  $\Delta_{\max}$  then
6:       Realizar el intercambio de jornadas
7:       Actualizar la distancia total
8:       Reiniciar la búsqueda desde el inicio
9:     break
10:   end if
11: end for
12: until no se producen mejoras en el calendario

```

4.1.3. Heurística 3: Intercambios local-visitante

Siguiendo el mismo esquema que en la heurística anterior, se presenta primero el movimiento clave en el que se basa esta heurística:

- **Intercambio de partido local-visitante:** En este caso, se modifica la condición de localía en un determinado encuentro. Si un equipo disputa dos partidos contra otro (uno como local y otro como visitante), se evalúa la posibilidad de intercambiar estos partidos con el objetivo de optimizar las rutas de viajes y reducir la distancia global acumulada. En la Figura 4.2 se puede observar el cambio local-visitante entre un partido que involucra a dos equipos en las jornadas 21 y 54.

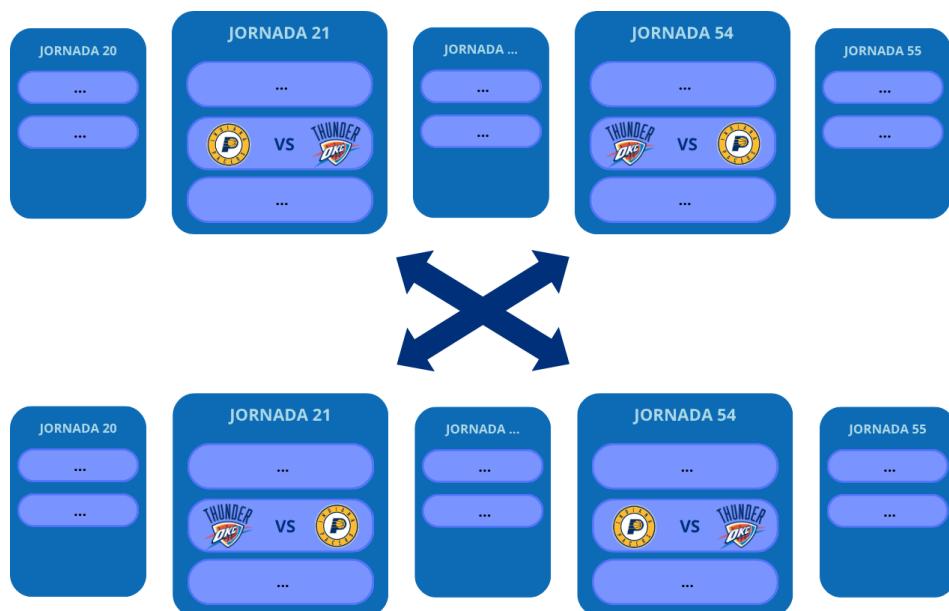


Figura 4.2: Intercambio local-visitante de un partido.

Tal como se observa en el Algoritmo 3, se recorren todos los partidos jornada a jornada y se identifica el partido local-visitante opuesto que involucra a los mismos equipos y que permite una mayor reducción de la distancia total manteniendo la condición Δ_{\max} . Cada vez que se realiza un cambio, la búsqueda se reinicia desde el principio, garantizando que al finalizar no existan más intercambios que puedan mejorar la solución.

Algoritmo 3 Heurística 3: Intercambios local-visitante

```

1: Considerar la distancia total del calendario correspondiente a la solución inicial
2: repeat
3:   for all jornadas del calendario do
4:     for all parejas de equipos  $(i, j)$  do
5:       if existe un partido entre  $i$  y  $j$  en la jornada actual then
6:         Buscar mejor intercambio equivalente de localía
7:         if intercambiar localía reduce la distancia y respeta la condición  $\Delta_{\max}$  then
8:           Realizar el intercambio local-visitante
9:           Actualizar la distancia total
10:          Reiniciar la búsqueda desde el inicio
11:          break bucles
12:        end if
13:      end if
14:    end for
15:  end for
16: until no se producen mejoras en el calendario

```

No se han podido desarrollar más heurísticas para este tipo de calendario. Esto se debe a que las posibles modificaciones son muy restringidas, ya que es necesario preservar la estructura del calendario, como el número de partidos de cada tipo y la condición de que todos los equipos jueguen un partido en cada jornada, lo que limita considerablemente las opciones de ajuste.

Una vez introducidas las heurísticas que trabajan con el calendario estructurado, se pasa a describir la heurística relacionada con el calendario organizado por días (no estructurado).

4.2. Heurísticas basadas en calendarios no estructurados

Con el objetivo de superar las limitaciones impuestas por la estructura fija de 82 jornadas, se planteó un nuevo enfoque que permite distribuir los partidos a lo largo de 163 días, manteniendo los 82 encuentros por equipo, pero sin la necesidad de que todos los equipos jueguen cada jornada. Esta mayor flexibilidad permite una reducción más significativa de las distancias recorridas. Para ello se implementó el siguiente algoritmo de inserción.

4.2.1. Heurística 4: Inserción

La lógica de este algoritmo consiste en recorrer los partidos uno a uno y explorar en qué posición del calendario pueden ubicarse de forma que minimicen el impacto en la distancia total. Este algoritmo se basa en el siguiente movimiento:

- **Reubicación de un partido:** Este movimiento consiste en seleccionar un partido del calendario y buscar otro día en el que ninguno de los dos equipos implicados tenga programado otro encuentro, asegurando además que el cambio reduzca la distancia total recorrida.

Dado que el calendario está organizado por días (en lugar de jornadas fijas), la reubicación debe respetar una serie de restricciones adicionales para garantizar la viabilidad del calendario modificado y el equilibrio en el avance de cada equipo en la competición:

- Cada equipo puede descansar un máximo de 4 días consecutivos.
- Cada equipo puede disputar un máximo de 2 partidos en días consecutivos.
- La diferencia de partidos jugados entre dos equipos cualesquiera en cualquier momento de la temporada no puede superar los 4 partidos.

Además, se mantiene la condición previamente impuesta sobre el equilibrio entre partidos como local y como visitante para cada equipo con una diferencia máxima permitida de $\Delta_{\text{máx}}$ partidos en cualquier momento de la temporada.

Este tipo de movimiento puede visualizarse gráficamente en la Figura 4.3, donde se reubica un partido programado para el día 21 (color rojo) y se inserta en el día 92 (color verde). Se puede ver cómo tras insertarse en el día 92 (color verde), el partido desaparece del día 21.



Figura 4.3: Reubicación de un partido.

Una vez presentado el movimiento principal, la heurística se basa principalmente en recorrer todos los partidos del calendario y, para cada uno de ellos, comprobar si se puede insertar en otro día de modo que se minimice la distancia total recorrida. El pseudocódigo del algoritmo se presenta en el Algoritmo 4.

Este enfoque más flexible permite relajar ciertas restricciones temporales y adaptar mejor la planificación, manteniendo al mismo tiempo un calendario competitivo y equilibrado para todos los equipos por igual.

Algoritmo 4 Heurística 4: Inserción

1: Considerar la distancia total del calendario correspondiente a la solución inicial
2: **repeat**
3: **for all** jornadas k_1 **do**
4: **for all** equipos i **do**
5: **if** equipo i tiene partido en jornada k_1 **then**
6: Obtener *local* y *visitante* del partido en la jornada k_1
7: **for all** jornadas $k_2 \neq k_1$ **do**
8: **if** *local* y *visitante* no tienen partido en k_2 **then**
9: Estimar la mejora de distancia al mover el partido de k_1 a k_2
10: **if** la mejora es positiva y respeta las condiciones de equilibrio **then**
11: Mover el partido de k_1 a k_2
12: Actualizar la distancia total
13: **break**
14: **end if**
15: **end if**
16: **end for**
17: **end if**
18: **end for**
19: **end for**
20: **until** no se producen mejoras en el calendario

Capítulo 5

Metaheurísticas

Con el objetivo de mejorar la calidad de las soluciones obtenidas a partir de las heurísticas básicas descritas en el capítulo anterior, se han implementado tres metaheurísticas que permiten una exploración más eficiente del espacio de soluciones. Estas técnicas están diseñadas para escapar de óptimos locales y encontrar soluciones más cercanas al óptimo global, aunque no sea posible probar la optimalidad de la mejor solución alcanzada.

En la Figura 5.1, se puede decir que partiendo de la ‘Posición actual’, las heurísticas aplicadas en el capítulo anterior hacían que solo se explorasen las inmediaciones de esa posición, llegando a mínimos locales. Sin embargo, la idea de estas metaheurísticas es explorar el espacio de soluciones factibles para alcanzar el mínimo global.

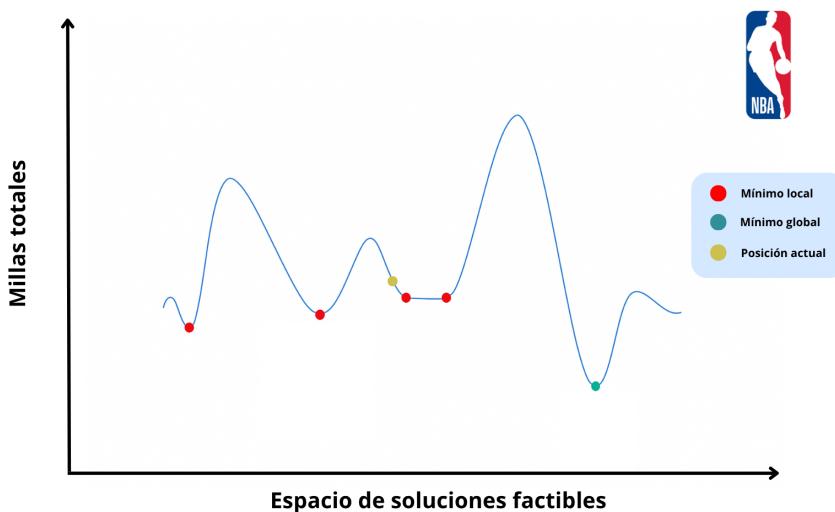


Figura 5.1: Espacio de soluciones.

En las tres metaheurísticas se han incorporado tres de las heurísticas previamente definidas: el intercambio íntegro de jornadas, el intercambio de partidos local-visitaante y la inserción de partidos.

5.1. Temple simulado

El Temple Simulado (*Simulated Annealing*) [21, 22] es una metaheurística de naturaleza probabilística propuesta originalmente por Kirkpatrick, Gelett y Vecchi en 1983. Esta técnica está inspirada en el proceso físico de enfriamiento lento de metales, conocido como recocido.

El algoritmo simula este proceso físico permitiendo, especialmente en las primeras etapas de ejecución, aceptar soluciones que empeoran el valor actual de la función objetivo. Esta aceptación va disminuyendo progresivamente a lo largo del tiempo, imitando el descenso gradual de temperatura en el enfriamiento de un metal, tal y como se puede observar en la Figura 5.2. De esta manera, el algoritmo evita quedar atrapado prematuramente en óptimos locales y favorece la exploración del espacio de soluciones.

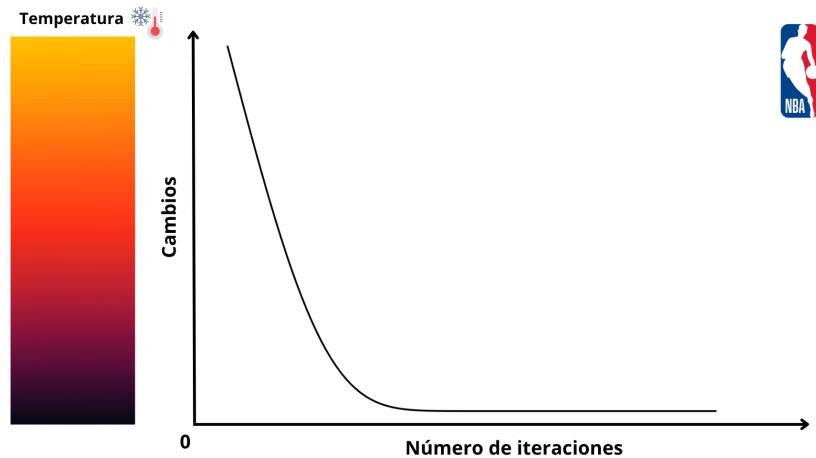


Figura 5.2: Metaheurística Temple Simulado.

El algoritmo implementado (pseudocódigo en el Algoritmo 5) parte de la solución obtenida mediante el Modelo 1. Se comienza con una temperatura inicial, que a lo largo del proceso iterativo irá disminuyendo progresivamente hasta alcanzar un umbral mínimo, conocido como temperatura final. El ritmo con el que la temperatura disminuye se controla mediante un factor de enfriamiento denominado α , que multiplica la temperatura al final de cada iteración. Cuanto más cercano a 1 sea este valor, más lento será el enfriamiento y por tanto, mayor será la exploración del espacio de soluciones. Tanto la temperatura inicial, la temperatura final y el factor de enfriamiento son parámetros del algoritmo que permitirán probar distintas configuraciones para obtener la mejor solución posible.

Durante cada iteración, se generan modificaciones sobre el calendario actual. Cabe destacar que estas modificaciones preservan el cumplimiento en todo momento de las restricciones explícadas en el Capítulo 4 en cuanto al equilibrio del calendario. La cantidad de cambios depende directamente de la temperatura: cuando la temperatura es elevada, el sistema permite más modificaciones, lo que conlleva una mayor exploración. Sin embargo, a temperaturas más bajas, se restringe la cantidad de cambios, favoreciendo una búsqueda más local y conservadora. Estas modificaciones consisten en operaciones como intercambios de jornadas, cambios de partidos entre local y visitante o inserción de partidos, que se corresponden con los presentados en el Capítulo 4.

Una de las características fundamentales del Temple Simulado es que acepta soluciones que pueden empeorar el valor de la función objetivo, es decir, soluciones que aumenten la distancia recorrida que se desea minimizar. Esto se hace deliberadamente para evitar quedar atrapado en óptimos locales y fomentar una exploración más amplia del espacio de búsqueda.

Una vez aplicados los cambios, se procede a una fase de mejora local, en la cual se emplea alguna de las heurísticas (2, 3 o 4) descritas con anterioridad, intentando reducir la distancia recorrida en el calendario. Al finalizar esa fase, se evalúa la calidad de la solución obtenida comparándola con la mejor solución conocida hasta el momento. Si la nueva solución resulta ser mejor, se actualiza como la mejor solución global encontrada.

El proceso se repite partiendo desde esta nueva solución en la siguiente iteración hasta que la temperatura desciende por debajo del umbral mínimo establecido. Finalmente, el algoritmo devuelve el mejor calendario generado, junto con su correspondiente distancia recorrida en millas.

Algoritmo 5 Temple Simulado

```

1: Inicializar mejor solución con calendario y distancia
2: while la temperatura no ha alcanzado el mínimo do
3:   Ajustar el número de movimientos por iteración según la temperatura
4:   for cierto número de movimientos do
5:     Realizar movimiento aleatorio en el calendario (2, 3 o 4)
6:     Actualizar la distancia total
7:   end for
8:   Aplicar mejora local (heurísticas 2, 3 o 4) hasta que la solución no pueda mejorar más.
9:   if calendario actual recorre menos distancia que el mejor calendario then
10:    Actualizar la mejor solución
11:    Actualizar la mejor distancia
12:  end if
13:  Reducir la temperatura con un factor de enfriamiento
14: end while
15: return mejor calendario y distancia encontrada

```

5.2. Búsqueda por Entornos Variables

La metaheurística VNS (*Variable Neighborhood Search*) [13, 14] fue propuesta por primera vez por Nenad Mladenović y Pierre Hansen, ambas figuras destacadas en el campo de la Investigación Operativa. La idea fue presentada inicialmente durante la conferencia *Optimization Days 1995* en la Universidad de Montreal, pero no fue hasta 1997 cuando se formalizó este método, el cual se ha consolidado como una técnica eficaz para abordar problemas de optimización combinatoria.

La búsqueda por entornos variables se basa en la idea de que explorar diferentes estructuras de vecindad puede ser más efectivo que permanecer en una única zona del espacio de soluciones. A través de esta estrategia, VNS logra evitar el estancamiento en óptimos locales, tal y como se puede observar en la Figura 5.3. A diferencia de otros métodos que exploran la misma vecindad, VNS cambia dinámicamente la cantidad de movimientos aplicados a la solución actual dependiendo del vecindario en el que se encuentre, lo que permite una exploración más rica del espacio de soluciones.

El algoritmo descrito como pseudocódigo en el Algoritmo 6 parte de nuevo de la solución obtenida mediante el Modelo 1, al cual se le va a aplicar un proceso de mejora iterativa. La estructura general de este algoritmo se compone de dos bucles anidados. El más externo controla el número máximo de iteraciones globales y el número de iteraciones sin mejora consecutivas. El interno gestiona las distintas vecindades que se irán explorando. Durante cada iteración principal, se recorre un conjunto de vecindades numeradas. Cada vecindad representa un tipo distinto de modificación que se aplica a la solución actual. En particular, se define para cada nivel de vecindad un número de cambios distintos que indican cuántas modificaciones se aplican en esa fase.

En cada vecindad, se aplican los cambios que coinciden con el intercambio de jornadas, el cambio local-visitante o la inserción de un partido, los cuales alteran la estructura del calendario y su distancia total. Cabe destacar que, al igual que con el Temple Simulado, estas modificaciones preservan el cumplimiento en todo momento de las restricciones descritas en el Capítulo 4 en lo que se refiere al equilibrio del calendario. Tras realizar los cambios, se ejecuta la fase de mejora local aplicando alguna de las heurísticas descritas anteriormente (2, 3 o 4).

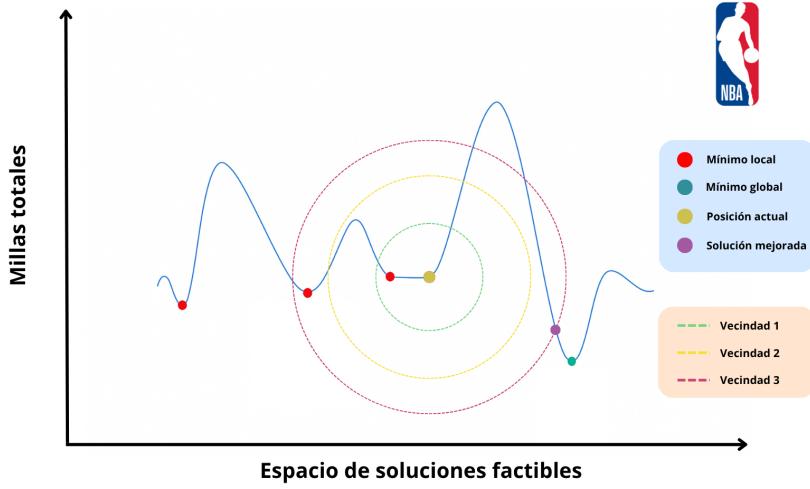


Figura 5.3: Metaheurística VNS.

Una vez finalizada la mejora local, se evalúa si la solución actual supera en calidad a la mejor solución obtenida hasta ese momento. En caso afirmativo, se actualiza tanto la mejor distancia como el mejor calendario, se reinicia el contador de iteraciones sin mejora y se vuelve a empezar desde la primera vecindad. Si no se produce mejora, se incrementa el nivel de vecindad para intentar escapar del óptimo local con una modificación mayor. Si ninguna de las vecindades genera una mejora, se incrementa el contador de iteraciones sin mejora. Este proceso se repite hasta alcanzar el número máximo de iteraciones o hasta superar el límite de iteraciones sin mejora. Es importante resaltar que en todas las iteraciones se parte del mejor calendario obtenido hasta el momento.

Finalmente, el algoritmo devuelve el mejor calendario encontrado junto con su distancia correspondiente en millas. Cabe destacar que tanto el número de vecindades, como los cambios por vecindad, el número de iteraciones máximas sin mejora y el número de iteraciones máximas actúan como parámetros del algoritmo para poder probar diferentes configuraciones hasta alcanzar la mejor solución.

Algoritmo 6 Búsqueda por Entornos Variables (VNS)

```

1: Inicializar mejor solución con calendario y distancia
2: Inicializar contador de iteraciones sin mejora
3: while no se supera límite de iteraciones sin mejora ni máximo de iteraciones do
4:   Inicializar la solución actual con la mejor solución encontrada hasta el momento
5:   Inicializar vecindad  $k \leftarrow$  primera vecindad
6:   while existen vecindades por explorar ( $k \leq \text{Num\_Vecindades}$ ) do
7:     Generar solución vecina aplicando cambios aleatorios (2, 3 o 4) según dicte  $k$ 
8:     Aplicar mejora local (heurísticas 2, 3 o 4) para optimizar la solución
9:     if la solución mejora la mejor encontrada then
10:       Actualizar mejor solución, reiniciar contador de iteraciones sin mejora y vecindad
11:     else
12:       Avanzar a la siguiente vecindad
13:     end if
14:   end while
15:   Incrementar contador de iteraciones sin mejora
16: end while
17: return mejor calendario y distancia encontrada

```

5.3. Búsqueda Tabú

La Búsqueda Tabú (*Tabu Search*) [12] es una metaheurística introducida a mediados de los años 80 por Fred Glover, formalizada posteriormente en 1989. Se caracteriza por el uso de estructuras de memoria que orientan la búsqueda hacia regiones prometedoras del espacio de soluciones, evitando retrocesos y ciclos innecesarios.

Su funcionamiento se basa en una búsqueda local intensiva, donde los movimientos recientes se almacenan en una estructura llamada lista tabú. Esta memoria a corto plazo, impide que el algoritmo repita acciones previamente realizadas durante un número determinado de iteraciones, promoviendo así una mayor diversidad en la exploración. Un ejemplo de este mecanismo puede observarse en la Figura 5.4, donde se intenta efectuar un movimiento previamente realizado; al encontrarse en la lista tabú, dicho movimiento se considera inválido y no se lleva a cabo.

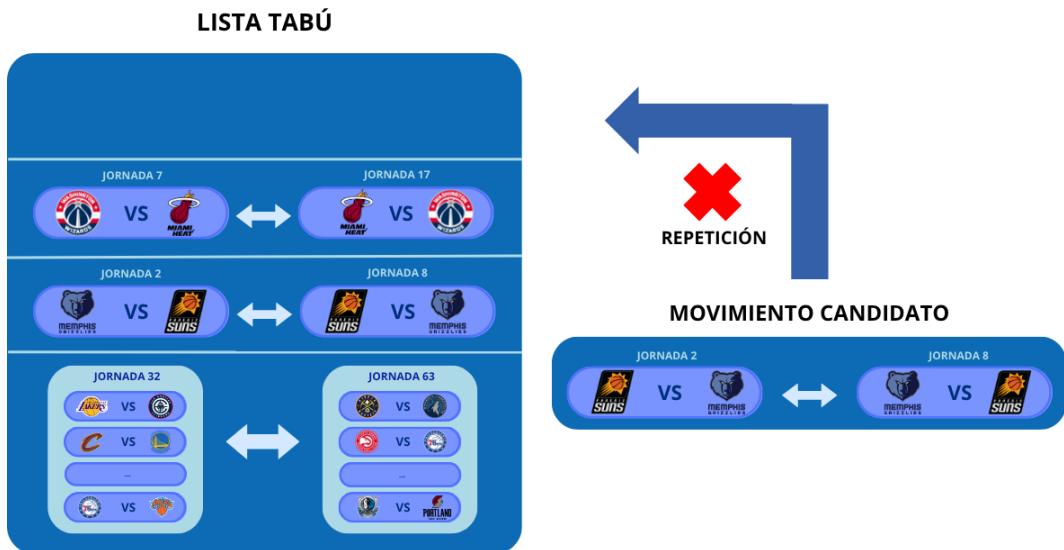


Figura 5.4: Metaheurística Búsqueda Tabú.

El algoritmo (descrito mediante el pseudocódigo del Algoritmo 7) parte de una solución inicial generada por el Modelo 1. En cada iteración evalúa todas las posibles modificaciones definidas por las heurísticas 2, 3 o 4, calculando el impacto de cada una sobre la distancia total recorrida. El mejor movimiento factible, que se corresponde con el que no está restringido por la lista tabú y representa la mayor mejora, es seleccionado y aplicado al calendario. Al igual que ocurría en el resto de metaheurísticas, los cambios se aseguran de preservar las restricciones de equilibrio del calendario.

Tras realizar el cambio, se actualiza la distancia recorrida y se incorpora el movimiento a la lista tabú. Si esta ha alcanzado su capacidad máxima, se elimina el elemento más antiguo siguiendo la política FIFO (*First-in, First-out*), garantizando así una rotación constante de movimientos prohibidos.

Al finalizar cada iteración, se compara la solución actual con la mejor solución obtenida hasta el momento. Si se ha producido mejora, se actualiza el calendario óptimo. Este proceso se repite hasta que se alcanza el número máximo de iteraciones, definido como parámetro del algoritmo, al igual que el tamaño de la lista tabú.

Finalmente, el procedimiento devuelve el mejor calendario encontrado junto con la distancia total recorrida en millas.

Algoritmo 7 Búsqueda Tabú

- 1: Inicializar mejor solución con calendario y distancia
 - 2: Inicializar lista tabú vacía
 - 3: **for** cada iteración hasta máximo permitido **do**
 - 4: Explorar vecindad para encontrar el mejor movimiento no tabú
 - 5: Actualizar solución con el mejor vecino encontrado
 - 6: Actualizar lista tabú con el movimiento realizado
 - 7: Mantener tamaño de lista tabú dentro del límite
 - 8: **if** mejora la mejor solución **then**
 - 9: Actualizar mejor solución encontrada
 - 10: **end if**
 - 11: **end for**
 - 12: **return** mejor calendario y distancia encontrada
-

Capítulo 6

Resultados

Con el objetivo de evaluar el rendimiento de los distintos métodos desarrollados a lo largo de este proyecto, este capítulo recoge los resultados obtenidos en los diversos experimentos realizados para la optimización del calendario de la NBA, cuyo objetivo principal ha sido minimizar la distancia total recorrida por los equipos a lo largo de una temporada regular.

En primer lugar, se presentará una breve descripción acerca del entorno de trabajo utilizado durante este proyecto, incluyendo las herramientas utilizadas en la implementación de los modelos y los algoritmos, así como las características técnicas del equipo empleado para la ejecución de las pruebas.

6.1. Entorno de experimentación

Antes de presentar y analizar los resultados obtenidos, es importante contextualizar el entorno en el que se han realizado los experimentos computacionales. Cabe destacar que las características del hardware y las herramientas de desarrollo utilizadas pueden influir directamente en los tiempos de ejecución y en la viabilidad de los cálculos realizados.

6.1.1. Especificaciones del equipo

Todos los experimentos y pruebas del proyecto se han ejecutado en un equipo portátil con las siguientes características técnicas:

- **Procesador:** Intel(R) Core(TM) i7-10510U CPU @ 1.80GHz, con una frecuencia máxima de 2.30 GHz.
- **Memoria RAM:** 16 GB (15.8 GB utilizables).
- **Arquitectura del sistema:** Sistema operativo de 64 bits, con procesador basado en arquitectura x64.
- **Sistema operativo:** Windows 11 Home.

Este entorno, a pesar de no ser una estación de alto rendimiento, ha resultado suficiente para realizar la mayoría de los experimentos requeridos en el marco del proyecto, a excepción de algunos casos que se verán a continuación en este capítulo.

6.1.2. Herramientas y lenguajes utilizados

Para el desarrollo e implementación de los modelos y algoritmos del proyecto se han empleado las siguientes herramientas y tecnologías:

- **Lenguaje de programación:** Todo el desarrollo, tanto de modelos matemáticos como de heurísticas y metaheurísticas, se ha realizado en C++, debido a su alto rendimiento y control sobre los recursos del sistema.
- **Optimizador matemático:** Se ha utilizado Gurobi Optimizer en su versión 11.0.3. Se trata de uno de los optimizadores más potentes del mercado (gratuito para estudiantes) y ha sido empleado para resolver los modelos matemáticos formulados en el Capítulo 3.
- **Entorno de desarrollo:** Todo el código ha sido desarrollado en Microsoft Visual Studio, un entorno de desarrollo integrado que facilita la depuración, la gestión de proyectos y la compilación eficiente de código en C++.

6.2. Casos de estudio y análisis de resultados

Como se ha adelantado en el Capítulo 2, se han considerado dos escenarios distintos para el estudio del calendario de la NBA, ambos con el objetivo de minimizar las distancias recorridas por los equipos. El primero corresponde a una versión de calendario estructurado (con tiempo restringido), en la que la temporada se estructura en 82 jornadas y todos los equipos juegan en cada una de ellas. El segundo caso plantea una versión no estructurada (con tiempo relajado), más realista, donde los 82 partidos de cada equipo se reparten a lo largo de 163 días, asemejándose en mayor medida al calendario oficial de la competición.

A su vez, dentro del enfoque de calendario estructurado, se han desarrollado versiones simplificadas con ligas de 4 y 8 equipos, con el propósito de analizar con mayor precisión el comportamiento y la efectividad de los algoritmos heurísticos y metaheurísticos implementados.

La experimentación realizada ha permitido analizar el comportamiento de los modelos matemáticos, heurísticas y metaheurísticas en ambos tipos de calendario: estructurado y no estructurado. Para realizar el análisis de los resultados se ha utilizado la diferencia relativa, conocida como *Gap*, definida como:

$$\text{Gap} = \left(\frac{\text{Solución actual} - \text{Solución de referencia}}{\text{Solución de referencia}} \right) \times 100$$

Este indicador refleja el porcentaje de mejora de una determinada solución con respecto a la solución de referencia. Si el porcentaje es negativo, significa que la solución actual recorre menos millas que la de referencia, por lo que representa una mejora. En cambio, si el porcentaje es positivo, implica que la solución actual es peor que la solución de referencia.

Antes de presentar y discutir los resultados obtenidos y su correspondiente análisis, se describen algunas especificaciones metodológicas aplicadas a los algoritmos empleados.

6.2.1. Especificaciones en las metaheurísticas

En el caso de las metaheurísticas, se llevó a cabo una búsqueda en rejilla (*Grid Search*) para ajustar y seleccionar las combinaciones de parámetros que proporcionan los mejores resultados, buscando siempre un equilibrio adecuado entre exploración y convergencia. A continuación, se describen los rangos de parámetros evaluados para cada metaheurística, ajustados cuidadosamente en función de la complejidad y las características propias de cada uno de los escenarios de liga considerados.

Para el Temple Simulado, en las ligas estructuradas reducidas de 4 y 8 equipos se exploraron temperaturas iniciales de 50 y 100, una temperatura mínima fija de 0.01, un amplio rango de cambios posibles (de 2 a 10) y valores de α de 0.90, 0.95 y 0.99, ya que al ser casos reducidos, permitieron una exploración más exhaustiva debido a su bajo coste computacional. En contraste, para el calendario estructurado de 30 equipos, se redujo el número de cambios a 2 y 8, ya que se

observó que mayores variaciones no aportaban mejoras significativas. Las temperaturas iniciales, el valor mínimo y los coeficientes de enfriamiento α se mantuvieron en rangos similares a los de las ligas reducidas. Para el escenario más complejo, la liga de 30 equipos extendida a 163 días (no estructurada), se aumentó la temperatura mínima a 0.1. Esto se debió a que, dado el gran tamaño del espacio de búsqueda, las últimas iteraciones aportaban escasas mejoras, por lo que se prefirió finalizar antes el proceso de búsqueda para preservar la eficiencia computacional. En este caso se evaluaron 2, 5 y 8 cambios.

Respecto a la Búsqueda por Entornos Variables (VNS), en todos los escenarios se mantuvo la estructura de tres vecindarios ($k_1 = 2$, $k_2 = 3$, $k_3 = 4$) para asegurar un balance entre diversidad en la exploración y eficiencia computacional. Se descartó el uso de más vecindarios ya que incrementaban considerablemente el coste computacional sin mejoras proporcionales en la calidad de las soluciones. Asimismo, se experimentó con distintos valores de iteraciones máximas (100, 200 y 400) y umbrales de iteraciones sin mejora (10 y 50), con el objetivo de controlar la complejidad computacional y evitar estancamientos. Todos estos parámetros se aplicaron de forma consistente en todos los escenarios.

Para la Búsqueda Tabú, los ajustes se realizaron en función del tipo de calendario. En escenarios estructurados, se probaron límites de iteraciones de 100, 200 y 500, junto con tamaños de lista tabú de 10, 50 y 100. En cambio, para el calendario no estructurado, debido a una mejora más lenta en las soluciones, se optó desde el inicio por un número elevado de iteraciones (500) para permitir una mayor exploración del espacio de búsqueda.

Cabe señalar que los tiempos mostrados para cada metaheurística corresponden al proceso completo de ejecución. No obstante, en algunos casos, la mejor solución se obtiene antes de finalizar dicho proceso, destinándose el tiempo restante a una exploración adicional del espacio de búsqueda que no conduce a mejoras significativas.

Por último, es importante destacar que todas las metaheurísticas aplicadas sobre calendarios estructurados se basaron en tres configuraciones heurísticas: la heurística 2, la heurística 3 y la combinación secuencial de la 2 seguida de la 3. Esta última opción fue la que, en general, ofreció los mejores resultados, como se verá a continuación. Por otro lado, las metaheurísticas aplicadas sobre calendarios no estructurados se basaron únicamente en la heurística 4.

6.2.2. Calendarios estructurados

Como se verá a continuación, al abordar el calendario de la NBA organizado por jornadas (en el que participan 30 equipos), el Modelo 2 implementado en Gurobi no permitió obtener una solución que minimizase las distancias recorridas de forma eficiente. Por ello, se recurrió al uso de heurísticas y metaheurísticas, presentadas en los capítulos 4 y 5, partiendo de una solución factible proporcionada por el Modelo 1, con el objetivo de aproximarse a una solución de calidad.

Tal como se ha mencionado anteriormente, con el fin de analizar cómo el modelo se ve afectado por el aumento en el número de equipos y restricciones, así como para evaluar la eficacia de los algoritmos implementados, se diseñaron versiones reducidas del calendario estructurado con ligas de 4 y 8 equipos.

6.2.2.1. Liga de 4 equipos

Esta liga reducida está compuesta por solo 4 equipos, en concreto los Boston Celtics, New York Knicks, Philadelphia 76ers y Brooklyn Nets, todos ellos miembros de la División Atlántico de la Conferencia Este. En esta configuración simplificada solo se disputan los partidos de división, por lo que cada equipo se enfrenta 4 veces a cada uno de sus 3 rivales, resultando en un total de 12 jornadas.

El resumen de los resultados de este caso estructurado reducido de 4 equipos se puede consultar en la Figura 6.1.

Equipo	RESUMEN RESULTADOS									
	Millas recorridas									
	Modelo	Heurística 1	Heurística 2	Heurística 3	Heurísticas 2 y 3	Heurísticas 3 y 2	Heurísticas 1, 2 y 3	Temple Simulado	VNS	Búsqueda Tabú
	626.47	1258.82	627.8	826.71	627.8	626.47	626.65	626.47	626.47	626.47
	626.65	1253.3	829.56	1265.54	825.56	642.94	629.32	626.65	626.47	633.37
	626.47	826.89	818.84	1019.08	818.84	818.66	629.32	626.47	626.65	815.99
	627.8	1257.12	642.7	1243.55	631.8	630.47	630.65	627.8	627.8	631.8
Millas Totales	2507.39	4596.13	2918.9	4354.88	2904	2718.54	2515.94	2507.39	2507.39	2707.63
Tiempo	2 seg	2 seg	1 seg	1 seg	1 seg	1 seg	2 seg	1 seg	1 seg	1 seg
Gap	-	83.30%	16.41%	73.68%	15.81%	8.42%	0.34%	0%	0%	7.98%

Figura 6.1: Resultados calendario estructurado versión reducida 4 equipos.

Modelos matemáticos

Dado el tamaño reducido del problema, se optó por aplicar directamente el Modelo 2, con el objetivo de minimizar las distancias recorridas. Este modelo incluyó un total de 3.000 restricciones y 1.056 variables binarias. En apenas 2 segundos se obtuvo una solución óptima con un total de 2.507,39 millas recorridas. Este valor servirá como referencia para evaluar la calidad de las soluciones proporcionadas por las heurísticas y metaheurísticas desarrolladas.

Por otro lado, para disponer de una solución factible inicial sobre la que aplicar dichos métodos aproximados, se resolvió el Modelo 1. Este modelo más simple, con 72 restricciones y 144 variables binarias, generó un calendario con un recorrido total de 6.260,22 millas (149,67 % de *Gap*) en tan solo un segundo de ejecución.

Heurísticas

En primer lugar, se aplicó la heurística 1, ejecutando el Modelo 1 en múltiples ocasiones. Tras 50 ejecuciones en un par de segundos, se obtuvo como mejor resultado un calendario con un recorrido total de 4.596,13 millas y un *Gap* del 83,30 %, es decir, una solución muy alejada del óptimo.

A continuación se utilizaron las heurísticas 2 y 3 de forma individual, observando que la heurística 2 proporcionó mejores resultados que la 3. La heurística 2 generó un calendario que reducía la distancia a 2.918,9 millas (16,41 %), mientras que la heurística 3 confeccionó una solución de 4.354,88 millas (73,68 %), ambas también con tiempos de ejecución muy bajos (alrededor de un segundo).

Seguidamente, se evaluaron combinaciones entre estas heurísticas. La combinación de la heurística 2 seguida de la heurística 3 resultó en un calendario de 2.904 millas (15,81 %), mientras que en el orden inverso (3 seguida de 2) la distancia se redujo aún más, alcanzando las

2.718,54 millas (8,42 %). No obstante, el mejor resultado (en el ámbito heurístico) se alcanzó al combinar de forma secuencial las tres heurísticas (1, 2 y 3, en ese orden), obteniendo un calendario de 2.515,94 millas, equivalente a tan solo un *Gap* del 0,34 % respecto a la solución óptima, en aproximadamente un par de segundos.

Metaheurísticas

Se aplicaron las tres metaheurísticas descritas en el Capítulo 5: Temple Simulado, VNS y Búsqueda Tabú.

Los mejores resultados obtenidos con el Temple Simulado fueron consistentes entre distintas ejecuciones ya que la mayoría de las configuraciones de parámetros llegaban a los siguientes resultados:

- **Temple Simulado con heurística 2:** 2.732,1 millas (8,96 %) en aproximadamente un segundo.
- **Temple Simulado con heurística 3:** 4.160,99 millas (65,94 %) en un segundo.
- **Temple Simulado con heurísticas 2 y 3:** 2.507,39 millas, que coincide con el óptimo proporcionado por el Modelo 2, en un segundo.

Al igual que con el Temple Simulado, la mayoría de configuraciones de VNS han alcanzado los mejores resultados:

- **VNS con heurística 2:** 2.732,1 millas (8,96 %) en un segundo.
- **VNS con heurística 3:** 4.160,99 millas (65,94 %) en un segundo.
- **VNS con heurísticas 2 y 3:** 2.507,39, alcanzando el óptimo en un segundo.

Por último, para la Búsqueda Tabú se obtuvieron en la mayoría de configuraciones los siguientes resultados:

- **Búsqueda tabú con heurística 2:** 2.733,43 millas (9,01 %) en un segundo.
- **Búsqueda tabú con heurística 3:** 5.215,22 millas (107,99 %) en un segundo.
- **Búsqueda tabú con heurísticas 2 y 3:** 2.707,63 millas (7,98 %) en un segundo.

En términos generales, las tres metaheurísticas alcanzaron sus mejores resultados al combinar las heurísticas 2 y 3. Tanto Temple Simulado como VNS demostraron una alta eficacia, siendo capaces de alcanzar de forma consistente la solución óptima en tiempos computacionales muy bajos. Por otro lado, la Búsqueda Tabú, aunque fue ligeramente menos precisa, también obtuvo resultados competitivos.

6.2.2.2. Liga de 8 equipos

Esta liga está formada por 8 equipos distribuidos en dos conferencias (Este y Oeste), con 4 equipos en cada una. Todos los equipos dentro de una conferencia pertenecen a la misma división en este caso. En esta configuración se disputan tanto los partidos de división como interconferencia. Cada equipo se enfrenta 4 veces a los equipos de su propia división y 2 veces

a cada uno de los equipos de la otra conferencia, lo que da lugar a un calendario total de 20 jornadas.

El resumen de los resultados de este caso estructurado reducido de 8 equipos se puede consultar en la Figura 6.2.

Equipo	RESUMEN RESULTADOS									
	Millas recorridas									
	Modelo	Heurística 1	Heurística 2	Heurística 3	Heurísticas 2 y 3	Heurísticas 3 y 2	Heurísticas 1, 2 y 3	Temple Simulado	VNS	Búsqueda Tabú
	-	5754.6	6612.6	6786.62	6612.6	6612.6	5720.17	5720.17	6976.57	6801.1
	-	7172.45	7280.19	7172.45	7280.19	7280.19	7276.71	6428.5	6250.93	7297.72
	-	5980	7002.33	6894.6	7002.33	7002.33	5961.89	5961.89	5565.53	5626.84
	-	6389.32	6246.26	6389.32	6246.26	6246.26	6246.9	6432.98	5751.61	5911.4
	-	6389.32	6389.32	6389.32	6389.32	6389.32	6389.32	6273.45	6380.55	6714.33
	-	5874.14	5847.21	5847.21	5847.21	5847.21	5874.14	6431.95	6253.16	6638.87
	-	6121.82	6121.82	6121.82	6121.82	6121.82	6121.82	6292.4	6292.58	6251.05
	-	6411.32	6246.9	6246.9	6246.9	6246.9	6411.32	5997.23	5977.63	6438.6
Millas Totales	50393.49	50093	51746.6	51848.2	51746.6	51746.6	50002.3	49536.6	49448.6	51679.9
Tiempo	44662 seg	2 seg	1 seg	1 seg	2 seg	1 seg	2 seg	4 seg	3 seg	1 seg
Gap	-	-0.59%	2.68%	2.88%	2.68%	2.68%	-0.77%	-1.70%	-1.87%	2.55%

Figura 6.2: Resultados calendario estructurado versión reducida 8 equipos.

Modelos matemáticos

Dado el tamaño del problema, se intentó resolver inicialmente el Modelo 2, como ya se había hecho en el caso reducido de 4 equipos, con el objetivo de minimizar la distancia total recorrida. Sin embargo, la solución exacta no pudo ser obtenida debido al elevado coste computacional, lo que evidencia cómo la complejidad del problema crece rápidamente al incrementar tan solo unos pocos equipos. Este modelo comprendía 35.312 restricciones y 12.610 variables binarias. Tras una ejecución de 12 horas, 24 minutos y 22 segundos, se logró conocer que hasta ese momento, la mejor solución encontrada era de 50.393,49 millas, con un límite inferior estimado en 32.925,45 millas. Esto supondría que la solución obtenida partiría de un gap de optimalidad proporcionado por Gurobi del 34,7 % con respecto a la hipotética mejor solución que se podría alcanzar.

Sin embargo, dado que no se puede garantizar que dicho límite inferior sea factible, se considerará la solución de 50.393,49 millas como referencia para comparar con los resultados obtenidos mediante el resto de algoritmos. Debido al elevado coste computacional y a la lentitud en la mejora de resultados, se decidió interrumpir la ejecución y pasar a aplicar las distintas heurísticas.

Con el objetivo de generar una solución inicial factible que sirviera como punto de partida para los métodos aproximados, se aplicó el Modelo 1. Este modelo incluía 272 restricciones

y 1120 variables binarias. En apenas un par de segundos se obtuvo una solución factible con un total de 51.848 millas recorridas, que supondría una solución muy cercana a la solución de referencia (2,88 % de *Gap*).

Heurísticas

En primer lugar, se aplicó la heurística 1. Tras 50 ejecuciones del Modelo 1, se obtuvo como mejor solución un calendario con un recorrido total de 50.093 millas (-0,59 %), en aproximadamente un par de segundos, superando la solución de referencia.

Posteriormente, se aplicaron las heurísticas 2 y 3 de forma individual. Sin embargo, ambas se mostraron poco eficaces en este escenario. La heurística 2 generó un calendario con 51.746,6 millas (2,68 %), mientras que la heurística 3 produjo un calendario con 51.848 millas (2,88 %). Ambos métodos ofrecieron soluciones en tiempos cercanos al segundo. Además, la combinación de estas dos heurísticas, en ambos órdenes (2 seguida de 3 y 3 seguida de 2), ofreció exactamente el mismo resultado: 51.746,6 millas (2,68 %).

Finalmente, la combinación completa de las tres heurísticas (1, 2 y 3) produjo el mejor resultado hasta el momento, con un recorrido total de 50.002,3 millas (-0,77 %) en un par de segundos, superando la solución de referencia y reafirmando la eficacia de la heurística 1 como punto de partida en problemas de menor tamaño, donde la resolución del Modelo 1 presenta bajo coste computacional.

Metaheurísticas

Al igual que para la liga de 4 equipos, se aplicaron tres metaheurísticas diferentes: Temple Simulado, VNS y Búsqueda Tabú.

A continuación se muestran los mejores resultados obtenidos con el Temple Simulado, acompañados de la configuración de parámetros con la que se obtuvieron:

- **Temple Simulado con heurística 2:** 49.538 millas (-1,69 %) en 4 segundos (*Temp. inicial* = 50, *Temp. mínima* = 0.01, *Cambios* = 8, α = 0.99).
- **Temple Simulado con heurística 3:** 49.851,5 millas (-1,07 %) en 2 segundos (*Temp. inicial* = 100, *Temp. mínima* = 0.01, *Cambios* = 10, α = 0.99).
- **Temple Simulado con heurísticas 2 y 3:** 49.536,6 millas (-1,70 %) en 4 segundos (*Temp. inicial* = 100, *Temp. mínima* = 0.01, *Cambios* = 10, α = 0.99).

Los resultados más destacados tras aplicar VNS fueron:

- **VNS con heurística 2:** 49.533 millas (-1,71 %) en 4 segundos (*Iteraciones máximas* = 200, *Iteraciones sin mejora* = 50).
- **VNS con heurística 3:** 50.615,6 millas (-0,44 %) en 1 segundo (la mayoría de las configuraciones de parámetros llegan a este resultado).
- **VNS con heurísticas 2 y 3:** 49.448,6 millas (-1,87 %) en 3 segundos (*Iteraciones máximas* = 100, *Iteraciones sin mejora* = 50).

Por último, se empleó la metaheurística de Búsqueda Tabú, que ofreció los siguientes mejores resultados para la mayoría de configuraciones de parámetros:

- **Búsqueda tabú con heurística 2:** 50.534,1 millas (0,27 %) en un segundo.
- **Búsqueda tabú con heurística 3:** 51.848,2 millas (2,88 %) en un segundo.
- **Búsqueda tabú con heurísticas 2 y 3:** 51.679,9 millas (2,55 %) en un segundo.

Una vez más, las metaheurísticas de Temple Simulado y VNS fueron las que ofrecieron los mejores resultados, alcanzando soluciones superiores a la de referencia, con *Gap* de -1,70 % y -1,87 % respectivamente. En contraste, la Búsqueda Tabú se mostró ineficaz en este escenario, produciendo soluciones considerablemente peores y sin capacidad real de mejora.

Estos resultados, junto con las escasas mejoras obtenidas en general, evidencian la dificultad de optimizar un calendario estructurado por jornadas debido a las fuertes restricciones que limitan los posibles movimientos, dificultando la exploración de alternativas significativamente mejores.

6.2.2.3. Liga regular

Esta liga corresponde a una temporada regular completa de la NBA (30 equipos), con la diferencia de que está organizada por jornadas (82) en vez de por días. A diferencia de las ligas reducidas de 4 y 8 equipos, donde no era necesario por haber un número muy bajo de jornadas, en esta liga se añadió una restricción adicional para equilibrar el calendario (vista en el Capítulo 3): todos los equipos deben mantener una diferencia máxima de 10 partidos entre los jugados como local y como visitante en cualquier momento de la temporada ($\Delta_{\max} = 10$). Esto asegura un calendario más justo y balanceado.

El resumen de los resultados del caso real estructurado de 30 equipos se puede consultar en la Figura 6.3.

Modelo matemático

Debido a la complejidad del problema, se descartó la ejecución del Modelo 2 desde el inicio ya que no se obtuvieron resultados por su elevado coste computacional. Este modelo contaba con 6.882.780 restricciones y 2.316.600 variables binarias.

Al no ser posible resolver este modelo exacto, no se pudo contar con una solución estructurada óptima como referencia, a diferencia del caso reducido de 4 equipos. Por ello, se decidió tomar como referencia el calendario oficial de la NBA 2022-2023, en el que se recorrián 1.465.250 millas en total. Aunque dicho calendario corresponde a una planificación no estructurada (con tiempo relajado), se empleó esta comparación para evaluar la eficiencia de calendarios estructurados frente al calendario oficial no estructurado.

En consecuencia, se optó por resolver el Modelo 1 para obtener una solución factible y poder continuar con los métodos aproximados. El Modelo 1 constaba de 9.120 restricciones y 71.340 variables binarias. La solución inicial alcanzada implicó un recorrido total de 2.284.150 millas en un tiempo de ejecución de 4 minutos y 20 segundos, lo que supone un resultado muy alejado a la solución de referencia (55,88 % de *Gap*).

Heurísticas

Primero se aplicó la heurística 1, ejecutando el Modelo 1 mediante Gurobi tan solo en 5 ocasiones, debido al elevado coste computacional, obteniéndose como mejor resultado un calendario con un recorrido de 2.241.610 millas tras 25 minutos y 30 segundos de ejecución. Se puede observar que, al tratarse de un problema más grande y complejo, esta heurística perdió eficacia.

RESUMEN RESULTADOS																	
Equipo	Millas recorridas																
	Calendario 82 jornadas												Calendario 163 días				
	Calendario 22-23	Heuristica 1	Heuristica 2	Heuristica 3	Heurísticas 2 y 3	Heurísticas 3 y 2	Heurísticas 1, 2 y 3	Temple Simulado	VNS	Búsqueda Tabú	Heurística 4	Heurística balanceado	Temple Simulado	VNS	Búsqueda Tabú		
Nordeste		52194.0	74881.3	66912	64644	62670.1	61421.7	55132.7	52769.9	56870	58828.5	23205.4	44628.6	40402.7	31634.7	40611.9	
		55411.3	77667.9	74232.4	65084	63572.7	64336.8	61607	53017	65273.5	65387.6	24365.2	50210.8	39151.7	35729.2	45128.8	
		54931.5	83102.3	67906.5	65150.6	60555	68227.8	67180.7	62695.8	46099	70151	27477	45207	40408.4	39031.5	47102.4	
		59803.9	76369.6	60707.9	60744	57541.9	64907.6	60318.9	55930.4	56309.2	61928.2	21519.6	43343	38702.3	34298.3	40619.5	
		57706.9	101467	92838.9	80746	74074	74852.1	71754.1	71115.6	67334.8	71233.3	33335.2	54197.8	42993.5	38869	52773	
Suroeste		49579.8	74520.4	50317.8	62642.8	53988.1	59968.8	57674.6	55993.7	54293.9	60430.4	26130.5	46373.1	42001.2	29035.2	41003	
		52244.0	73005.5	65321.2	55058.2	59243.6	54480.7	48972.7	60451.9	51043.9	60339.8	24934	45722.8	41371.9	32516.7	45506.3	
		44075.6	80958.2	54266.8	63807.7	57024.7	65488.1	55437.1	56746.7	58086.6	63190.6	23998.7	41587.9	40527.6	35156.9	46308.4	
		48013.4	72051.4	68066.2	55680	59761.3	57244.3	52931.3	52731.2	55647.8	55822.8	24324.7	45582.8	41669.3	30992.3	42041.9	
		52017.4	87390.4	69524.7	58011.4	56821.9	62515.9	64104.2	51337.9	46175.2	60560.7	23655.4	39515.2	36723.3	36158.3	47814.7	
Pacifico		49479.4	82934.7	70810.7	65146.4	67467.9	65183	57545	61987.9	51823.6	62089.8	28074.6	40244	36848.9	38615.4	46507.4	
		51948.0	76419.2	73594.2	65231.4	69349.1	65807	55177.2	57321.6	55940.2	61920.7	25412.1	43134.6	42888	36086.3	52402	
		56292.7	86477.5	76590.6	66214.7	65635.3	64443.9	56910.4	62440.2	61401.1	67961.8	30731.7	50925.7	44277	38560	49875.1	
		55726.9	100104	63534.2	74625.3	62184.1	66728.9	55997.5	48701.9	64692.9	70086.9	28917.7	51254.4	48051	39361.7	49451.2	
		54124.1	85359.8	68351.9	66916.6	58386.4	58141	65063.8	64368	59308.3	60729.8	26621.3	47101.2	40493.2	38943.9	42897.5	
Atlántico		49054.4	68584	67880.7	60151.7	53249.9	58379.6	54575.9	57234.6	53412	51999.2	22665.7	37572.4	43549.9	40661.4	37444.6	
		43363.5	68793.8	59253	55906.3	51384.7	54728.9	55363.9	55041.6	50239.8	49702	23027.4	42700.3	40375	31930.6	39247.6	
		42881.6	65726.8	57413.4	56893.9	50867.7	57716.8	54244.7	50638.4	48785.5	58341.6	22948.7	39242.3	34987	38052.4	42332.2	
		46909	64459.7	58736.1	60819.7	50033.2	54452.2	51097.6	49369.8	46281.7	54494.3	22070.1	36247.3	33625.5	34811.1	45300.2	
		40878.3	66153.7	56665.7	57816	49199.8	53234.6	54422.4	56110.2	48566.8	58872.8	22577.3	34547.9	35325.3	32399.6	40991.1	
Central		47296.5	56397.1	51886.7	58859.9	49086.5	59574.5	53290.1	50001.7	44478.6	53039.3	21244.6	37766.3	36869.1	30191.8	45213.2	
		41371	62366.9	53764.9	53568.8	51974.6	52764.5	49917.4	50593.7	44985.9	52863.3	20415.8	41068.2	36392	33685	38192.4	
		39725.9	65567.4	54315.6	62063.6	49735	60566.5	53808.7	46199.4	40668.3	54895.8	23592	35120.5	30328.8	30826.8	36375.7	
		42217.9	62519.6	57790.2	57037.1	45531.5	54952.7	46135.4	51424.1	50342.5	48438.6	23960.4	41165.8	44197.8	38254.4	42888.4	
		46970.1	64492.4	52268.8	53117.4	48025.7	52396	53303.9	46310.6	51716.8	47936.2	21446.8	38476.4	36676.2	31231.6	37786.7	
Sureste		43459.5	61633.3	57154.5	53226.3	52697.5	52085	56614.5	54085	54491.8	54095.4	24823.6	47213.9	34448.6	29740.7	34390.1	
		41250.9	68145.5	59591.6	62697.5	52679.6	64568	47435.7	48238.1	47606.8	65213.4	20231.3	37317.2	34831	33690.4	40340.2	
		49912.2	78625.1	56441.6	59988.9	57377.6	59284.6	57090.8	48821.5	56035.6	50528.9	28896.3	49352.2	40812.3	37975.1	42250.1	
		45782.8	67277.8	51170.9	62268.2	48719.3	57077.2	48457	47624.5	49947.2	53965	23240.6	34648.2	36186.9	28230.1	36538.3	
		50626.2	88157.1	77917.5	64483.7	64885.3	54091	62013.3	65028.2	49098	60454	21616.4	39941.3	40336.2	33931.4	44644	
Millas Oeste	793549	1232710	1031980	969703	928276	953748	885807	867610	850300	950662	392703	689029	616510	534959	690043		
Millas Este	671700	1008900	872251	878899	775448	845872	797772	776730	736658	814840	342758	592380	558942	505612	603935		
Millas Totales	1465250	2241610	1904230	1848600	1703720	1799620	1683580	1644340	1586960	1765500	735461	1281410	1175450	1040570	1293980		
Tiempo	-	1530 seg	4 seg	10 seg	11 seg	14 seg	1683 seg	238 seg	726 seg	32 seg	3 seg	4 seg	117 seg	1158 seg	2640 seg		
Gap	-	52.98%	29.95%	26.16%	16.27%	22.81%	14.90%	12.22%	8.30%	20.49%	-49.80%	-12.54%	-19.77%	-28.98%	-11.66%		

Figura 6.3: Resultados finales liga regular, 30 equipos.

La ejecución del modelo resultó mucho más costosa en tiempo y las soluciones obtenidas estaban muy alejadas de la referencia (52,98 % de *Gap*), a diferencia de lo observado en los casos reducidos de 4 y 8 equipos.

Posteriormente se aplicaron heurísticas 2 y 3 de forma individual. La heurística 2 generó un calendario con 1.904.230 millas (29,95 %) en apenas 4 segundos, mientras que la heurística 3, con

mayor coste computacional, produjo un calendario de 1.848.600 millas (26,16 %) en 10 segundos.

A continuación, se probaron combinaciones de estas heurísticas. La combinación de la heurística 2 seguida de la 3 produjo un calendario con 1.703.720 millas (16,27 %) en tan solo 11 segundos, ofreciendo un excelente balance entre calidad de solución y tiempo de cómputo. Por otro lado, al invertir el orden (3 seguida de 2), se obtuvo un calendario de 1.799.620 millas (22,81 %) en 14 segundos.

Finalmente, la combinación de las heurísticas 1, 2 y 3 (en ese orden) alcanzó el mejor resultado hasta el momento, con un total de 1.683.580 millas (14,90 %), aunque con un coste computacional considerablemente mayor, tardando 28 minutos y 3 segundos, convirtiéndola en una opción poco eficiente.

Metaheurísticas

Se aplicaron las metaheurísticas de Temple Simulado, VNS y Búsqueda Tabú, obteniendo los resultados que se verán a continuación.

En primer lugar, para el Temple Simulado se alcanzaron los siguientes mejores resultados con sus respectivas configuraciones:

- **Temple Simulado con heurística 2:** 1.841.540 millas (25,68 %) en 1 minuto y 56 segundos (*Temp. inicial* = 50, *Temp. mínima* = 0.01, *Cambios* = 2, α = 0.95).
- **Temple Simulado con heurística 3:** 1.828.110 millas (24,76 %) en 4 segundos (*Temp. inicial* = 50, *Temp. mínima* = 0.01, *Cambios* = 2, α = 0.95).
- **Temple Simulado con heurísticas 2 y 3:** 1.644.340 millas (12,22 %) en 3 minutos y 58 segundos (*Temp. inicial* = 50, *Temp. mínima* = 0.01, *Cambios* = 8, α = 0.95).

Para VNS se obtuvieron los siguientes resultados:

- **VNS con heurística 2:** 1.812.750 millas (23,71 %) en 19 minutos y 15 segundos (*Iteraciones máximas* = 400, *Iteraciones sin mejora* = 50).
- **VNS con heurística 3:** 1.805.830 millas (23,24 %) en 11 segundos (*Iteraciones máximas* = 200, *Iteraciones sin mejora* = 10).
- **VNS con heurísticas 2 y 3:** 1.586.960 millas (8,30 %) en 12 minutos y 6 segundos (*Iteraciones máximas* = 200, *Iteraciones sin mejora* = 50).

Por último, mediante la Búsqueda Tabú se obtuvieron:

- **Búsqueda tabú con heurística 2:** 1.902.980 millas (29,87 %) en 1 minuto y 20 segundos (*Iteraciones máximas* = 500, *Tamaño de la lista tabú* = 100).
- **Búsqueda tabú con heurística 3:** 1.914.060 millas (30,63 %) en 3 segundos (*Iteraciones máximas* = 500, *Tamaño de la lista tabú* = 10).
- **Búsqueda tabú con heurísticas 2 y 3:** 1.765.500 millas (20,49 %) en 32 segundos (*Iteraciones máximas* = 200, *Tamaño de la lista tabú* = 10).

Como se ha podido observar, los mejores resultados en metaheurísticas se producen en la combinación de las heurísticas 2 y 3, en ese orden. VNS destacó nuevamente como la que mejor resultados produce, alcanzando un *Gap* del 8,30 %, aunque con un coste computacional elevado.

El Temple Simulado también presentó buenos resultados, algo peores que los de VNS, pero con menor coste computacional. Por su parte, la Búsqueda Tabú siguió obteniendo resultados peores, mostrando mejoras muy limitadas.

Tras recopilar y analizar todos los resultados correspondientes al caso regular con calendario estructurado (30 equipos), se observa que no ha sido posible reducir la distancia total recorrida respecto al calendario oficial de la temporada 2022-2023. Esto pone de manifiesto cómo la rigidez de una planificación basada en jornadas limita significativamente la posibilidad de optimizar y reducir las millas recorridas.

6.2.3. Calendarios no estructurados

Una vez finalizado el estudio del calendario estructurado en 82 jornadas, se abordó una nueva variante basada en un calendario diario, más cercano al formato real utilizado por la NBA, que se reconocerá como calendario no estructurado. En este nuevo enfoque, la temporada se extiende a lo largo de 163 días, distribuyendo los partidos día a día en lugar de hacerlo por jornadas fijas.

Se siguieron dos estrategias para afrontar este problema:

- **Enfoque 1:** Partir de la mejor solución obtenida en el calendario estructurado de 82 jornadas, expandirla a 163 días y optimizarla mediante la aplicación de la heurística de inserción (heurística 4).
- **Enfoque 2:** Ampliar directamente el calendario inicial estructurado (Modelo 1) a 163 días y aplicar las metaheurísticas adaptadas al nuevo tipo de calendario diario.

El resumen de los resultados del caso real no estructurado de 30 equipos se puede consultar en la Figura 6.3.

6.2.3.1. Enfoque 1

Este enfoque parte de la mejor solución obtenida en el caso regular estructurado, concretamente el calendario generado con VNS, que alcanzaba un recorrido de 1.586.960 millas (8,30 % de *Gap*). El proceso de extensión consistió en intercalar descansos de 1 y 3 días alternativamente cada dos partidos. Esta estrategia inicial permitía respetar la restricción de mantener un equilibrio en la diferencia de partidos jugados por cada equipo como local y como visitante en cualquier momento de la temporada ($\Delta_{\text{máx}} = 10$).

En primer lugar, se aplicó la heurística 4 sin restricciones adicionales, logrando un calendario de 735.461 millas en tan solo 3 segundos, que suponía un *Gap* del -49,80 %. Sin embargo, al analizar el calendario resultante, se detectaron violaciones de criterios presentes en el calendario oficial de la NBA, como secuencias excesivas de partidos consecutivos o días de descanso excesivos.

Para corregirlo, se incorporó un máximo de 4 días consecutivos de descanso para cada equipo, así como de 2 partidos en días consecutivos. Con estas condiciones añadidas, se generó un calendario que aumentaba el número de millas anterior hasta 1.034.620 millas en 4 segundos, lo que suponía un *Gap* del -29,38 %.

Posteriormente, se observó que algunos equipos acumulaban muchos más partidos que otros en determinados momentos de la temporada. Para mejorar el balance, se añadió a la heurística 4 la restricción de que la diferencia máxima de partidos jugados entre cualquier par de equipos en cualquier momento de la temporada no superase los 4 partidos, obteniendo un nuevo calendario de 1.169.040 millas (-20,21 %) en 4 segundos.

Finalmente, se detectaron desequilibrios en la distribución de partidos como local y visitante para algunos equipos. Aunque esta restricción se había considerado desde el inicio (calendario inicial de 82 jornadas y posterior extensión a 163 días), la heurística 4 no la tenía incorporada de forma explícita. Para corregirlo, se modificó la heurística estableciendo el límite máximo $\Delta_{\text{máx}}$ a 10 partidos de diferencia entre los encuentros jugados en casa y fuera para cualquier equipo. Con esta adaptación, se obtuvo un calendario final de 1.281.410 millas (-12,54 %), con un tiempo de cómputo de 4 segundos.

A medida que se fueron incorporando restricciones para acercar el calendario a una versión más realista, la distancia total recorrida por los equipos fue aumentando. Esto se debe a que las nuevas condiciones limitaban progresivamente la libertad para realizar ciertos movimientos. Sin embargo, estas restricciones también contribuyeron a mejorar la calidad del calendario, permitiendo alcanzar una versión final más equilibrada, que incluso representa una mejora respecto al calendario oficial de la NBA en términos de distancia. No obstante, cabe destacar que esta comparación no es completamente equitativa, ya que la NBA considera muchos otros factores en la elaboración del calendario más allá de la minimización de distancias.

6.2.3.2. Enfoque 2

En este segundo enfoque, se partió directamente de la solución inicial de 82 jornadas sin optimizar (Modelo 1) y se amplió a 163 días utilizando el mismo esquema de descansos alternos (1 y 3 días). A partir de esta base extendida, se aplicaron las metaheurísticas descritas en el Capítulo 5 adaptadas para este nuevo tipo de calendario diario.

Durante cada modificación, se aseguró que se cumplieran las restricciones de equilibrio en días de descanso, secuencia de partidos seguidos, número de partidos jugados por equipo y balance entre partidos como local y visitante. Así, cada paso en la búsqueda producía soluciones factibles y realistas desde el punto de vista deportivo.

En esta fase se aplicaron las tres metaheurísticas: Temple Simulado, VNS y Búsqueda Tabú, pero esta vez utilizando la heurística 4 en su versión balanceada (es decir, cumpliendo todas las restricciones mencionadas anteriormente).

Los resultados obtenidos fueron incluso mejores que en el enfoque anterior. La metaheurística VNS volvió a destacar como la más eficaz, alcanzando un calendario de 1.040.570 millas, lo que supone un *Gap* del -28,98 %, aunque con un coste computacional elevado de 19 minutos y 18 segundos (*Iteraciones máximas* = 400, *Iteraciones sin mejora* = 50). El Temple Simulado también mostró una buena actuación, logrando un calendario de 1.175.450 millas (-19,77 %) en un tiempo significativamente menor, de 1 minuto y 57 segundos (*Temp. inicial* = 100, *Temp. mínima* = 0.01, *Cambios* = 8, α = 0.9), lo que lo convierte en una alternativa eficiente, aunque con resultados ligeramente peores. Por el contrario, la Búsqueda Tabú volvió a demostrar una baja competitividad. Su mejor resultado fue un calendario de 1.293.980 millas (-11,66 %), pero con una mejora muy limitada, un progreso lento y un alto coste computacional, con un tiempo de ejecución de 44 minutos (*Iteraciones máximas* = 500, *Tamaño de la lista tabú* = 10).

6.3. Discusiones finales

Del análisis de todos los escenarios anteriores pueden extraerse varias conclusiones clave. En primer lugar, los calendarios estructurados ofrecen menor margen de maniobra. Esto dificulta la optimización, ya que las estructuras rígidas por jornadas limitan drásticamente la posibilidad de realizar movimientos eficientes para minimizar distancias.

En cambio, los calendarios no estructurados permiten una mayor flexibilidad en la planificación. Gracias a ello, se alcanzan soluciones más realistas y eficientes, llegando incluso a mejorar el calendario oficial en algunos casos, aunque, como se ha comentado previamente, esta compa-

ración debe hacerse con cautela. Además, es importante señalar que aplicar una metaheurística sobre el espacio de soluciones de un calendario de 82 jornadas es mucho más restrictivo que hacerlo sobre el espacio correspondiente a un calendario de 163 días. En este último, se permite una mayor cantidad de movimientos eficaces, lo que mejora notablemente la capacidad de optimización de las técnicas aplicadas. De ahí la mejoría de los resultados del Enfoque 2 con respecto al Enfoque 1.

Con respecto a las heurísticas, en el ámbito de calendarios estructurados, la mejor combinación fue la de la heurística 2 seguida de la heurística 3, especialmente útil en contextos complejos donde es necesario equilibrar la calidad de la solución con la eficiencia computacional. Sin embargo, a nivel global, la heurística 4 ha sido la mejor, ya que permite más libertad de movimientos.

En cuanto al rendimiento de las metaheurísticas, VNS (Búsqueda por Entornos Variables) fue la más eficaz de forma consistente. Aunque su coste computacional es elevado, su mejora es constante a lo largo del tiempo y alcanza las mejores soluciones globales. A diferencia del Temple Simulado, no se estanca en las fases finales, y por ello, incluso si se les da el mismo tiempo de ejecución, VNS logra superar al resto. Por otro lado, el Temple Simulado tiene la ventaja de proporcionar buenas soluciones iniciales rápidamente. Sin embargo, su mejora se vuelve muy lenta en las fases más tardías de enfriamiento, ya que se va quedando poco a poco en mínimos locales al producirse cada vez menos cambios. Por último, la Búsqueda Tabú ha sido claramente la menos competitiva. No solo ha tenido un coste computacional muy elevado, sino que su mejora es progresiva pero muy limitada, quedando lejos del rendimiento del resto de metaheurísticas.

Para finalizar, cabe destacar que el mejor calendario obtenido durante este proyecto, generado mediante VNS aplicando la heurística de inserción, presenta mejoras significativas frente al calendario oficial de la temporada 2022-2023. No solo se ha logrado una reducción global de más de 400.000 millas en los desplazamientos totales, sino que también se ha conseguido un reparto mucho más equilibrado entre conferencias. En el calendario oficial, los equipos del Oeste recorrian unas 120.000 millas más que los del Este. Sin embargo, en el calendario generado con VNS, esa diferencia se reduce a menos de 30.000 millas. Específicamente, los equipos de la Conferencia Este recorren de media 33.707,5 millas y los del Oeste 35.664 millas, frente a las 44.780 y 52.903 millas respectivamente del calendario original. Asimismo, se consiguió disminuir la desigualdad entre el equipo que más viaja y el que menos: Atlanta Hawks recorrió 28.230,1 millas, mientras que Boston Celtics fue el equipo con mayor recorrido con 40.661,4 millas. Esto implica que esta diferencia se redujo de aproximadamente 20.000 millas en el calendario oficial a solo 12.000 millas en esta versión optimizada. Todo ello no solo implica una mejora logística general, sino que también promueve una mayor equidad entre los equipos, reduciendo desigualdades en el calendario. No obstante, cabe resaltar de nuevo que el propósito que persigue la NBA para confeccionar los calendarios no es solo el de minimizar la distancia de los desplazamientos.

Capítulo 7

Conclusiones y trabajo futuro

La planificación de un calendario como el de la NBA ha demostrado ser un problema de notable dificultad tanto desde el punto de vista matemático como computacional, especialmente por la cantidad de variables, restricciones y criterios a tener en cuenta.

Se ha trabajado con distintos enfoques para abordar este problema, comenzando por modelos matemáticos exactos que permiten representar fielmente las restricciones reales del calendario. Se ha probado que la resolución de estos modelos presenta limitaciones computacionales significativas debido al tamaño del problema y la carga que supone para el optimizador Gurobi.

Ante estas dificultades, se exploraron métodos alternativos basados en heurísticas y metaheurísticas que aunque no garantizan la optimalidad, han demostrado ser eficaces para encontrar soluciones de alta calidad en un tiempo muy razonable. De hecho, se ha logrado construir un calendario que mejora la distancia total recorrida por los equipos respecto a la planificación oficial de la NBA para la temporada 2022-2023, superando el umbral de 1.465.250 millas.

El análisis detallado ha revelado que la metaheurística VNS ha resultado ser la más eficaz, aunque costosa computacionalmente. El Temple Simulado ofrece buenos resultados en menos tiempo, pero con menor calidad final. En cambio, la Búsqueda Tabú ha mostrado un rendimiento claramente inferior en todos los experimentos.

A su vez, se ha observado que aplicar metaheurísticas al espacio de soluciones de calendarios estructurados por jornadas ofrece menos margen de mejora que hacerlo sobre el espacio más flexible de calendarios no estructurados, distribuidos en 163 días. Este último permite realizar movimientos más eficaces y ha conducido a resultados notablemente mejores en términos de minimización de distancias.

Además de los resultados obtenidos, este proyecto ha permitido desarrollar competencias clave en el ámbito de la programación matemática, el uso de herramientas de optimización como Gurobi y el diseño de algoritmos heurísticos y metaheurísticos. Se ha trabajado con técnicas avanzadas y se ha adquirido una perspectiva más amplia sobre la importancia real que tiene una buena planificación en el deporte profesional, tanto desde el punto de vista logístico como desde el humano y económico.

En conclusión, aunque el problema del calendario NBA es altamente complejo, ha sido posible diseñar soluciones razonables y eficientes mediante técnicas adecuadas.

7.1. Trabajo futuro

A lo largo del desarrollo de este proyecto han surgido varias líneas de trabajo futuro que podrían abordarse en el futuro, tanto desde el punto de vista técnico como desde el impacto social, económico y deportivo que genera el calendario de la NBA.

En primer lugar, se han evidenciado ciertas limitaciones computacionales al ejecutar el segundo modelo de programación matemática, especialmente debido a la complejidad que implica minimizar las distancias recorridas por los equipos durante la temporada regular. En este sentido, se podría estudiar un refinamiento del modelo presentado, basado en la inclusión de restricciones más fuertes que permitan aproximar la envolvente convexa del conjunto de soluciones enteras factibles del problema, con el fin de mejorar la eficiencia y calidad de las soluciones obtenidas.

A su vez, como línea de mejora inmediata, se sugiere la exploración de nuevas heurísticas que permitan construir mejores soluciones y faciliten el trabajo de las metaheurísticas. En paralelo, también sería recomendable refinar la implementación de la Búsqueda Tabú en busca de mejorar significativamente su rendimiento.

Además, este trabajo ha servido de base para plantear nuevas variantes del problema original, abriendo la puerta a diferentes enfoques de confección de calendarios. Una de las líneas más prometedoras consiste en aprovechar el modelo y los algoritmos de minimización de distancias para aplicar criterios medioambientales, como por ejemplo la minimización de la huella de carbono generada por los desplazamientos. Para ello, sería necesario recopilar información específica sobre el consumo energético y las emisiones contaminantes de los aviones chárter utilizados por las franquicias de la NBA. A partir de esos datos, se podría reformular la función objetivo del modelo, sustituyendo las distancias por estimaciones de emisiones de CO₂, favoreciendo así una competición más sostenible.

Otra línea futura de trabajo tiene que ver con el bienestar físico de los jugadores. Actualmente, el calendario puede implicar secuencias de partidos exigentes con poco descanso entre ellos, lo que incrementa el riesgo de lesiones, afectando negativamente al rendimiento deportivo. Un modelo que priorice la salud de los jugadores podría incorporar restricciones adicionales que eviten partidos en días consecutivos o que garanticen un mínimo de descanso entre encuentros. Esto no solo contribuiría al cuidado físico de los deportistas, sino que también mejoraría la calidad del espectáculo, repercutiendo positivamente en la imagen de la liga.

En relación a la equidad competitiva, una idea interesante sería rediseñar el calendario para que todos los equipos de nivel similar tuvieran calendarios similares en cuanto a dificultad. Para ello, se podrían utilizar datos históricos recientes acerca de la competitividad de los equipos o predicciones previas a la temporada, como las proporcionadas por las casas de apuestas, para asignar un nivel de competitividad a cada equipo.

Por otro lado, el papel de los derechos televisivos también es un aspecto fundamental a considerar en la elaboración del calendario. Como se describió en el Capítulo 1, la NBA genera gran parte de sus ingresos a través de retransmisiones nacionales e internacionales. Se podría investigar en mayor profundidad cómo se distribuyen estos ingresos y cómo afectan a la liga, con el fin de diseñar un calendario que los optimice. Esto podría incluir programación estratégica de partidos relevantes en horarios de máxima audiencia, concentración de encuentros durante los fines de semana e incluso la adecuación a franjas horarias internacionales para captar audiencia en mercados clave como Europa o Asia.

Finalmente, todas estas líneas de investigación podrían combinarse en un modelo más completo que tenga en cuenta de manera simultánea todos estos objetivos mencionados con anterioridad. Para ello, podría recurrirse al uso de funciones objetivo ponderadas que permitan equilibrar estos aspectos en función de las prioridades de la competición, con el fin de diseñar el calendario más justo, eficiente y realista posible.

Apéndice A

Archivos desarrollados

En este apéndice se presenta una breve descripción del contenido de los distintos archivos incluidos en el repositorio fuente de este trabajo: <https://github.com/javia102/TFGMates>.

A.1. Datos

Contienen los datos principales acerca de los equipos que forman la liga, la matriz de distancias, el número de equipos por división, por conferencia, número de jornadas, etc.

- `datosNBA.h`: Datos principales correspondiente a la liga de 30 equipos y 82 jornadas.
- `datosNBA_4eq.h`: Datos principales correspondiente a la liga de 4 equipos y 12 jornadas.
- `datosNBA_8eq.h`: Datos principales correspondiente a la liga de 8 equipos y 20 jornadas.
- `datosNBA163.h`: Datos principales correspondiente a la liga de 30 equipos y 163 días.

A.2. Calendarios iniciales

Archivos de texto que contienen los calendarios base utilizados como punto de partida (producidos por el Modelo 1) para aplicar las heurísticas y metaheurísticas. Están codificados mediante los índices de los equipos, donde cada línea representa el itinerario correspondiente a un equipo a lo largo de la temporada.

- `calendario_balanceado_lyv.txt`: Calendario generado por el Modelo 1 correspondiente a la liga de 30 equipos.
- `calendario_4eq.txt`: Calendario generado por el Modelo 1 correspondiente a la liga de 4 equipos.
- `calendario_8eq.txt`: Calendario generado por el Modelo 1 correspondiente a la liga de 30 equipos.
- `calendario_22_23.txt`: Calendario correspondiente a la temporada oficial 2022-2023.
- `mejor_calendario_82.txt`: Mejor calendario estructurado obtenido mediante VNS correspondiente a la liga de 30 equipos.

A.3. Código fuente

Archivos en C++ que contienen la implementación de los modelos matemáticos, heurísticas y metaheurísticas utilizadas en el desarrollo del proyecto.

A.3.1. Modelos

- `CalendarioNBA_Mod1.cpp`: Modelo 1 correspondiente a la liga de 30 equipos.
- `CalendarioNBA_Mod2.cpp`: Modelo 2 correspondiente a la liga de 30 equipos.
- `CalendarioNBA_Mod1_4eq.cpp`: Modelo 1 correspondiente a la liga de 4 equipos.
- `CalendarioNBA_Mod2_4eq.cpp`: Modelo 2 correspondiente a la liga de 4 equipos.
- `CalendarioNBA_Mod1_8eq.cpp`: Modelo 1 correspondiente a la liga de 8 equipos.
- `CalendarioNBA_Mod2_8eq.cpp`: Modelo 2 correspondiente a la liga de 8 equipos.

A.3.2. Heurísticas

- `Heuristica_1.cpp`: Heurística 1 correspondiente a la liga de 30 equipos.
- `Heuristica_1_4eq.cpp`: Heurística 1 correspondiente a la liga de 4 equipos.
- `Heuristica_1_8eq.cpp`: Heurística 1 correspondiente a la liga de 8 equipos.
- `Heuristica_2.cpp`: Heurística 2 correspondiente a la liga de 30 equipos.
- `Heuristica_2_4eq.cpp`: Heurística 2 correspondiente a la liga de 4 equipos.
- `Heuristica_2_8eq.cpp`: Heurística 2 correspondiente a la liga de 8 equipos.
- `Heuristica_3.cpp`: Heurística 1 correspondiente a la liga de 30 equipos.
- `Heuristica_3_4eq.cpp`: Heurística 1 correspondiente a la liga de 4 equipos.
- `Heuristica_3_8eq.cpp`: Heurística 1 correspondiente a la liga de 8 equipos.
- `Heuristica_1_2_3.cpp`: Combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 30 equipos.
- `Heuristica_1_2_3_4eq.cpp`: Combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 4 equipos.
- `Heuristica_1_2_3_8eq.cpp`: Combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 8 equipos.
- `Heuristica_2_3.cpp`: Combinación de heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- `Heuristica_2_3_4eq.cpp`: Combinación de heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- `Heuristica_2_3_8eq.cpp`: Combinación de heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- `Heuristica_3_2.cpp`: Combinación de heurísticas 3 y 2 correspondiente a la liga de 30 equipos.
- `Heuristica_3_2_4eq.cpp`: Combinación de heurísticas 3 y 2 correspondiente a la liga de 4 equipos.
- `Heuristica_3_2_8eq.cpp`: Combinación de heurísticas 3 y 2 correspondiente a la liga de 8 equipos.

- `Insercion.cpp`: Heurística 4 sin restricciones correspondiente a la liga de 30 equipos.
- `Insercion_desc.cpp`: Heurística 4 con restricciones de descansos y partidos seguidos correspondiente a la liga de 30 equipos.
- `Insercion_balan.cpp`: Heurística 4 añadiendo restricciones de diferencia de partidos entre dos equipos cualesquiera correspondiente a la liga de 30 equipos.
- `Insercion_lyv.cpp`: Heurística 4 añadiendo restricciones de diferencia de partidos como local y visitante correspondiente a la liga de 30 equipos.

A.3.3. Metaheurísticas

A.3.3.1. Temple simulado

- `TS_h2.cpp`: Temple Simulado con heurística 2 correspondiente a la liga de 30 equipos.
- `TS_h2_4eq.cpp`: Temple Simulado con heurística 2 correspondiente a la liga de 4 equipos.
- `TS_h2_8eq.cpp`: Temple Simulado con heurística 2 correspondiente a la liga de 8 equipos.
- `TS_h3.cpp`: Temple Simulado con heurística 3 correspondiente a la liga de 30 equipos.
- `TS_h3_4eq.cpp`: Temple Simulado con heurística 3 correspondiente a la liga de 4 equipos.
- `TS_h3_8eq.cpp`: Temple Simulado con heurística 3 correspondiente a la liga de 8 equipos.
- `TS_h2_3.cpp`: Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- `TS_h2_3_4eq.cpp`: Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- `TS_h2_3_8eq.cpp`: Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- `TS_h4.cpp`: Temple Simulado con heurística 4 correspondiente a la liga de 30 equipos.

A.3.3.2. VNS

- `VNS_h2.cpp`: VNS con heurística 2 correspondiente a la liga de 30 equipos.
- `VNS_h2_4eq.cpp`: VNS con heurística 2 correspondiente a la liga de 4 equipos.
- `VNS_h2_8eq.cpp`: VNS con heurística 2 correspondiente a la liga de 8 equipos.
- `VNS_h3.cpp`: VNS con heurística 3 correspondiente a la liga de 30 equipos.
- `VNS_h3_4eq.cpp`: VNS con heurística 3 correspondiente a la liga de 4 equipos.
- `VNS_h3_8eq.cpp`: VNS con heurística 3 correspondiente a la liga de 8 equipos.
- `VNS_h2_3.cpp`: VNS con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- `VNS_h2_3_4eq.cpp`: VNS con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- `VNS_h2_3_8eq.cpp`: VNS con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- `VNS_h4.cpp`: VNS con heurística 4 correspondiente a la liga de 30 equipos.

A.3.3.3. Búsqueda Tabú

- `Tabu_h2.cpp`: Búsqueda Tabú con heurística 2 correspondiente a la liga de 30 equipos.
- `Tabu_h2_4eq.cpp`: Búsqueda Tabú con heurística 2 correspondiente a la liga de 4 equipos.
- `Tabu_h2_8eq.cpp`: Búsqueda Tabú con heurística 2 correspondiente a la liga de 8 equipos.
- `Tabu_h3.cpp`: Búsqueda Tabú con heurística 3 correspondiente a la liga de 30 equipos.
- `Tabu_h3_4eq.cpp`: Búsqueda Tabú con heurística 3 correspondiente a la liga de 4 equipos.
- `Tabu_h3_8eq.cpp`: Búsqueda Tabú con heurística 3 correspondiente a la liga de 8 equipos.
- `Tabu_h2_3.cpp`: Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- `Tabu_h2_3_4eq.cpp`: Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- `Tabu_h2_3_8eq.cpp`: Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- `Tabu_h4.cpp`: Búsqueda Tabú con heurística 4 correspondiente a la liga de 30 equipos.

A.4. Resultados

Archivos que recogen el mejor calendario obtenido por cada heurística y metaheurística. Se incluye a su vez el resultado del único modelo exacto de minimización que se logró resolver. Además, en el caso de las metaheurísticas, se incluyen ficheros `resultados_X.txt` que detallan los resultados obtenidos para cada combinación de parámetros explorada mediante *Grid Search*, donde X indica la metaheurística correspondiente acompañada del número de equipos que componen la liga.

A.4.1. Modelos

- `Modelo_2_4eq.txt`: Calendario resultante de la ejecución del Modelo 2 en el caso reducido de 4 equipos.

A.4.2. Heurísticas

- `Heuristica_1.txt`: Mejor calendario heurística 1 correspondiente a la liga de 30 equipos.
- `Heuristica_1_4eq.txt`: Mejor calendario heurística 1 correspondiente a la liga de 4 equipos.
- `Heuristica_1_8eq.txt`: Mejor calendario heurística 1 correspondiente a la liga de 8 equipos.
- `Heuristica_2.txt`: Mejor calendario heurística 2 correspondiente a la liga de 30 equipos.
- `Heuristica_2_4eq.txt`: Mejor calendario heurística 2 correspondiente a la liga de 4 equipos.
- `Heuristica_2_8eq.txt`: Mejor calendario heurística 2 correspondiente a la liga de 8 equipos.
- `Heuristica_3.txt`: Mejor calendario heurística 3 correspondiente a la liga de 30 equipos.

- **Heuristica_3_4eq.txt**: Mejor calendario heurística 3 correspondiente a la liga de 4 equipos.
- **Heuristica_3_8eq.txt**: Mejor calendario heurística 3 correspondiente a la liga de 8 equipos.
- **Heuristica_1_2_3.txt**: Mejor calendario combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 30 equipos.
- **Heuristica_1_2_3_4eq.txt**: Mejor calendario combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 4 equipos.
- **Heuristica_1_2_3_8eq.txt**: Mejor calendario combinación de heurísticas 1, 2 y 3 correspondiente a la liga de 8 equipos.
- **Heuristica_2_3.txt**: Mejor calendario combinación de heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- **Heuristica_2_3_4eq.txt**: Mejor calendario combinación de heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- **Heuristica_2_3_8eq.txt**: Mejor calendario combinación de heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- **Heuristica_3_2.txt**: Mejor calendario combinación de heurísticas 3 y 2 correspondiente a la liga de 30 equipos.
- **Heuristica_3_2_4eq.txt**: Mejor calendario combinación de heurísticas 3 y 2 correspondiente a la liga de 4 equipos.
- **Heuristica_3_2_8eq.txt**: Mejor calendario combinación de heurísticas 3 y 2 correspondiente a la liga de 8 equipos.
- **Insercion.txt**: Mejor calendario heurística 4 sin restricciones correspondiente a la liga de 30 equipos.
- **Insercion_desc.txt**: Mejor calendario heurística 4 con restricciones de descansos y partidos seguidos correspondiente a la liga de 30 equipos.
- **Insercion_balan.txt**: Mejor calendario heurística 4 añadiendo restricciones de diferencia de partidos entre dos equipos cualesquiera correspondiente a la liga de 30 equipos.
- **Insercion_lyv.txt**: Mejor calendario heurística 4 añadiendo restricciones de diferencia de partidos como local y visitante correspondiente a la liga de 30 equipos.

A.4.3. Metaheurísticas

A.4.3.1. Temple Simulado

- **TS_h2.txt**: Mejor calendario Temple Simulado con heurística 2 correspondiente a la liga de 30 equipos.
- **resultados_ts_h2.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 2 en una liga de 30 equipos.
- **TS_h2_4eq.txt**: Mejor calendario Temple Simulado con heurística 2 correspondiente a la liga de 4 equipos.

- **resultados_ts_h2_4eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 2 en una liga de 4 equipos.
- **TS_h2_8eq.txt**: Mejor calendario Temple Simulado con heurística 2 correspondiente a la liga de 8 equipos.
- **resultados_ts_h2_8eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 2 en una liga de 8 equipos.
- **TS_h3.txt**: Mejor calendario Temple Simulado con heurística 3 correspondiente a la liga de 30 equipos.
- **resultados_ts_h3.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 3 en una liga de 30 equipos.
- **TS_h3_4eq.txt**: Mejor calendario Temple Simulado con heurística 3 correspondiente a la liga de 4 equipos.
- **resultados_ts_h3_4eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 3 en una liga de 4 equipos.
- **TS_h3_8eq.txt**: Mejor calendario Temple Simulado con heurística 3 correspondiente a la liga de 8 equipos.
- **resultados_ts_h3_8eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 3 en una liga de 8 equipos.
- **TS_h2_3.txt**: Mejor calendario Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- **resultados_ts_h2_3.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurísticas 2 y 3 en una liga de 30 equipos.
- **TS_h2_3_4eq.txt**: Mejor calendario Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- **resultados_ts_h2_3_4eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurísticas 2 y 3 en una liga de 4 equipos.
- **TS_h2_3_8eq.txt**: Mejor calendario Temple Simulado con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- **resultados_ts_h2_3_8eq.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurísticas 2 y 3 en una liga de 8 equipos.
- **TS_h4.txt**: Mejor calendario Temple Simulado con heurística 4 correspondiente a la liga de 30 equipos.
- **resultados_ts_h4.txt**: Resultados para distintas configuraciones de parámetros para Temple Simulado con heurística 4 en una liga de 30 equipos.

A.4.3.2. VNS

- **VNS_h2.txt**: Mejor calendario VNS con heurística 2 correspondiente a la liga de 30 equipos.
- **resultados_vns_h2.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 2 en una liga de 30 equipos.

- **VNS_h2_4eq.txt**: Mejor calendario VNS con heurística 2 correspondiente a la liga de 4 equipos.
- **resultados_vns_h2_4eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 2 en una liga de 4 equipos.
- **VNS_h2_8eq.txt**: Mejor calendario VNS con heurística 2 correspondiente a la liga de 8 equipos.
- **resultados_vns_h2_8eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 2 en una liga de 8 equipos.
- **VNS_h3.txt**: Mejor calendario VNS con heurística 3 correspondiente a la liga de 30 equipos.
- **resultados_vns_h3.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 3 en una liga de 30 equipos.
- **VNS_h3_4eq.txt**: Mejor calendario VNS con heurística 3 correspondiente a la liga de 4 equipos.
- **resultados_vns_h3_4eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 3 en una liga de 4 equipos.
- **VNS_h3_8eq.txt**: Mejor calendario VNS con heurística 3 correspondiente a la liga de 8 equipos.
- **resultados_vns_h3_8eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 3 en una liga de 8 equipos.
- **VNS_h2_3.txt**: Mejor calendario VNS con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- **resultados_vns_h2_3.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurísticas 2 y 3 en una liga de 30 equipos.
- **VNS_h2_3_4eq.txt**: Mejor calendario VNS con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- **resultados_vns_h2_3_4eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurísticas 2 y 3 en una liga de 4 equipos.
- **VNS_h2_3_8eq.txt**: Mejor calendario VNS con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- **resultados_vns_h2_3_8eq.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurísticas 2 y 3 en una liga de 8 equipos.
- **VNS_h4.txt**: Mejor calendario VNS con heurística 4 correspondiente a la liga de 30 equipos.
- **resultados_vns_h4.txt**: Resultados para distintas configuraciones de parámetros para VNS con heurística 4 en una liga de 30 equipos.

A.4.3.3. Búsqueda Tabú

- **Tabu_h2.txt**: Mejor calendario Búsqueda Tabú con heurística 2 correspondiente a la liga de 30 equipos.
- **resultados_tabu_h2.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 2 en una liga de 30 equipos.

- **Tabu_h2_4eq.txt**: Mejor calendario Búsqueda Tabú con heurística 2 correspondiente a la liga de 4 equipos.
- **resultados_tabu_h2_4eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 2 en una liga de 4 equipos.
- **Tabu_h2_8eq.txt**: Mejor calendario Búsqueda Tabú con heurística 2 correspondiente a la liga de 8 equipos.
- **resultados_tabu_h2_8eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 2 en una liga de 8 equipos.
- **Tabu_h3.txt**: Mejor calendario Búsqueda Tabú con heurística 3 correspondiente a la liga de 30 equipos.
- **resultados_tabu_h3.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 3 en una liga de 30 equipos.
- **Tabu_h3_4eq.txt**: Mejor calendario Búsqueda Tabú con heurística 3 correspondiente a la liga de 4 equipos.
- **resultados_tabu_h3_4eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 3 en una liga de 4 equipos.
- **Tabu_h3_8eq.txt**: Mejor calendario Búsqueda Tabú con heurística 3 correspondiente a la liga de 8 equipos.
- **resultados_tabu_h3_8eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 3 en una liga de 8 equipos.
- **Tabu_h2_3.txt**: Mejor calendario Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 30 equipos.
- **resultados_tabu_h2_3.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurísticas 2 y 3 en una liga de 30 equipos.
- **Tabu_h2_3_4eq.txt**: Mejor calendario Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 4 equipos.
- **resultados_tabu_h2_3_4eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurísticas 2 y 3 en una liga de 4 equipos.
- **Tabu_h2_3_8eq.txt**: Mejor calendario Búsqueda Tabú con heurísticas 2 y 3 correspondiente a la liga de 8 equipos.
- **resultados_tabu_h2_3_8eq.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurísticas 2 y 3 en una liga de 8 equipos.
- **Tabu_h4.txt**: Mejor calendario Búsqueda Tabú con heurística 4 correspondiente a la liga de 30 equipos.
- **resultados_tabu_h4.txt**: Resultados para distintas configuraciones de parámetros para Búsqueda Tabú con heurística 4 en una liga de 30 equipos.

Apéndice B

Detalles de implementación

En este apéndice se presenta una descripción detallada de las estructuras de datos principales empleadas en la implementación de las heurísticas y metaheurísticas utilizadas en el proyecto. Estas estructuras han sido diseñadas para facilitar la representación del calendario, los equipos, los recorridos y las restricciones asociadas al problema.

B.1. Estructuras de datos

B.1.1. Información de los equipos

Para almacenar la información principal de cada equipo, se ha definido una estructura (**struct**) denominada **InfoEquipo**. Esta estructura contiene los siguientes campos:

```
struct InfoEquipo {
    int id;                                     // Identificador numerico del equipo
    string nombre;                               // Nombre del equipo
    string conferencia;                         // Conferencia a la que pertenece
    string division;                             // Division dentro de su conferencia
    vector<int> rivales_division;             // Rivales de su misma division
    vector<int> rivales_conf1;                 // Rivales misma conferencia con 4p
    vector<int> rivales_conf2;                 // Rivales con 3 partidos (2c, 1f)
    vector<int> rivales_conf3;                 // Rivales con 3 partidos (1c, 2f)
    vector<int> rivales_interconf;            // Rivales de la otra conferencia
    int partidos_jug;                           // Partidos jugados acumulados
};
```

Una vez definida esta estructura, se construye un vector de estructuras **InfoEquipo** denominado **equipos**, donde cada elemento representa la información de un equipo concreto. Esta estructura permite acceder de forma directa y eficiente a los datos relacionados con cada equipo durante todo el proceso.

B.1.2. Matriz de distancias

Las distancias entre las ciudades de los equipos se almacenan en una matriz de tipo **double**, denotada como **distanciasNBA[i][j]**. Esta matriz representa la distancia en millas entre los equipos *i* y *j*, tal y como se detalló en el Capítulo 2.

Esta matriz se utiliza para calcular los recorridos y evaluar la calidad de los calendarios generados.

B.1.3. Recorridos y rivales

Para representar los desplazamientos de los equipos durante la temporada, se define la matriz de enteros ([int](#)) `viajes[i][k]`, donde:

- i representa el equipo
- k representa la jornada (o día, en el caso del calendario no estructurado)
- `viajes[i][k]` almacena el identificador de la ciudad en la que se encuentra el equipo i en la jornada o día k

Adicionalmente, se crea otra matriz de enteros denominada `rivales[i][k]`, que guarda el identificador del equipo contra el que juega el equipo i en la jornada k .

Ambas matrices permiten reconstruir el calendario de manera eficiente.

Cabe destacar que en los calendarios organizados por días, en los que los equipos no juegan todos los días, se representa como -1 el valor de `viajes[i][k]` y `rivales[i][k]` si el equipo i no juega en el día k .

B.1.4. Balance del calendario

En los calendarios diarios (no estructurados), se incorporan restricciones adicionales para garantizar el equilibrio competitivo entre los equipos. Para implementar estas restricciones, se crean las siguientes estructuras de datos:

- `partidos_dia[i][k]`: matriz de enteros que guarda en cada posición el número acumulado de partidos jugados por el equipo i hasta el día k
- `maximo_dia[k]`: vector de enteros que guarda en cada posición el número máximo de partidos jugados por un equipo hasta el día k
- `minimo_dia[k]`: vector de enteros que guarda en cada posición el número mínimo de partidos jugados por un equipo hasta el día k

Estas estructuras permiten llevar la cuenta de la cantidad de partidos que ha jugado cada equipo en todo momento y así poder verificar la restricción de que la diferencia entre partidos jugados por cualquier par de equipos nunca supere 4 partidos.

Bibliografía

- [1] 2022-23 NBA Schedule — Basketball-Reference.com — basketball-reference.com. https://www.basketball-reference.com/leagues/NBA_2023_games.html. [Accessed 19-06-2025].
- [2] Seyed Erfan Alesahebfosoul. «Balancing the Game: Comparative Analysis of Single Heuristics and Adaptive Heuristic Approaches for Sports Scheduling Problem». Tesis de mtría. The University of Bergen, 2023.
- [3] Renjun Bao. «Time relaxed round robin tournament and the NBA scheduling problem». En: (2009).
- [4] James C Bean y John R Birge. «Reducing travelling costs and player fatigue in the national basketball association». En: *Interfaces* 10.3 (1980), págs. 98-102.
- [5] EK Burke et al. «Applying Heuristic Methods to Schedule Sports Competitions on Multiple Venues Extended Abstract». En: (2004). Extended abstract.
- [6] Robert Thomas Campbell y DS Chen. «A minimum distance basketball scheduling problem». En: *Management science in sports* 4 (1976), págs. 15-26.
- [7] Shengqi Chai et al. «National Basketball Association Scheduling Simulation». En: (2016).
- [8] Angelos Dimitzas et al. «A pragmatic approach for solving the sports scheduling problem». En: *Proc. 13th Int. Conf. Pract. Theory Autom. Timetabling, PATAT*. Vol. 3. 2022, págs. 195-207.
- [9] *El origen de la NBA: Cómo fue el inicio de la liga desde dentro* — spain.id.nba.com. <https://spain.id.nba.com/reportajes/el-origen-de-la-liga>. [Accessed 24-06-2025].
- [10] Jacques A Ferland y Charles Fleurent. «Computer aided scheduling for a sport league». En: *INFOR: Information Systems and Operational Research* 29.1 (1991), págs. 14-25.
- [11] Winston Fraser. «The role of computer simulation in building the National Hockey League Schedule». En: *IBM Canada Limited, Montreal, Quebec, Canada* (1982).
- [12] Fred Glover y Manuel Laguna. *Tabu search*. Springer, 1998.
- [13] Pierre Hansen y Nenad Mladenovic. «A tutorial on variable neighborhood search». En: *Les Cahiers du GERAD ISSN* 711 (2003), pág. 2440.
- [14] Pierre Hansen y Nenad Mladenović. «Variable neighborhood search: Principles and applications». En: *European journal of operational research* 130.3 (2001), págs. 449-467.
- [15] M Henz. «Playing with constraint programming and large neighborhood search for traveling tournaments». En: *Proceedings PATAT*. Vol. 2004. 2004, págs. 23-32.
- [16] *How the NBA Schedule is Made* — nbastuffer.com. <https://www.nbastuffer.com/analytics101/how-the-nba-schedule-is-made/>. [Accessed 24-06-2025].
- [17] Imgur. *imgur.com* — imgur.com. <https://imgur.com/c4qqI0G>. [Accessed 12-06-2025].
- [18] Hua-An Lu y Wei-Cheng Chen. «A tournament scheduling model for National Basketball Association (NBA) regular season games». En: *Journal of Advanced Research in Social Sciences and Humanities* 2.1 (2017), págs. 17-26.
- [19] *NBA Map — Teams — Logos - Sport League Maps* — sportleaguemaps.com. <https://sportleaguemaps.com/basketball/nba/>. [Accessed 19-06-2025].
- [20] George L Nemhauser y Michael A Trick. «Scheduling a major college basketball conference». En: *Operations research* 46.1 (1998), págs. 1-8.
- [21] *Simulated annealing* - Wikipedia — en.wikipedia.org. https://en.wikipedia.org/wiki/Simulated_annealing. [Accessed 24-06-2025].
- [22] Peter JM Van Laarhoven et al. *Simulated annealing*. Springer, 1987.
- [23] Mike Wright. «Timetabling county cricket fixtures using a form of tabu search». En: *Journal of the operational research society* 45.7 (1994), págs. 758-770.
- [24] Mike B Wright. «Scheduling fixtures for basketball New Zealand». En: *Computers & Operations Research* 33.7 (2006), págs. 1875-1893.
- [25] Feng-Cheng Yang. «NBA sports game scheduling problem and GA-based Solver». En: *2017 International Conference on Industrial Engineering, Management Science and Application (ICIMSA)*. IEEE. 2017, págs. 1-5.