

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías Industriales

Integración de planificación de consultas y quirófanos en el sector sanitario

Autor: María de los Ángeles Ramírez Rojas

Tutor: Víctor Fernández-Viagas Escudero

Tutor Externo: José M. Molina Pariente

**Dpto. Organización Industrial y Gestión de
Empresas I
Escuela Técnica Superior de Ingeniería**

Sevilla, 2020



Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

Integración de planificación de consultas y quirófanos en el sector sanitario

Autor:

María de los Ángeles Ramírez Rojas

Tutor:

Víctor Fernández-Viagas Escudero

Tutor Externo:

José M. Molina Pariente

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020

Proyecto Fin de Carrera: Integración de planificación de consultas y quirófanos en el sector sanitario

Autor: María de los Ángeles Ramírez
Rojas

Tutor: Víctor Fernández-Viagas Escudero

Tutor José M. Molina Pariente
externo:

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal

A mi familia

A mis maestros

AGRADECIMIENTOS

Me gustaría dedicar este trabajo a mis padres, por recordarme cada día lo que es verdaderamente importante durante estos años de esfuerzo, a mi hermana Julia, por ser mi pilar fundamental y a Javi, por su paciencia y apoyo incondicional.

También quisiera agradecer sinceramente a mis tutores por darme la oportunidad de aplicar los conocimientos adquiridos en el grado al sector de la salud, que personalmente considero tan humano y gratificante.

M^aÁngeles Ramírez Rojas

ÍNDICE

Agradecimientos	viii
Índice	ix
Índice de Tablas	xi
Índice de figuras	xii
1 Introducción	1
1.1 Objeto del proyecto	1
1.2 Justificación	1
1.3 Sumario	3
2 Comparación de la gestión de operaciones en el sistema industrial y el sanitario	5
2.1 Introducción a la gestión de operaciones	5
2.2 Similitud en los objetivos	6
2.3 Diferencias principales entre ambos entornos	6
2.4 Introducción a la programación de operaciones	7
2.5 Diferencia entre planificación y programación	9
3 Antecedentes	11
3.1 Métodos más comunes en la gestión de operaciones en sanidad	11
3.2 Planificación y programación del quirófano: revisión de la literatura	11
4 Descripción del problema	15
4.1 Limitaciones de los recursos	15
4.2 Descripción del sistema y tipos de pacientes	15
4.3 Función objetivo	15
5 Metodología de resolución	17
5.1 Modelado	17
5.1.1 Notación de los modelos	17
5.1.2 Datos básicos	18
5.1.3 Otros datos	18
5.1.4 Características de las máquinas (α)	18
5.1.5 Restricciones (β)	19
5.1.6 Función objetivo (γ)	19
5.2 Presentación de los posibles métodos de programación	20
5.2.1 Métodos exactos	20
5.2.2 Métodos aproximados	20
5.2.3 Método elegido: Programación Lineal Entera (ILP)	21

5.3	Implementación y optimización del modelo	21
5.3.1	Lenguaje de programación	21
5.3.2	Software de optimización	22
5.4	Evaluación del modelo: Estudio de simulación	22
6	Complejidad	25
6.1	Complejidad computacional	25
6.2	Complejidad de los entornos reales	25
7	Aplicación de la metodología	27
7.1	Modelo lineal Entero	27
7.2	Implementación del modelo	31
7.3	Diseño del banco de pruebas	32
7.3.1	Factores principales	32
7.3.2	Datos relevantes a las cirugías	33
7.3.3	Datos relevantes a los cirujanos	34
7.4	Experimentos computacionales	34
7.4.1	Análisis de los resultados en función de los recursos	35
7.4.2	Influencia del número de pacientes y cirujanos	37
7.5	Caso práctico: valor que toman las variables	37
8	Conclusiones	45
8.1	Líneas de futuro	46
	Referencias	47
	Anexos	49

ÍNDICE DE TABLAS

Tabla 1-1: Listas de espera de más de 6 meses en la Comunidad de Aragón (Gobierno de Aragón).	3
Tabla 2-1: Similitudes y diferencias entre el sistema industrial y sanitario (Vissers and Beech, 2005:27).	7
Tabla 3-1: Restricciones consideradas en la literatura (Cardoen y otros, 2009:17).	13
Tabla 7-1: Conjuntos, parámetros y variables usados en el modelo lineal entero de decisión.	27
Tabla 7-2: Factores considerados para la generación de datos de entrada.	33
Tabla 7-3: Parámetros de pacientes considerados para la generación de datos de entrada.	33
Tabla 7-4: Número medio de cirugías en función de la capacidad.	35
Tabla 7-5: Valor de las variables tras la llamada a la función.	35
Tabla 7-6: Resultados del modelo ILP para resolver el problema de decisión.	36
Tabla 7-7: Resultados del modelo ILP en función de Us .	36
Tabla 7-8: Resultados del modelo ILP en función de β y α .	37
Tabla 7-9: Datos propuestos.	38
Tabla 7-10: Datos de pacientes generados.	38
Tabla 7-11: Valor de X_{ich} .	39
Tabla 7-12: Valor de Z_{sch} .	40
Tabla 7-13: Valor de Y_{isch} .	41
Tabla 7-14: Valor de X'_{ijh} .	42
Tabla 7-15: Valor de Z'_{sjh} .	43
Tabla 7-16: Valor de Y'_{isjh} .	44

ÍNDICE DE FIGURAS

Figura 1-1: Evolución de las listas de espera quirúrgica del SNS (2016-2019) (Ministerio de Sanidad, Consumo y Bienestar Social).....	2
Figura 2-1: Proceso genérico en un sistema industrial (Elaboración propia).....	5
Figura 2-2: Proceso genérico en un hospital (Vissers and Beech, 2005: 2).	6
Figura 5-1: Procedimiento de resolución de problemas de programación (Apuntes de programación de operaciones)	17

1 INTRODUCCIÓN

Este primer capítulo introductorio describe el propósito del TFG, la motivación que ha llevado a su realización y finaliza con un resumen del contenido del mismo.

1.1 Objeto del proyecto

Este proyecto aborda el problema de planificación de la atención de pacientes en consultas y quirófanos de un hospital. En la primera etapa, el problema de decisión consiste en establecer para un conjunto de pacientes en lista de espera, una fecha disponible en la que serán atendidos en una consulta por un cirujano. En la segunda etapa, se asigna a estos pacientes una fecha de intervención y un quirófano adecuado, en el cual se llevará a cabo la operación por el mismo cirujano que lo atendió previamente.

Establecer una adecuada planificación a largo plazo facilita la posterior toma de decisiones. De aquí nace el objetivo de este proyecto, consistente en optimizar la programación avanzada dentro de los distintos niveles de decisión. Los posteriores niveles, que implican una programación a corto plazo, quedan fuera del ámbito del proyecto.

La función objetivo establecida busca minimizar la tardanza total en el paso por quirófano de los pacientes, teniendo en cuenta la prioridad clínica de cada cirugía, y asegurando a su vez el uso de los recursos siempre que estén disponibles.

Para llevarlo a cabo, en primer lugar se ha desarrollado el modelo lineal entero en base a las restricciones establecidas por el hospital para posteriormente programarlo en el lenguaje *C#* en *Microsoft Visual Studio*, empleando el software comercial *Gurobi* versión 9.0.1.

Una vez abordado el problema de forma exacta se concluye un buen funcionamiento para un caso de tamaño reducido, incrementándose la dificultad de resolución a medida que se aumentan los recursos, el horizonte de planificación y el tamaño de las listas de espera. Por tanto, se considera un buen punto de partida para el posterior desarrollo de algoritmos aproximados, que puedan ser aplicados en el entorno real de trabajo.

1.2 Justificación

Hoy en día los centros hospitalarios se ven sometidos a una presión creciente con el objetivo de ofrecer sus servicios al mínimo coste posible, garantizando la calidad asistencial en término de tiempos de espera. Esto se debe fundamentalmente a tres factores: los presupuestos restrictivos, el envejecimiento general de la población y la tendencia creciente de las listas de espera en los últimos años (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015). Este último aspecto puede apreciarse en el diagrama de la figura 1-1, en la que se observa un aumento de pacientes y del tiempo medio de espera a partir de 2018 y especialmente en el año 2019.

Por este motivo, establecer un nivel de decisión adecuado en los centros hospitalarios, puede tener un gran impacto en la mejora de la calidad de los servicios prestados, así como en la reducción de costes. Esto se cumple especialmente en el ámbito de los quirófanos, ya que es el recurso que consume la mayor parte del presupuesto (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015).

Esta idea se ve reforzada en el contexto sanitario actual, en el que destaca un gran incremento en las listas de espera quirúrgicas debido al parón asistencial que ha provocado la pandemia de la COVID-19, llegando a aumentar en algunas comunidades autónomas como la de Aragón hasta el 88% de su valor total a inicios de año, como se puede observar en la tabla 1-1. En esta situación excepcional es esencial contar con una planificación de los recursos efectiva, para reducir los efectos negativos, tanto

económicos como humanitarios, en el menor tiempo posible.

En este caso en concreto, partiendo del análisis del nivel de decisión tradicional que se ha llevado a cabo en la Unidad de Gestión Clínica de Cirugía Plástica y Grandes Quemados del Hospital “Virgen del Rocío” (Sevilla), se evidencia que una planificación de las cirugías basada únicamente en la experiencia sin el soporte de una herramienta de optimización, conduce a tamaños de listas de espera mayores, aumentando el tiempo de acceso de los pacientes (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015). Si es cierto que actualmente se implementan algoritmos para apoyar este proceso de decisión, este estudio surge de la necesidad de integrar en un único modelo la programación de los quirófanos junto con una etapa previa de consultas, dado que ambas etapas comparten los cirujanos disponibles en un servicio quirúrgico.

Por último, indicar que este es un campo de estudio actual que se encuentra en desarrollo. Hoy en día son pocas las instituciones que apoyan un sistema basado en la dirección de operaciones en el sector sanitario. Algunos ejemplos en Estados Unidos y Reino Unido son (Ortiga Fontgivell, 2013):

- El Institute of Health Improvement (IHI)
- La Joint Commission Accreditation of Health Care Organizations (JCAHO)
- El National Committee for Quality Assurance (NCQA)
- El Institute for Innovation and Improvement del NHS

En España actualmente no existe ninguna organización oficial de estas características.



Figura 1-1: Evolución de las listas de espera quirúrgica del SNS (2016-2019) (Ministerio de Sanidad, Consumo y Bienestar Social)

Tabla 1-1: Listas de espera de más de 6 meses en la Comunidad de Aragón (Gobierno de Aragón)

Especialidad Quirúrgica	Nº de pacientes Abril 2020	Nº de pacientes Enero 2020	Incremento
Angiología / Cirugía vascular	315	136	179
Cirugía Cardíaca	18	0	18
Cirugía Gral y de digestivo	1131	612	519
Cirugía Maxilofacial	214	88	126
Cirugía Pediátrica	198	62	136
Cirugía Plástica	151	100	51
Cirugía Torácica	4	1	3
Dermatología	11	8	3
Ginecología	13	7	6
Neurocirugía	558	429	129
Oftalmología	1274	140	1134
Otorrinolaringología	365	220	145
Traumatología	2548	1767	781
Urología	376	238	138
TOTAL	7176	3808	3368

1.3 Sumario

Este trabajo fin de grado se estructura en ocho capítulos y dos anexos que van a ser comentados en esta sección con la intención de facilitar la comprensión del problema a tratar, así como de exponer el procedimiento tenido en cuenta para su estudio y resolución.

El capítulo 2 surge con la idea de contextualizar la naturaleza del problema de decisión dentro del marco de la gestión de operaciones. A través de una comparación entre el sistema industrial y sanitario, en términos de objetivos y gestión, se puede concluir que los métodos existentes para el primero pueden ser aplicados de forma similar al segundo. En especial, se hace una introducción a la programación de operaciones, ya que es el área de estudio que ha servido de base para el desarrollo de la solución a este problema.

En el capítulo 3 se exponen los antecedentes relacionados con el problema en cuestión. Por un lado, se plantean de forma general los principales métodos de gestión de operaciones llevados a cabo en la actualidad en los centros sanitarios, y por otro lado, se realiza una revisión en particular de la literatura existente sobre la planificación y programación de los quirófanos, dando una visión de las distintas perspectivas que se pueden tener en cuenta al realizar estos estudios de investigación.

En el capítulo 4, se realiza una descripción específica del problema a estudiar, se definen las características de los recursos limitantes, los tipos de paciente y en general, se exponen los distintos requisitos exigidos por el entorno hospitalario que hay que tener en cuenta para el posterior modelado.

Una vez planteado el problema de decisión, en el capítulo 5 se pasa a realizar una descripción de la metodología de resolución seguida generalmente en los problemas de programación de la producción, indicando a lo largo de cada sección las similitudes con el problema que ocupa este proyecto. Además,

se plantean los posibles métodos de resolución del modelo (exactos y no exactos) especificando a continuación, la idoneidad del método elegido y los pasos que éste involucra. Los últimos pasos son la implementación en un lenguaje de programación, la optimización a través de un software específico y por último la evaluación del modelo mediante un estudio de simulación, profundizando en las características que se deben tener en cuenta para su correcta ejecución en el capítulo 7.

Antes de proceder a la aplicación de la metodología, el capítulo 6 se dedica a definir la complejidad del modelo. Esta complejidad es definida en base a distintas perspectivas que resultan interesantes para este estudio.

Tras la definición de la metodología elegida y comprendido su alcance, en el capítulo 7 se procede a su aplicación. Se dedica una sección al modelado del problema, a su implementación, a la generación del banco de pruebas y también se realiza un análisis de los resultados obtenidos. En la última sección se genera un caso práctico para visualizar el valor que toman las variables tras ejecutar el programa y poder comprobar el cumplimiento de las distintas restricciones.

En el capítulo 8 se recogen las conclusiones extraídas tras el estudio y se plantean posibles líneas de futuro.

En último lugar, se adjuntan dos anexos. El primer anexo es el código en C# del programa desarrollado para este Trabajo Fin de Grado y el segundo anexo es el fichero generado tras su ejecución, y que ha servido como base para el análisis de la solución y extracción de conclusiones.

2 COMPARACIÓN DE LA GESTIÓN DE OPERACIONES EN EL SISTEMA INDUSTRIAL Y EL SANITARIO

En este capítulo se pretende enmarcar el problema de decisión referente a la planificación de quirófanos y consultas en un hospital dentro de la gestión de operaciones. Haciendo una comparación entre el sistema industrial y sanitario, se justifica la posible adaptación de los métodos estudiados para la programación de la producción al ámbito de la salud. Esta conclusión es respaldada por el *Institute of Health Improvement* de EE.UU, ya que considera que muchas de las herramientas y métodos de gestión de operaciones procedentes de otros sectores pueden aplicarse con éxito a la atención sanitaria (Ortiga Fontgivell, 2013).

2.1 Introducción a la gestión de operaciones

El término gestión de operaciones se refiere a la planificación y el control de los procesos que transforman las entradas de un sistema en productos. Esta definición también se aplica a la gestión de operaciones de la salud. Si se considera la consulta individual médico/paciente, la entrada al proceso de consulta es un paciente con una solicitud de atención médica. La salida del proceso de consulta podría ser que el paciente sea diagnosticado, enviado a un proceso posterior, o curado (Vissers & Beech, 2005).

En la figura 2-2 se presenta una visión de la gestión de operaciones de un hospital individual, destacando tres procesos genéricos para transformar los insumos en productos: clínicos, de gestión y auxiliares. Comparando con la figura 2-1, que presenta un proceso genérico que podría seguirse en un sistema de producción industrial, es posible observar la similitud entre la esquemática de ambos entornos.

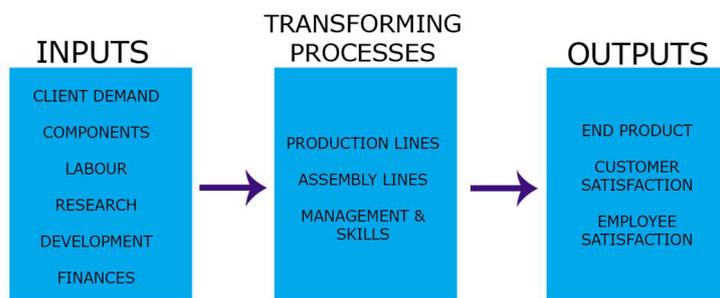


Figura 2-1: Proceso genérico en un sistema industrial (Elaboración propia)

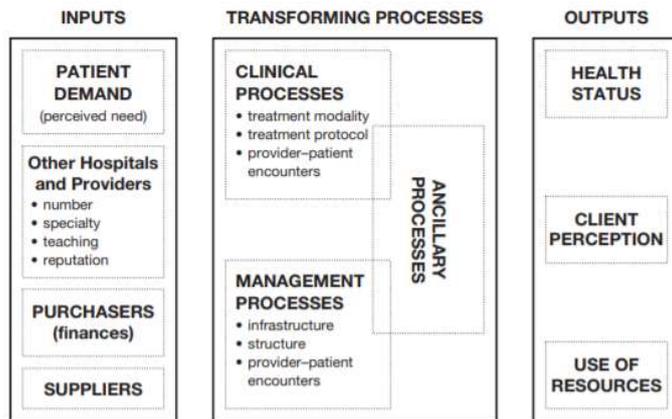


Figura 2-2: Proceso genérico en un hospital (Vissers and Beech, 2005: 2).

2.2 Similitud en los objetivos

La competencia en el sector industrial genera una gran presión sobre el rendimiento en términos de calidad, eficiencia y flexibilidad. Algunos de los objetivos generales de las empresas son la reducción del tiempo total de producción y entrega, los costes de entrega, la mejora de resultados e incremento de beneficios de la organización. La manufactura orientada a la logística ha contribuido en muchas circunstancias a mejorar el servicio de entrega al cliente a un menor coste (Vissers & Beech, 2005).

La atención sanitaria tiene ante sí retos similares: (Vissers & Beech, 2005)

- El incremento de la complejidad de los procesos, a través de las nuevas tecnologías y especialización, junto con una reducción asociada de la estancia media del paciente.
- La necesidad del uso eficiente de los recursos y de la reducción de costes.
- La presión creciente por mejorar la calidad de los servicios, relacionada en gran medida con los tiempos en lista de espera.
- La necesidad de controlar la carga de trabajo del personal (enfermeros, cirujanos...) para evitar los efectos negativos que podría tener una sobrecarga en las condiciones de trabajo.

2.3 Diferencias principales entre ambos entornos

Sin embargo, hay que distinguir que un hospital no es una organización industrial sino una especie de organización de servicio. Por tanto, se deben especificar las siguientes diferencias fundamentales del sistema sanitario (Vissers & Beech, 2005):

- El control de la producción en una fábrica se centra en el flujo de materiales, mientras que en la atención sanitaria éste es secundario poniendo el foco en el flujo de pacientes.
- En sanidad la interacción entre precio y rendimiento es mucho menor que la que existe en un entorno de producción.
- En general, los métodos de control de la producción cuentan con una serie de requisitos del producto final y entrega muy específicos y por el contrario en el ámbito sanitario, estos son a menudo subjetivos y difusos.
- En las organizaciones de atención a la salud normalmente no existe una única línea de mando, suele crearse un equilibrio de poder entre distintos grupos de interés, cada uno de los cuales tiene su propia idea sobre cuál debería ser el objetivo del centro.
- El personal clave del proceso son profesionales altamente capacitados (médicos especialistas), se encargan de generar las órdenes y también están involucrados en la entrega del servicio.
- La atención al paciente no es un producto que pueda ser almacenado, un hospital es una

organización orientada al servicio.

A modo de resumen las principales diferencias y similitudes se recogen en la tabla 2-1.

Teniendo en cuenta estas similitudes y diferencias podemos describir el control de la producción en las organizaciones de atención de la salud (de forma análoga al contexto industrial) como:

El diseño, la planificación, la implementación y el control de los mecanismos de coordinación entre el flujo de pacientes y las actividades diagnósticas y terapéuticas en los centros de salud para maximizar el rendimiento con los recursos disponibles, teniendo en cuenta los diferentes requisitos en la flexibilidad de entrega (electiva/cita, semi-urgente, urgente), las normas aceptables en cuanto a la fiabilidad de la entrega (listas de espera, tiempos de espera) y los resultados médicos aceptables (Vissers, 1994).

Tabla 2-1: Similitudes y diferencias entre el sistema industrial y sanitario (Vissers and Beech, 2005:27)

Características	Sistema industrial	Sistema sanitario
Objeto	Flujo de materiales	Flujo de pacientes
Requisitos del producto final	Especificado de antemano	Subjetivo y confuso
Medios de producción	Equipamiento y plantilla	Equipamiento y plantilla
Buffers	Stock o plazos de entrega	Listas de espera y plazos de atención
Objetivo final	Beneficio	Control del coste
Entorno de mercado	Mercado competitivo	Mercado competitivo limitado

2.4 Introducción a la programación de operaciones

En términos generales, la programación se ocupa de la asignación de recursos a tareas a lo largo del tiempo. En general, esa definición puede abarcar un gran número de aplicaciones en la vida real, como la asignación de cirujanos a los turnos de sus hospitales, la programación de aeronaves en aeropuertos, etc. (Framiñán, Leisten & Ruiz, 2014). En esta sección se va a comparar la programación de la producción con la programación en el sector hospitalario, dado que se requiere cierto nivel de abstracción para gestionar estos recursos, muchos de los modelos y métodos que se aplican en el primero pueden aplicarse fuera del ámbito de la fabricación.

La programación y control de operaciones surge dentro del ámbito de la Organización de la Producción (*Production Management*) como un proceso industrial consistente en la toma de un elevado número de decisiones a lo largo del tiempo para asegurar la entrega al cliente de los productos deseados con la máxima calidad, mínimo coste y en un menor plazo. Estas decisiones pueden ser de alto impacto y a largo plazo, como decidir si cierta planta fabricará o no un nuevo producto, o decisiones a corto plazo y de pequeño impacto, tales como qué producto es el próximo en ser fabricado en cierta máquina del taller. Las primeras se denominan decisiones estratégicas, en el sentido de que tienen un enorme impacto en el resultado final de las empresas y, además, los recursos físicos

involucrados hacen que sea prácticamente imposible su revisión en menos de un horizonte que se mide normalmente en años (Framiñán, Leisten & Ruiz, 2014).

Una vez diseñada la estructura de fabricación y distribución, hay un gran número de decisiones relacionadas con la forma de hacer el mejor uso de esta estructura. Estas decisiones se descomponen, a su vez, en decisiones a medio plazo (decisiones tácticas) y decisiones de corto plazo (decisiones operativas). Normalmente se establece un plan de producción agregado que servirá para estimar la capacidad (agregada) que se utilizará, la cantidad de materias primas que se comprará, etc., pero no como una indicación detallada de las órdenes de producción que deben ser entregadas a la planta de producción. Este exhaustivo plan suele dejarse para una etapa posterior en la que se conoce un detallado estado de la planta y de los pedidos en firme a procesar. El resultado de este proceso es, idealmente, un programa donde se especifica qué trabajo debe llevarse a cabo en cada recurso, y cuándo (Framiñán, Leisten & Ruiz, 2014).

De forma paralela se puede describir en qué consiste el proceso de decisión en los hospitales. Consta generalmente de dos niveles: el primero de ellos, llamado “*Off-line*” se resuelve tradicionalmente en dos pasos (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015): el primer paso (conocido como programación avanzada) implica la determinación del quirófano y el día en el que será ejecutada la operación, mientras que en el segundo paso (llamado programación de la asignación) se obtiene una secuencia específica de cirugías para cada quirófano. El segundo nivel, llamado “*On-line*”, trata de poner en marcha mecanismos de control para monitorizar el proceso y reaccionar ante posibles sucesos no anticipados, ya que en este ámbito pueden llegar a existir grandes discrepancias entre la duración teórica de la programación y la duración real de las intervenciones.

Algo en común que comparten los dos entornos que ocupan esta sección es que los planes raramente ocurren según lo estimado, esto puede ser una exageración en otros ámbitos, pero es especialmente destacable tanto en la sanidad como en la fabricación. Por tanto, el control de la producción es fundamental en ambos casos, poniendo en marcha mecanismos para monitorizar las desviaciones del programa y ejecutar acciones correctoras, o lo que se ha llamado en el caso del hospital: *nivel On-Line*.

En relación a lo anterior, se puede deducir que la programación consiste en un proceso adaptativo y dinámico de toma de decisiones iterativas relacionadas con la asignación de recursos a distintos “trabajos” (en este caso pacientes). Estas decisiones pueden variar enormemente entre empresas o distintos ámbitos, pero tienen en común una serie de características genéricas (Framiñán, Leisten & Ruiz, 2014):

- Son decisiones complejas, ya que implican el desarrollo de planes detallados para asignar tareas a los recursos a lo largo del tiempo.
- Las decisiones de programación son decisiones de intervalos de tiempo cortos que deben tomarse una y otra vez. La vida media de un programa es muy corta, y de hecho muchos autores se refieren a un proceso continuo de decisión de programación.
- A pesar de ser una decisión a corto plazo, la programación a largo plazo afecta al nivel de servicio al cliente o paciente y en los costes totales.

En ambos casos el proceso es llevado a cabo por humanos (programadores o responsables de la toma de decisiones) que pueden ser asistidos por una herramienta informática que les ayuda a modelar algunos problemas de decisión que surgen durante este proceso, y para encontrar una solución. Hay que hacer cuatro observaciones respecto a definición (Framiñán, Leisten & Ruiz, 2014):

1. La programación es un proceso.
2. En segundo lugar, el proceso de programación está dirigido por un humano, un equipo o una organización. La visión de los sistemas totalmente automatizados que realizan y ejecutan la programación de decisiones sin guía humana no refleja la realidad actual en la mayoría de los entornos. Por lo tanto, hay una necesidad de entender e incorporar el elemento humano en

este proceso de decisión.

3. En tercer lugar, la naturaleza relativamente estructurada del proceso de programación y su inherente complejidad lleva naturalmente a la conveniencia de formular el problema de la decisión en términos de modelos matemáticos para evaluar mejor las decisiones a tomar por los programadores, así como los métodos correspondientes para resolver estos modelos.
4. Cuarto, aunque estrictamente hablando no hay necesidad de una herramienta para apoyar a los programadores, en la mayoría de los casos es completamente impráctico llevarlas a cabo sin ninguna ayuda de una herramienta de software. Simplemente el mero volumen de datos referidos a los recursos y tareas o pacientes que se encuentran en la mayoría de los talleres u hospitales sería suficiente para justificar la necesidad de tal herramienta.

Tras esta sección se puede concluir que los métodos desarrollados para la gestión de operaciones en el sector industrial pueden ser aplicados de forma similar en el ámbito hospitalario, como se detallará más detenidamente en el capítulo 5, Metodología de resolución.

2.5 Diferencia entre planificación y programación

Alcanzado este punto, se quiere enfatizar la diferencia entre la planificación a medio o largo plazo que es objetivo de este proyecto y la programación de la producción enfocada al corto plazo y a la obtención de una secuenciación detallada, en la que se sabe con exactitud el inicio y fin del procesado de cada trabajo en cada máquina. Como se ha visto en la sección 2.4, el proceso de programación de operaciones es similar en el sistema sanitario e industrial, pasando progresivamente de una planificación más general en el tiempo a una programación específica, que posteriormente será controlada a través de diversos mecanismos. En este caso, los métodos expuestos en el capítulo 5, son desarrollados para la programación de la producción, y adaptados al problema de decisión hospitalario a medio plazo obteniendo buenas soluciones que se van ajustando a las demandas de la actividad diaria en niveles posteriores de decisión. Esta metodología ha sido aplicada en sus diversas formas al nivel de decisión *offline* como se verá en la sección 3.2, revisión de la literatura.

3 ANTECEDENTES

En este capítulo se pretende poner de relieve los métodos más comunes llevados a cabo en la gestión de operaciones en la sanidad y a la vez, hacer una revisión de la literatura existente, para aclarar qué avances se han realizado en el ámbito que ocupa este proyecto.

3.1 Métodos más comunes en la gestión de operaciones en sanidad

Debido al tamaño y la complejidad de los sistemas de salud es prácticamente inviable gestionar las operaciones mediante la intuición, fuga de ideas o análisis del benchmark (Litvak y otros, 2007).

Por este motivo se han ido desarrollando distintos métodos para gestionar la planificación en el ámbito de la salud, entre los más utilizados destacan los siguientes según el *Institute of Health Improvement* (Litvak y otros, 2007):

- *Método del camino crítico (shifting bottleneck)*. Este método tiene en cuenta tres aspectos: la relación lógica establecida entre las distintas tareas, el tiempo asignado a cada una y sus fechas de entrega. Consiste en dibujar un grafo y localizar la ruta crítica, es decir, el camino más largo que determina la duración mínima. A mayor incertidumbre del proyecto menor es la fiabilidad de este método.
- *Limitaciones del sistema*. Se trata de localizar los cuellos de botella del sistema. En los hospitales suelen ser por ejemplo la disponibilidad de los quirófanos o de las camas. Es fundamental identificar estos factores limitantes ya que de ellos depende el rendimiento de todo el sistema, aportando un camino esencial para posibles mejoras.
- *Metodología de la variabilidad*. Este método consiste en identificar y reducir al máximo la variabilidad artificial en el sistema sanitario, que en los hospitales se debe sobre todo a las admisiones programadas, especialmente en el caso de las cirugías.
- *Modelo de simulación*. Gracias a una simulación virtual dentro del sistema es posible explorar el efecto de los distintos diseños, permitiendo experimentar cómo reaccionarían ante posibles cambios o en diferentes escenarios. La utilización de estos modelos de simulación puede conducir a un sistema sanitario más seguro, puntual, efectivo y centrado en el paciente y al mismo tiempo reducir costes. La obtención de la información necesaria para ejecutar la simulación requiere de la colaboración directa con los proveedores de salud.

3.2 Planificación y programación del quirófano: revisión de la literatura

La programación avanzada de quirófanos en el nivel de decisión *Off-line* (explicado anteriormente en la sección 2.4, introducción a la programación de operaciones), entendida como la planificación de la cirugía en cuanto a fecha y quirófano, es un tema ampliamente investigado. Acorde a la literatura existente en esta área, los estudios pueden analizarse desde varias perspectivas (Cardoen, Demeulemeester & Belien, 2009). En este caso se explica de forma más detallada el tipo de análisis y la técnica de solución empleada, por ser los aspectos más relevantes para este proyecto:

- *Características de los pacientes*: según el estado electivo (pacientes con una posible planificación ya sean internos o externos) o no electivo (urgencia o emergencia) del paciente. El estudio sobre el primer grupo es mucho mayor en comparación con el no electivo, aunque en muchos casos ni si quiera se especifica.

- *Criterios de actuación:* se han considerado criterios de rendimiento variados para evaluar este tipo de problemas como son el tiempo de espera, aplazamiento de cirugías, utilización de los recursos, el *makespan*, el valor financiero, preferencias o la productividad entre otros.
- *Nivel de decisión:* en función del tipo de decisión que debe ser tomada (fecha, duración, quirófano o capacidad).
- *Tipo de análisis:* Una parte sustancial de la literatura sobre planificación y programación de quirófanos consiste en contribuciones en las que se expone un problema y se optimiza consecutivamente. Estos enfoques de optimización son o bien exactos, es decir, que conducen finalmente a una solución para la que se puede probar la optimización, o bien de naturaleza heurística. Además, se puede distinguir entre los enfoques de objetivo único y múltiple basados en el número de criterios de rendimiento que deben ser optimizados. Aunque a menudo se afirma que los procesos heurísticos son indispensables para resolver problemas prácticos o de tamaño real eficientemente, existen muchos enfoques exactos potentes, incluso cuando se consideran criterios múltiples. También es necesario tener en cuenta la naturaleza de las restricciones, en la tabla 3-1 se muestran los tipos más comunes de restricciones extraídos de la literatura sobre la optimización de quirófanos.
- *Técnica de solución:* La literatura sobre planificación y programación de quirófanos exhibe una amplia gama de metodologías que se recuperan de los ámbitos de la gestión de operaciones como se vio en el capítulo 2 (Comparación de la gestión de operaciones en el sistema industrial y el sanitario). Los métodos más utilizados son los siguientes (Cardoen y otros, 2009:20):
 - Programación matemática:
 - Programación lineal
 - Programación cuadrática
 - Programación de objetivos
 - Programación mixta de enteros
 - Dinámica
 - Generación de columnas
 - Branch and Price
 - Otros
 - Heurísticas constructivas
 - Metaheurísticas:
 - Algoritmo genético
 - Búsqueda Tabú
 - Otros
 - Branch and Bound
 - Simulación:
 - Monte-Carlo
 - Por eventos discretos
- *Incertidumbre:* dependiendo si los investigadores incorporan la incertidumbre de las llegadas o las duraciones (enfoque estocástico) o no (enfoque determinista).
- *Aplicabilidad del estudio:* información aportada sobre el testeo de la investigación y su

implementación en un entorno real.

Tabla 3-1: Restricciones consideradas en la literatura (Cardoen y otros, 2009:17)

Recurso limitante	Otros
Zonas de espera	Restricciones de precedencia (por ejemplo por enfermedades infecciosas) o retrasos necesarios (limpieza)
Salas	
UCI (Unidad de Cuidados Intensivos)	
PACU (Unidad de Cuidados Post-anestesia)	Fechas de entrega y llegada
Equipamiento	
Personal quirúrgico	
Presupuesto	Restricciones de demanda
Capacidad regular de quirófanos	

4 DESCRIPCIÓN DEL PROBLEMA

Una vez conocida la naturaleza del problema de decisión en cuestión y las distintas formas en las que ha sido abordado en la literatura, en este capítulo, por primera vez, se exponen las restricciones, límites de capacidad, tipos de pacientes que pueden entrar en el sistema y el objetivo establecido, que caracterizan a este estudio.

4.1 Limitaciones de los recursos

El hospital objeto de análisis consta de una serie de consultas y quirófanos multifuncionales disponibles cada día durante 8 horas. Se ha de indicar que la capacidad de una consulta al día coincide con el tiempo disponible de un cirujano, ya que la presencia de éste en consulta es la que limita su utilidad.

Además, como suele ocurrir en entornos reales (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015), se debe considerar que un porcentaje del total de las operaciones a programar pueden ser llevadas a cabo exclusivamente en un quirófano especializado durante un día determinado a la semana, por ser el mejor equipado para ese tipo específico de cirugía.

Cada día, cada cirujano tiene un tiempo máximo disponible para trabajar en una consulta o en la zona de quirófanos, siendo 0 (si ese día no está disponible) u 8 horas (si trabaja ese día la jornada completa). El número de quirófanos donde puede encontrarse un cirujano al día (Us) se limita con la intención de reducir su tiempo ocioso y el solapamiento de cirugías consecutivas realizadas por el mismo (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015).

Finalmente, el resto de los recursos perioperativos, tanto humanos como instrumentales, se consideran disponibles en cualquier momento, no representando un cuello de botella para este problema.

4.2 Descripción del sistema y tipos de pacientes

El proceso consta de dos etapas consecutivas, la etapa 1 hace referencia a las consultas y la etapa 2 a los quirófanos. Dentro del conjunto de pacientes hay que distinguir dos variantes, por un lado, aquellos que ya han pasado por consulta en una etapa previa y esperan directamente a ser operados y, por otro lado, aquellos que sí deben pasar por consulta antes de ser establecidas sus fechas de cirugía. En el primer caso el cirujano ya ha sido asignado, mientras que en el segundo caso, el mismo cirujano que les atienda en consulta será el encargado de su posterior intervención. De esta forma se dispone de dos listas de espera, una para cada etapa. El tiempo de duración de las consultas y las operaciones es estimado por el personal encargado en base al historial de datos y a las características del paciente (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015).

Hay que tener en cuenta que antes de poder fijar la fecha de cirugía, debe transcurrir un periodo de tiempo mínimo que es necesario para la preparación del paciente (*release date*) desde que es atendido en consulta.

4.3 Función objetivo

Por último, la función objetivo se ha establecido basada en la estrategia real a seguir por un jefe de servicio. Dada una fecha límite de cirugía para cada paciente conocida como fecha de entrega o *due date*, el objetivo es minimizar el retraso en el tiempo de acceso de dichos pacientes, teniendo en cuenta su prioridad clínica (a mayor urgencia mayor peso), maximizando a su vez la utilización de los recursos siempre que estén disponibles.

5 METODOLOGÍA DE RESOLUCIÓN

En este capítulo se expone el procedimiento general a seguir para resolver problemas de programación. El primer paso es el modelado del mismo, para, a continuación, encontrar métodos de resolución para los modelos obtenidos - y no para el problema real - y, en tercer lugar, se trata de transferir estas soluciones al mundo real (Framiñán, Leisten & Ruiz, 2014). Este proceso puede observarse esquemáticamente en la figura 5-1.

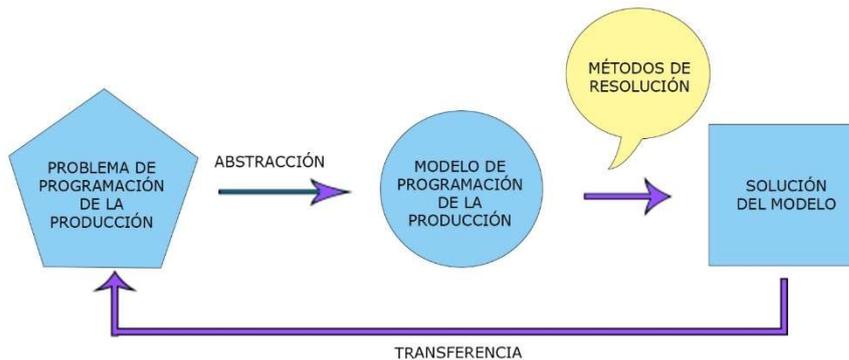


Figura 5-1: Procedimiento de resolución de problemas de programación (Apuntes de programación de operaciones)

5.1 Modelado

En general un modelo es una representación o visión abstracta de una realidad compleja, detallada y amplia. La forma en la que éste se construye está estrechamente ligada al uso que se le pretende dar.

Sobre los modelos hay que señalar que están llenos de limitaciones. La interpretación de los resultados de un modelo sólo es válida para el modelo, y por tanto la extrapolación a la realidad tiene que hacerse de manera sensata y científica. El modelo tiene que ser validado y los resultados se deben examinar detenidamente.

Existen diversos tipos de modelo y métodos de modelado. Los principales modelos empleados en la programación determinista son los modelos matemáticos y algorítmicos.

En particular, un modelo de programación es la abstracción formal de un problema de decisión de programación bien definido, en este caso, el problema de decisión descrito en el capítulo 4 a través de las limitaciones de recursos, restricciones de procesamiento y los distintos criterios. Éste puede concretarse aún más seleccionando los valores específicos (es decir, numéricos) de los recursos, tareas y demás elementos que lo componen, en este caso serían: pacientes, cirujanos, tiempos estimados, etc. Esta selección específica se denomina una instancia del modelo de programación.

5.1.1 Notación de los modelos

Algunos autores proponen la siguiente notación para los modelos basada en tres elementos (Framiñán, Leisten & Ruiz, 2014): $\alpha/\beta/\gamma$, siendo igual a:

- α : tipo de entorno del proceso. Se indica con la característica y el número de máquinas.

- β : restricciones del proceso.
- γ : función objetivo.

En las siguientes subsecciones se van a exponer los diferentes elementos que constituyen un modelo de programación de la producción identificando su correspondencia con los elementos que conforman el problema descrito en el capítulo 4, enmarcado en un entorno hospitalario.

5.1.2 Datos básicos

- *Trabajos*: se asume que un modelo está determinado por un número de trabajos conocido, finito y determinista, en el ámbito de este proyecto se habla de un conjunto de pacientes.
- *Máquinas*: son los recursos productivos por los que deben pasar los trabajos. En este estudio estos recursos hacen referencia a las consultas y los quirófanos.
- *Ruta de proceso*: cada trabajo tiene una ruta predefinida a través de la planta de producción al igual que en el hospital se exige que algunos pacientes deben pasar primero por la etapa 1 y a continuación por la 2, y otros solo por la 2 sin pasar previamente por ningún proceso.
- *Tiempo de proceso*: es el lapso de tiempo conocido que se necesita en una determinada máquina para procesar un trabajo en concreto. En este estudio corresponde a la duración estimada del paso de un paciente por consulta y quirófano.

5.1.3 Otros datos

- *Fecha de llegada (release date)*: es el instante de disponibilidad a partir del cual el trabajo puede empezar a ser procesado. En este caso la fecha de llegada hace referencia al tiempo mínimo necesario que debe transcurrir antes de la cirugía de un paciente desde que fue visto en consulta debido a pruebas y preparación previa.
- *Fecha de entrega (due date)*: instante de tiempo en que un trabajo debe estar terminado, o lo que es lo mismo, en la que un paciente debe ser operado. En el problema que ocupa este trabajo, estas fechas pueden violarse pero sujetas a una penalización.
- *Peso*: prioridad del trabajo o de la cirugía. Se establece en función de la urgencia del paciente como se verá en más detalle en la subsección 7.3.2, datos relevantes a las cirugías.

5.1.4 Características de las máquinas (α)

En cuanto al tipo de entorno (*layout*) se clasifican de acuerdo a la disposición de las diferentes máquinas y también según las diferentes rutas de procesamiento. Las principales disposiciones son las siguientes:

- *Una máquina (Single Machine)*
- *Máquinas en paralelo (Parallel machines)*: idénticas, uniformes o no relacionadas.
- *Taller de flujo regular (Flowshop)*: la ruta siempre es máquina 1, máquina 2, máquina 3...
- *Taller de trabajos (Job shop)*: cada trabajo tiene una ruta diferente.
- *Taller abierto (Open shop)*: no hay ruta predeterminada.
- *Híbridos (Hybrid)*: se constituye por etapas.

En el hospital en particular se dispone de un entorno híbrido de dos etapas en taller de flujo regular, es decir la etapa 2 siempre es posterior a la etapa 1, y ambas etapas son a su vez máquinas paralelas idénticas, o lo que es lo mismo, cada recurso tanto quirófanos como consultas son idénticos, cada paciente pasa independientemente por uno de ellos y su tiempo es independiente al recurso.

5.1.5 Restricciones (β)

Uno de los elementos fundamentales en la descripción de los modelos de programación son las restricciones. Como se recogió en la tabla 3-1, restricciones consideradas en la literatura, son de distintas naturalezas y son similares a las empleadas en el sector de la producción. Algunos ejemplos frecuentes son (Framiñán, Leisten & Ruiz, 2014):

- Interrupción de los trabajos
- Fechas de entrega y llegada
- Tiempos de setup
- Lotes
- Tiempos de proceso especiales
- Precedencia
- Elegibilidad de la máquina

De las anteriores pueden identificarse en el problema en cuestión por ejemplo la restricción relacionada con la fecha de llegada y la elegibilidad. En problemas de máquinas paralelas, o en entornos híbridos en los que las etapas tienen máquinas paralelas, es normal no esperar que todas las máquinas sean capaces de procesar todos los trabajos. Las fábricas evolucionan con el tiempo y la vieja maquinaria es sustituida por nuevos equipos con mejores características. Esto no solo afecta a la variación en el tiempo de los procesos, en algunos casos algunas máquinas solo son capaces de procesar una serie de trabajos y/o algunos trabajos solo pueden ser asignados a un conjunto de máquinas (Framiñán, Leisten & Ruiz, 2014). Esto se denomina elegibilidad de la máquina y se da de la misma manera en los quirófanos. Como se comentó en el capítulo 4, en la realidad existen ciertas cirugías que por su complejidad o características particulares solo pueden ser llevadas a cabo en un quirófano específico por ser el más apropiado o mejor equipado para su ejecución.

5.1.6 Función objetivo (γ)

En gestión de la producción, y por consiguiente en programación de la producción, el objetivo requerido suele clasificarse en varias categorías: costes (beneficio), tiempo, calidad y flexibilidad (Framiñán, Leisten & Ruiz, 2014).

Los objetivos empleados en la construcción de modelos de programación de la producción están relacionados con la optimización de algún indicador, denominados medidas de rendimiento. Algunos de los más importantes en base al tiempo son los siguientes (Framiñán, Leisten & Ruiz, 2014):

- C_j (*Completion time*): Tiempo de terminación del trabajo j . Instante en el que el trabajo acaba.
- F_j (*Flow Time*): Tiempo de flujo del trabajo j . Tiempo que está en el entorno.
- L_j (*Lateness*): Retraso del trabajo j .
- T_j (*Tardiness*): Tardanza del trabajo j . Si el trabajo termina antes de la fecha de entrega no pasa nada.
- E_j (*Earliness*): Adelanto del trabajo j . Si el trabajo termina después de la fecha de entrega no pasa nada.
- U_j (*Tardy job*): Trabajo tarde.

A partir de estos indicadores se pueden definir los objetivos más comunes, que de forma resumida se clasifican en dos tipos según se trate de minimizar el máximo indicador de todos los trabajos o la suma total.

Por último, indicar que dados los diferentes pesos para los trabajos, podemos modificar cualquier objetivo con la intención de expresar la prioridad, el costo u otro índice de importancia relativa que se pueda necesitar, como sería en este problema la urgencia clínica.

Se puede apreciar que, por las similitudes ya comentadas entre ambos sistemas, los indicadores anteriores son fácilmente extrapolables a la planificación de quirófanos, de tal manera que, según lo establecido en el capítulo 4, se pretende minimizar la suma del *tardiness* o tardanza de todos los pacientes en su asignación a la segunda etapa del proceso, indicando además su prioridad clínica a través de unos pesos dados.

La diferencia fundamental es que en la programación de la producción todos los trabajos son procesados dentro del horizonte de planificación mientras que en el hospital los pacientes pueden ser operados o no en ese periodo de tiempo. Esto junto con el hecho de que existen pacientes con una fecha de entrega mayor al límite del horizonte temporal, conduce a la necesidad de añadir un término a la función objetivo que busque maximizar la utilización de los recursos siempre que estén disponibles, tal y como se verá en el capítulo 7.

5.2 Presentación de los posibles métodos de programación

Un método de programación es un procedimiento formal que puede aplicarse a cualquier instancia de un modelo de programación a fin de obtener una solución factible que (presumiblemente) obtenga buenos resultados con respecto a un objetivo buscado (Framiñán, Leisten & Ruiz, 2014).

El procedimiento puede describirse mediante una serie de pasos finitos, por lo que puede ser codificado y finalmente ejecutado por un ordenador. El nombre técnico de ese procedimiento es *algoritmo*.

En esta sección se van a presentar brevemente los métodos existentes para proporcionar soluciones (buenas o incluso óptimas) para un modelo de programación dado (Framiñán, Leisten & Ruiz, 2014).

5.2.1 Métodos exactos

Los algoritmos exactos garantizan que ningún otro programa funcione mejor que el obtenido con respecto a los objetivos buscados. La solución obtenida se denomina solución óptima u óptimo.

A grandes rasgos, hay dos tipos de algoritmos exactos para los problemas de programación:

1. Algoritmos constructivos exactos (polinomiales): son algoritmos que explotan algunas propiedades del modelo específico para construir una solución que está garantizada como óptima. En principio estos procedimientos constructivos solo abarcan un espectro muy limitado de problemas bastante sencillos, por lo que no se han considerado una opción para el problema a resolver, aunque si es cierto que pueden ser utilizados como sub-módulos en la aproximación de problemas de programación más complejos.
2. Algoritmos enumerativos (no polinomiales): Son algoritmos que implícita o explícitamente garantizan la evaluación de todas las posibles soluciones del modelo obteniendo así el óptimo. Incluye métodos como la enumeración completa, el *branch and bound* o la programación matemática. Al ser algoritmos no polinomiales dan soluciones óptimas solo de instancias pequeñas. En otras palabras, los enfoques de enumeración ofrecen la comodidad y seguridad de soluciones óptimas exactas a expensas del esfuerzo computacional que, en muchos casos, crece exponencialmente con el tamaño del problema, haciendo que los problemas de gran tamaño sean casi intratables para su solución.

5.2.2 Métodos aproximados

Los algoritmos aproximados no garantizan que la solución sea óptima, se puede decir que aportan soluciones adecuadas o de buena calidad.

Se pueden clasificar como heurísticas, que son métodos aproximados específicamente adaptados a un problema de decisión particular, y procedimientos más genéricos conocidos como metaheurísticas. Además, se puede diferenciar entre heurísticas constructivas y heurísticas de mejora. La diferencia

entre estas dos últimas es que las heurísticas constructivas generan una solución esperanzadoramente factible y buena desde cero, es decir, basándose exclusivamente en los datos de entrada, sin referirse a otra solución del problema en cuestión. Por el contrario, las heurísticas de mejora tratan de encontrar una solución mejor respecto a otras soluciones encontradas hasta el momento. Dentro de los procedimientos de optimización (heurísticas) más sofisticados, la heurística constructiva se utiliza a menudo como punto de partida para los enfoques de mejora.

Finalmente, dado que muchos problemas de optimización tienden a atascarse fácilmente en el subóptimo local (como consecuencia de la no convexidad) al aplicar heurísticas constructivas en combinación con enfoques deterministas de búsqueda de vecindades de "mejora permanente", durante los últimos años se han desarrollado varios enfoques que tratan de superar o al menos reducir este problema. Estos enfoques son las metaheurísticas.

5.2.3 Método elegido: Programación Lineal Entera (ILP)

Los enfoques exactos proporcionan la solución óptima, pero sólo son viables para problemas sencillos e instancias pequeñas, por lo tanto no es aconsejable utilizarlos en entornos complejos en los que se van a programar cientos o incluso miles de trabajos. Sin embargo, sí es muy útil tenerlos en cuenta como métodos de partida y pueden servir posteriormente para inspirar la construcción de métodos aproximados (Framiñán, Leisten & Ruiz, 2014). Para estudiar el problema descrito en el capítulo anterior, se ha elegido como método de resolución la programación lineal entera con la intención de desarrollar una base sobre la que después construir algoritmos aproximados que den buenos resultados para instancias de tamaños reales.

Este método consistiría básicamente en dos pasos:

1. Formular el modelo de decisión en términos de programación lineal entera (ILP). Cabe destacar que una formulación inteligente del modelo podría reducir significativamente el esfuerzo computacional.
2. Aplicar un software de programación entera (IP) al modelo ILP generado en el paso previo.

5.3 Implementación y optimización del modelo

Para llevar a cabo este paso se ha elegido Microsoft Visual Studio como entorno integrado de desarrollo (IDE). Además del editor estándar y el depurador que ofrecen la mayor parte de IDE disponibles en el mercado, Visual Studio incluye compiladores, herramientas de finalización de código, diseñadores gráficos y otras muchas características para facilitar el proceso de desarrollo de software (<https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2019>). El siguiente paso es decidir un lenguaje de programación adecuado, así como un software de optimización como se detalla en las dos siguientes subsecciones.

5.3.1 Lenguaje de programación

Para comenzar a implementar el modelo y automatizar el procedimiento se requiere de un lenguaje de programación, algunos de los más comunes son, por ejemplo: C++, Java, C#, Python, etc. En este caso en particular se ha optado por C#. A continuación, se introducen algunas características y ventajas que ofrece este lenguaje (<https://docs.microsoft.com/es-es/dotnet/csharp/>).

C# es un lenguaje de programación moderno, basado en objetos y con seguridad de tipo y tiene sus raíces en la familia de lenguajes C.

Tiene un *sistema de tipo unificado*, es decir, todos los tipos de C#, incluidos los tipos primitivos como *int* y *double*, se heredan de un único tipo *object* raíz. Por lo tanto, todos los tipos comparten un conjunto de operaciones comunes, y los valores de todos los tipos se pueden almacenar, transportar y

utilizar de manera coherente. Además, C# admite tipos de valor y tipos de referencia definidos por el usuario, lo que permite la asignación dinámica de objetos, así como almacenamiento en línea de estructuras ligeras.

Las siguientes características de C# son muy útiles para la construcción de aplicaciones sólidas y duraderas:

- *La recolección de elementos no utilizados*: reclama de forma inmediata la memoria ocupada por objetos no usados.
- *El control de excepciones*: proporciona un enfoque estructurado y extensible para la detección y recuperación de errores.
- *El diseño con seguridad de tipos*: hace que sea imposible leer desde variables no inicializadas, inicializar matrices más allá de sus límites o realizar conversiones de tipo no comprobadas.

Además, uno de los puntos fuertes de este lenguaje es, que para asegurarse de que los programas y las bibliotecas de C# puedan evolucionar a lo largo del tiempo de manera compatible, se ha puesto mucho énfasis en el *versionamiento* de su diseño. Éste es un aspecto al que pocos lenguajes de programación prestan atención y, en consecuencia, los programas escritos en dichos lenguajes se interrumpen con más frecuencia de lo necesario cuando se introducen nuevas versiones de las bibliotecas dependientes.

5.3.2 Software de optimización

Posteriormente el modelo se carga a un software de optimización. Éstos se clasifican en general en dos categorías basadas en la concesión de licencias: software comercial y de código abierto (Anand, R., Aggarwal, D. & Kumar, V., 2017).

Algunos ejemplos de programas informáticos que trabajan con modelos ILP son: el solver de Microsoft Excel, LINGO de Lindo Systems, CPLEX de IBM y muchos otros, incluido el solver comercial Gurobi que ha sido empleado en este proyecto, en concreto la versión 9.0.1.

La idoneidad de un solver de optimización depende de la naturaleza del problema y del tiempo de cálculo. En base a ciertos análisis, se conoce que CPLEX y Gurobi proporcionan los resultados más competitivos en problemas de entornos reales abierto (Anand, R., Aggarwal, D. & Kumar, V., 2017).

En particular, *Gurobi Optimizer* es un solver moderno y potente para problemas matemáticos de optimización de distintas naturalezas (LP, MILP, QP, MIQP, QCP, MIQCP). Entre sus características fundamentales destaca que está escrito en C y que es accesible desde varios lenguajes de programación, como es en este caso C# (<https://www.gurobi.com/products/gurobi-optimizer/>).

5.4 Evaluación del modelo: Estudio de simulación

El método más extendido para la evaluación del rendimiento de algoritmos son los estudios de simulación en los que se utilizan un conjunto de instancias de muestra para la evaluación del método correspondiente. En general pueden resumirse en dos pasos (Framiñán, Leisten & Ruiz, 2014):

1. *La generación del banco de pruebas (testbed)*: consiste en generar un conjunto de instancias que, o bien se generan específicamente para el modelo en cuestión, o bien se adaptan de otros modelos existentes. Esta adaptación puede tener lugar en dos sentidos, es decir, la adaptación puede referirse únicamente al método de generación de instancias de prueba, mientras que las instancias propiamente dichas se generan de nuevo, o en cambio son las instancias las que se extraen directamente de las referencias o de algunas bases de datos públicas.

2. *Análisis de los algoritmos*: este paso se refiere a la comparación de los algoritmos a tener en cuenta, de forma que se ejecutan con cada instancia del problema y se evalúa su rendimiento relativo empleando unas ciertas medidas para finalmente deducir cuál funciona mejor. Se debe indicar que este paso queda fuera del ámbito de este proyecto, ya que solo se va a analizar el comportamiento modelo ILP, sirviendo como base para posteriores análisis de algoritmos aproximados como se detallará más detenidamente en la sección 8.1, líneas de futuro.

En lo que respecta a la generación del banco de pruebas, se debe tener en cuenta lo siguiente (Framiñán, Leisten & Ruiz, 2014):

- El tamaño de las instancias (por ejemplo indicado en este estudio por el número de pacientes y cirujanos) debería ser acorde a las dimensiones del problema real y/o a las dimensiones de las instancias utilizadas en estudios comparables. Esto incluye también las diferentes combinaciones entre número de pacientes y número de cirujanos, número de quirófanos, etc.
- El número de instancias generadas debería ser suficiente para obtener resultados útiles, en particular en este estudio, se pretende determinar el alcance de la programación del modelo ILP.
- Los principales parámetros del modelo (como tiempos de consulta y quirófanos, *release* y *due dates*, etc.) deben elegirse adecuadamente con respecto al objetivo previsto. En este proyecto los valores se basan en las observaciones del caso del mundo real específico.
- Cuando se amplía/adapta un banco de pruebas existente a otro modelo, como es el caso que ocupa este trabajo, las características adicionales del modelo en cuestión, por ejemplo el número de consultas, tienen que ser cuidadosamente determinadas e incorporadas en las instancias de prueba.

6 COMPLEJIDAD

Para comprender el alcance del método de resolución propuesto en este proyecto es necesario conocer su complejidad y por ende, es importante definir las bases sobre las que esta complejidad se sustenta.

La complejidad en problemas de programación puede ser definida desde dos puntos de vista. Puede restringirse, por un lado, al ámbito teórico, o también puede abordarse de manera más general y luego extenderse al ámbito del mundo real (Framiñán, Leisten & Ruiz, 2014).

6.1 Complejidad computacional

En el primer caso se denomina complejidad computacional y se refiere a la complejidad del problema formal y de los correspondientes algoritmos para resolver dicho problema.

De manera informal se pueden definir dos tipos de complejidad computacional (Framiñán, Leisten & Ruiz, 2014):

- *Complejidad de los algoritmos*: se refiere al comportamiento de un algoritmo específico para resolver el modelo dado, asignándole una estimación de coste (en tiempo). Esta estimación se conoce como la complejidad temporal o tiempo de ejecución. Este tiempo puede depender de varios factores, tales como la instancia específica del modelo para el que se aplica el algoritmo (tamaño de los datos de entrada) o la capacidad de la máquina.
- *Complejidad de los modelos*: proporciona la complejidad del “mejor algoritmo” capaz de encontrar la solución óptima.

En términos generales, este concepto de complejidad sirve para distinguir entre dos tipos de algoritmos:

- *Algoritmos polinomiales*: cuyo esfuerzo de cálculo puede estar limitado por alguna función polinómica, es decir, normalmente pueden lograrse considerando un esfuerzo de cálculo razonable.
- *Algoritmos no polinomiales*: en los que aún no se ha establecido tal límite polinómico y donde, probablemente, nunca se encontrará.

A su vez, esta distinción conduce a una clasificación de los modelos según su complejidad:

- *Modelos con algoritmos polinomiales (P)*: “fáciles” de resolver de forma exacta.
- *Modelos con algoritmos no polinomiales (NP)*: “difíciles” o “muy difíciles” de resolver de forma exacta.

En particular, los algoritmos que realizan una enumeración explícita o implícita de cada solución factible son de tipo no polinómico (NP). Por tanto, según el método exacto elegido en la sección 5.2, que consiste en la programación matemática del modelo ILP, se deduce la naturaleza no polinómica de este modelo en cuestión.

6.2 Complejidad de los entornos reales

Es lógico asumir que los problemas del mundo real son complejos y sus respectivos problemas teóricos también lo son, sin importar cómo se defina la complejidad. Los problemas de decisión en el mundo real suelen implicar la dificultad impuesta por un marco de decisión intrincado. Esto ocurre naturalmente tanto en el ámbito de la programación industrial como en un entorno hospitalario, donde existen multitud sutilezas a tener en cuenta como pueden ser (Framiñán, Leisten & Ruiz, 2014):

- Objetivos diferentes, en ocasiones sin ser claramente definidos.
- Una gran variedad de limitaciones y restricciones, a veces ni siquiera claramente preestablecidas.
- Un gran número, tal vez incluso especificadas de manera difusa, de posibles acciones.
- Un sistema jerárquico de planificación y toma de decisiones posiblemente poco definido.
- La complejidad añadida por la dinámica del día a día y la incertidumbre.
- La interacción de todos los aspectos mencionados anteriormente, dentro de su categoría y también, entre las categorías.

Como ya se ha indicado anteriormente, no existe una definición concreta de la complejidad. Sin embargo, existe una clasificación de la complejidad (Reiss, 1993) que puede servir al menos como un primer enfoque para aclarar las cuestiones de complejidad de un problema de decisión. Si bien en la literatura (Framiñán, Leisten & Ruiz, 2014) este enfoque ha sido particularizado para la programación de la producción, resulta de interés en este estudio extraer conclusiones similares para el ámbito sanitario que ocupa este proyecto como se ha hecho en capítulos anteriores.

Según Reiss (1993), los aspectos que conforman la complejidad pueden clasificarse de dos formas:

1. *Aspectos de masa*, divididos a su vez en:

- La multiplicidad, es decir, el número de elementos e interacciones en el sistema, y
- La variedad, o lo que es lo mismo, el número de elementos e interacciones *diferentes* en el sistema.

2. *Los aspectos del caos*, divididos a su vez en:

- La ambigüedad, es decir, el grado de incertidumbre de las características de los elementos e interacciones en el sistema, y
- La variabilidad, o lo que es lo mismo, el cambio de las características de los elementos y las interacciones a lo largo del tiempo.

Aplicando esta estructura de clasificación de la complejidad a la planificación de consultas y quirófanos en un hospital, se determina la multiplicidad del sistema, por ejemplo, por el número de recursos, el número de etapas, etc. La variedad puede referirse, por ejemplo, a la (no) homogeneidad de los tiempos de consultas y cirugías, a la diversidad de las rutas que podrían seguirse en este entorno, etc.

Los aspectos caóticos de la programación hacen referencia a la ambigüedad/incertidumbre (programación determinista vs. estocástica) y/o la variabilidad (programación estática vs. dinámica) de los entornos. Este aspecto ha sido estudiado en la literatura tal y como se vio en la revisión de la literatura en la sección 3.2. Ciertos estudios ponen de relieve la solidez de los planes de quirófano obtenidos mediante un enfoque estocástico, en comparación con los generados mediante un enfoque determinista, y la importancia de tener en cuenta tanto las camas de la UCI como las de la sala de operaciones en la planificación de los quirófanos (Jebali & Diabat, 2015). En este proyecto se busca un primer acercamiento a la planificación conjunta con una etapa previa de consultas, por lo que el enfoque estocástico queda fuera de su alcance.

7 APLICACIÓN DE LA METODOLOGÍA

Comprendida la complejidad del problema y descritos los métodos elegidos para su resolución, en este capítulo se expone el paso a paso seguido durante la aplicación de la metodología. También se realiza el análisis conveniente de las soluciones obtenidas y se extraen los valores de las variables para una instancia específica.

7.1 Modelo lineal Entero

A continuación se muestra el modelo lineal entero desarrollado para representar el problema de programación descrito en el capítulo 4. En la tabla 7-1 se recogen los conjuntos, parámetros y variables empleados.

Tabla 7-1: Conjuntos, parámetros y variables usados en el modelo lineal entero de decisión

Índices y conjuntos	
$i \in I$	Conjunto de pacientes
$s \in S$	Conjunto de cirujanos
$c \in C$	Conjunto de consultas
$j \in J$	Conjunto de quirófanos
$h \in H$	Conjunto de periodos de tiempo dentro del horizonte de planificación
Parámetros	
t_{1i}, t_{2i}	Tiempo estimado del paciente i en la etapa 1 y 2 respectivamente (en minutos)
a_{sh}	Capacidad del cirujano s el día h (en minutos)
m_{jh}	Capacidad del quirófano j el día h (en minutos)
r_i	Tiempo mínimo entre consulta y quirófano para el paciente i
d_i	Fecha límite de la cirugía i
w_i	Prioridad clínica de la cirugía i
δ_{ijh}	Parámetro binario que vale 1 si el paciente i puede ser operado en el quirófano j el día h ; 0 en otro caso
p_i	Parámetro binario que vale 1 si el paciente i debe ser asignado a consulta; 0 e.o.c
γ_i	Cirujano encargado del paciente i en lista de espera para quirófano
U_s	Número entero no negativo de quirófanos en los que cada cirujano puede realizar cirugías en el mismo día.
Variables	
X_{ich}	1 si el paciente i es visto en la consulta c el día h , 0 e.o.c
X'_{ijh}	1 si el paciente i es operado en el quirófano j el día h , 0 e.o.c
Z_{sch}	1 si el cirujano s se encuentra en la consulta c el día h , 0 e.o.c
Z'_{sjh}	1 si el cirujano s se encuentra en el quirófano j el día h , 0 e.o.c
Y_{isch}	1 si el paciente i es atendido por el cirujano s en consulta c el día h , 0 e.o.c
Y'_{isjh}	1 si el paciente i es operado por el cirujano s en el quirófano j el día h , 0 e.o.c
M_{sh}	1 si el cirujano s se encuentra en algún quirófano el día h , 0 e.o.c
T_i	Tardanza o <i>tardiness</i> en la fecha de operación del paciente i

$$\text{Minimizar } \sum_{i \in I} w_i * T_i + \frac{I - \sum_{i \in I} \sum_{j \in J} \sum_{h \in H} X'_{ijh}}{I}$$

Sujeto a:

Restricciones Objetivo

$$T_i \geq \sum_{j \in J} \sum_{h \in H} X'_{ijh} * h - d_i \quad (\forall i \in I) \quad (\text{R.1})$$

$$T_i \geq (H - d_i) * \left(1 - \sum_{j \in J} \sum_{h \in H} X'_{ijh} \right) \quad (\forall i \in I) \quad (\text{R.2})$$

Etapa 1 (Consultas)

$$\sum_{c \in C} \sum_{h \in H} X_{ich} \leq p_i \quad (\forall i \in I) \quad (1)$$

$$\sum_{s \in S} Z_{sch} = 1 \quad (\forall c \in C, \forall h \in H) \quad (2)$$

$$\sum_{c \in C} Z_{sch} + M_{sh} \leq 1 \quad (\forall s \in S, \forall h \in H) \quad (3.1)$$

$$\sum_{j \in J} Z'_{sjh} \leq U_s * M_{sh} \quad (\forall s \in S, \forall h \in H) \quad (3.2)$$

$$X_{ich} + Z_{sch} \leq Y_{isch} + 1 \quad (\forall i \in I, \forall c \in C, \forall s \in S, \forall h \in H) \quad (4.1)$$

$$X_{ich} + Z_{sch} \geq 2 * Y_{isch} \quad (\forall i \in I, \forall c \in C, \forall s \in S, \forall h \in H) \quad (4.2)$$

$$\sum_{s \in S} \sum_{c \in C} \sum_{h \in H} Y_{isch} = 0 \quad (\forall i \in I | p_i = 0) \quad (4.3)$$

$$\sum_{i \in I} \sum_{c \in C} t_{1i} * Y_{isch} \leq a_{sh} \quad (\forall s \in S, \forall h \in H) \quad (5)$$

Etapa 2 (Quirófanos)

$$\sum_{j \in J} \sum_{h \in H} X'_{ijh} * h + \left(1 - \sum_{j \in J} \sum_{h \in H} X'_{ijh} \right) * M \geq \sum_{c \in C} \sum_{h \in H} X_{ich} * h + r_i \quad (\forall i \in I) \quad (6.1)$$

$$\sum_{c \in C} \sum_{h \in H} X_{ich} \geq \sum_{j \in J} \sum_{h \in H} X'_{ijh} \quad (\forall i \in I | p_i = 1) \quad (6.2)$$

$$X'_{ijh} = 0 \quad (\forall i \in I, \forall j \in J, \forall h \in H | \delta_{ijh} = 0) \quad (7)$$

$$X'_{ijh} \leq \sum_{s \in S} Y'_{isjh} \quad (\forall i \in I, \forall j \in J, \forall h \in H) \quad (8.1)$$

$$\sum_{s \in S} Y'_{isjh} \leq 1 \quad (\forall i \in I, \forall j \in J, \forall h \in H) \quad (8.2)$$

$$X'_{ijh} + Z'_{sjh} \geq 2 * Y'_{isjh} \quad (\forall i \in I, \forall j \in J, \forall s \in S, \forall h \in H) \quad (8.3)$$

$$\sum_{i \in I} \sum_{j \in J} t_{1i} * Y'_{isjh} \leq a_{sh} \quad (\forall s \in S, \forall h \in H) \quad (9.1)$$

$$\sum_{i \in I} t_{2i} * X'_{ijh} \leq m_{jh} \quad (\forall j \in J, \forall h \in H) \quad (9.2)$$

$$\sum_{h \in H} \sum_{c \in C} Y_{isch} \geq \sum_{h \in H} \sum_{j \in J} Y'_{isjh} \quad (\forall i \in I | p_i = 1, \forall s \in S) \quad (10.1)$$

$$\sum_{h \in H} \sum_{j \in J} Y'_{isjh} = 0 \quad (\forall i \in I | p_i = 0, \forall s \neq s_i) \quad (10.2)$$

$$X_{ich} \in \{0,1\} \quad (\forall i \in I, \forall c \in C, \forall h \in H) \quad (11)$$

$$X'_{ijh} \in \{0,1\} \quad (\forall i \in I, \forall j \in J, \forall h \in H) \quad (12)$$

$$Z_{sch} \in \{0,1\} \quad (\forall s \in S, \forall c \in C, \forall h \in H) \quad (13)$$

$$Z'_{sjh} \in \{0,1\} \quad (\forall s \in S, \forall j \in J, \forall h \in H) \quad (14)$$

$$Y_{isch} \in \{0,1\} \quad (\forall i \in I, \forall s \in S, \forall c \in C, \forall h \in H) \quad (15)$$

$$Y'_{isjh} \in \{0,1\} \quad (\forall i \in I, \forall s \in S, \forall j \in J, \forall h \in H) \quad (16)$$

$$M_{sh} \in \{0,1\} \quad (\forall s \in S, \forall h \in H) \quad (17)$$

La función objetivo está compuesta por la suma de dos términos, el primero de ellos minimiza el *tardiness* total del conjunto de pacientes mientras que el segundo tendrá un valor entre 0 y 1, siendo 0 en el mejor de los casos en el que todos los pacientes han pasado por quirófano y 1 en el caso contrario.

Las restricciones R1 y R2 calculan el valor del *tardiness* de cada paciente. Si el paciente pasa por quirófano dentro del horizonte temporal, R1 compara su *due date* con dicha fecha, y si la cirugía es posterior a su fecha límite, la diferencia entre ambas será el valor del *tardiness*, en caso contrario el *tardiness* tiene un valor nulo. En el caso de que el paciente no sea asignado a ningún quirófano, la restricción R2 obliga a que su *tardiness* sea igual a la diferencia entre el último día del horizonte de planificación considerado y su *due date*, siendo esta la mayor penalización posible hasta el momento.

La restricción (1) fuerza a que cada paciente nuevo ($p_i = 1$) sea visto en consulta como máximo una vez durante el horizonte de planificación, mientras que impide a los pacientes en lista de espera a quirófanos ($p_i = 0$) pasar por consulta.

La restricción (2) asegura que cada consulta esté ocupada por un cirujano cada día.

La restricción (3.1) asegura que un cirujano trabaje o bien en una consulta, o bien en la zona de quirófanos en un mismo día. En el segundo caso la restricción (3.2) limita la presencia de cada cirujano en un número establecido de quirófanos.

El conjunto de restricciones (4.1) y (4.2) asocia qué cirujano atiende a cada paciente en consulta, teniendo en cuenta que ambos se encuentren en la misma consulta el mismo día. Por tanto, la restricción (4.3) impide que ningún cirujano sea asignado a pacientes en lista de espera a quirófanos.

La restricción (5) prohíbe que la cantidad total de tiempo asignado a cada cirujano en la etapa 1 sea mayor que su capacidad en cualquier día.

La restricción (6.1) obliga a que la programación del día de operación de un paciente respete el tiempo mínimo desde que fue atendido en consulta, en caso de nuevos pacientes, o desde el instante inicial para pacientes que no deben pasar por consulta. Al mismo tiempo la restricción (6.2) prohíbe que si un paciente no es visto en consulta éste pueda ser operado en la siguiente etapa, solo en el caso de pacientes nuevos.

La restricción (7) asegura que cada cirugía sea llevada a cabo en un quirófano y día adecuado al tipo de operación.

La restricción (8.1) asegura que un paciente que se encuentre en un quirófano en un determinado día sea operado por alguno de los cirujanos. La restricción (8.2) complementa a la anterior limitando que sea como máximo un cirujano el que lo opere, mientras que la (8.3) obliga a que el cirujano asignado se encuentre en dicho quirófano el día de la intervención.

(9.1) y (9.2) son restricciones de capacidad. La restricción (9.1) prohíbe que la cantidad total de tiempo asignado a cada cirujano en la etapa 2 sea mayor que su capacidad en cualquier día. Mientras que la restricción (9.2) se asegura de no se sobrepasase el tiempo máximo disponible de cada quirófano en cada día.

La restricción (10.1) controla que un paciente sea visto en consulta y operado por el mismo cirujano, solo en el caso de pacientes nuevos. A su vez, la restricción (10.2) impide que para pacientes en lista de espera les sea asignado un cirujano diferente al que le corresponde.

De la (11) a la (17) son restricciones binarias para las variables de decisión.

7.2 Implementación del modelo

En esta sección se va a comentar brevemente la estructura del código en C#, para dar una idea general del proceso que se lleva a cabo con su ejecución.

En primer lugar, se van a indicar las acciones que se realizan en el cuerpo principal del código (*Main*):

1. Creación de un fichero en el que volcar los datos una vez optimizado el modelo.
2. Declaración de los factores principales en forma de vector (detallados en la subsección 7.3.1) fijando su valor.
3. Automatización del proceso de ejecución del código para las distintas instancias a través de una serie de bucles. En cada iteración se combinan los valores anteriores y se calculan los datos relevantes a los pacientes, a las cirugías y a los cirujanos (subsecciones 7.3.2 y 7.3.3) de manera aleatoria, obteniendo las correspondientes instancias.
4. Cálculo del tiempo límite de ejecución de la función para la combinación de datos dada según la siguiente fórmula:

$$\text{Tiempo Límite} = I * R * H * 0.025$$

5. Llamada a la función *Optimización* pasándole los parámetros que requiere para su correcto funcionamiento y las variables en las que posteriormente volcar la solución de la correspondiente instancia.
6. Escribir los valores correspondientes en el fichero para su posterior análisis.

A continuación, de igual manera, se exponen los pasos que construyen la función *Optimización*:

1. Creación de un entorno *Gurobi* en el que se recogen todos los datos asociados al conjunto de ejecuciones de optimización.
2. Creación del modelo en *Gurobi* asociado al entorno anterior, se indica el nombre: "*Consultas y Quirófanos*".
3. Declaración de las variables de *Gurobi*, indicando cuál de ellas aparece en la función objetivo.
4. Especificar el sentido de la función objetivo, en este caso minimizar.

5. Actualizar el modelo de *Gurobi* para integrar las variables anteriores.
6. Definir una a una las restricciones de ambas etapas y las necesarias para la función objetivo y añadirlas al modelo.
7. Indicar el tiempo límite de ejecución para la instancia en concreto y generar un archivo llamado “*ejemplo.lp*” para facilitar la comprobación del modelo que se está creando.
8. Optimizar el modelo.
9. Recoger los valores interesantes para su posterior análisis y volcarlos en las variables que han sido pasadas como referencia desde el *Main*:
 - El tiempo de ejecución.
 - Si es óptimo, factible (si el tiempo de ejecución ha sido mayor que el tiempo límite) o infactible.
 - El valor de las variables *óptimo*, *bound* y *gap*. Estas variables toman los valores mostrados en la tabla 7-6 (sección 7.4) según el estado del modelo.

7.3 Diseño del banco de pruebas

Como se describió en la sección 5.4 sobre a la generación del *testbed* o banco de pruebas, se debe indicar que para llevar a cabo el análisis del modelo ILP descrito en la sección 7.1 se ha tomado como referencia el artículo “*New heuristics for planning operating rooms*” (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015), adaptando el método de generación de instancias propuesto e incorporando ciertos datos particulares para este modelo como son el número de consultas y el tiempo estimado de duración en esta primera etapa.

7.3.1 Factores principales

Los principales factores a tener en cuenta para la generación de un banco de pruebas según las contribuciones existentes en el ámbito de la planificación de quirófanos son los siguientes:

- $|H|$: número de días en el horizonte temporal.
- $|J|$: número de quirófanos.
- $|C|$: número de consultas (necesario para este caso en particular).
- $|I|$: número de pacientes en lista de espera total. Esta cantidad puede ser establecida según una lista de espera real, de forma arbitraria por el autor o generando uno a uno los pacientes hasta que la suma de sus tiempos estimados en el sistema supere un $\beta\%$ de la capacidad total disponible de los recursos en el horizonte temporal, en este caso consultas más quirófanos (se indica en el capítulo 4, descripción del problema). Para este modelo se ha optado por la última opción, considerando como tiempo medio de un paciente en el sistema la suma de la media de sus tiempos estimados en ambas etapas. En cuanto al factor β , se ha optado por un 100% y un 125% de la capacidad, escenario muy común en la realidad, en el que la duración estimada de las cirugías en lista de espera suele exceder a la capacidad disponible, debido a los presupuestos restrictivos.
- $|mds|$: número de días a la semana en los que un cirujano está disponible tanto para trabajar en consultas como en quirófanos.
- $|S|$: número de cirujanos. Al igual que el número de cirugías este valor puede ser establecido según datos reales, arbitrariamente elegido por el autor o como se ha optado en este proyecto aplicando la siguiente fórmula (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015):

$$S = \alpha * \frac{\text{Capacidad Total}}{m d s * l * a}$$

Siendo la capacidad total la suma del tiempo disponible en consultas y quirófanos durante todo el horizonte, l el número de semanas dentro del horizonte temporal, en este caso solo consideramos una semana, y a el tiempo disponible al día de un cirujano (especificado en el capítulo 4, descripción del problema). El factor de control α se ha establecido en 1.5 y 2, de forma que como mínimo cada recurso pueda estar siempre ocupado por un cirujano.

- U_s : es un entero no negativo que limita el número de quirófanos en los que puede estar presente un cirujano al día.

En la tabla 7-2 se exponen los valores elegidos para estos factores, basados en el artículo mencionado en la introducción a esta sección, considerando un número de consultas de 2 y 5 como podría darse perfectamente en un hospital real.

Tabla 7-2: Factores considerados para la generación de datos de entrada

$ H $	$ J $	$ C $	$\beta\%$	α	$ m d s $	U_s
5	3 / 6	2 / 5	100% / 125%	1.5 / 2	4	1 / $ J $

7.3.2 Datos relevantes a las cirugías

Los parámetros de los pacientes requeridos para el problema se han generado de la siguiente forma y se recogen en la tabla 7-3:

Tabla 7-3: Parámetros de pacientes considerados para la generación de datos de entrada

t_{1i}	t_{2i}	p_i	\forall_i	r_i	d_i	w_i
U[15,30]	U[90,120]	U[0,1]	U[0,S-1]	U[1,3]	U[1,10]	1,2,5

- t_{1i}, t_{2i} : los tiempos estimados de los pacientes en ambas etapas se obtienen mediante una distribución uniforme discreta dados unos valores máximos y mínimos (en minutos) establecidos de forma arbitraria como se puede observar en la tabla 7-3.
- p_i : puede valer 0 o 1 de forma aleatoria, determinando en qué lista de espera se encuentra el paciente
- \forall_i : toma un valor aleatorio entre 0 y S-1, asignando un cirujano arbitrario del conjunto S a los pacientes que solo deben pasar por quirófano.
- r_i : se ha establecido de forma arbitraria pudiendo valer de 1 a 3 días el tiempo requerido para pruebas antes de poder realizar la correspondiente cirugía. Se ha considerado como hipótesis el reprocesado del *release date* para aquellos pacientes que van directamente a quirófano, teniendo en cuenta este periodo de tiempo desde el inicio del horizonte de planificación.
- d_i : en la literatura se ha calculado este valor como la diferencia entre el tiempo máximo antes

de tratamiento (que puede ser de 30, 180 o 360 días según la gravedad) y el número de días en lista de espera (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015). En el ámbito de este estudio por simplificación, se ha considerado que la fecha límite de cirugía puede tomar un valor aleatorio entre 1 y 10 días, teniendo en cuenta que el periodo de planificación es de 5 días como se puede observar en la tabla 7-2.

- w_i : la ponderación de cada cirugía se ha calculado en otros estudios como una combinación lineal de la prioridad médica y el número de días en lista de (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015). Sin embargo, al igual que en el caso anterior, se ha decidido simplificar esta ponderación de manera que este peso sea un indicador de la urgencia médica de la operación en función de los siguientes porcentajes basados en el tiempo máximo antes de tratamiento: el 50% de los casos son de prioridad baja (360 días), el 30% de prioridad media (180 días) y el resto de prioridad alta (30 días).

Por último, se va a especificar el parámetro δ_{ijh} . Dado que en la realidad existen determinados tipos de cirugía que solo pueden ser ejecutadas en un quirófano especializado (como se indicó en la descripción del problema, capítulo 4), se ha restringido para el caso que nos ocupa, que un 20% de las intervenciones solo puedan producirse en el quirófano 1, por considerarse el mejor equipado, durante el tercer día del horizonte temporal.

7.3.3 Datos relevantes a los cirujanos

En cuanto a los cirujanos, es esencial elaborar una programación semanal que especifique cada día cuál de ellos se encuentra disponible para trabajar en el hospital. Para ello se ha seguido un procedimiento de tres pasos, con la intención de que cada cirujano esté disponible tantos días a la semana como se haya indicado anteriormente (mds) y evitar recursos inutilizados (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015):

1. En el primer paso, se asignan aleatoriamente, pertenecientes al conjunto S, tantos cirujanos como recursos existan (consultas y quirófanos), para que como mínimo siempre pueda haber un cirujano en un determinado lugar de trabajo.
2. En el segundo paso, se compara el número de días asignados en la etapa anterior con mds, añadiendo esta diferencia a cada cirujano de forma aleatoria.
3. Por último, si según los pasos anteriores un cirujano trabaja un determinado día su tiempo disponible a_{sh} valdrá 8 horas (como se estableció en el capítulo 4, descripción del problema), mientras que por el contrario si no trabaja su capacidad se reduce a 0, o lo que es lo mismo, no está disponible.

7.4 Experimentos computacionales

En esta sección se presentan los experimentos computacionales usando el modelo de decisión de ILP implementado anteriormente. Estos experimentos han sido llevados a cabo en un PC con un procesador de 2.40 GHz Intel Core i7-4700MQ y 8 GBytes de memoria RAM.

De acuerdo a los datos de entrada propuestos en la sección 7.3 y considerando las 32 posibles combinaciones de los factores $|J|$, $|C|$, β , α y U_s , se han generado 10 instancias para cada combinación, resultando en total 320 instancias que serán objeto de análisis. El tamaño de la lista de espera depende de β , $|J|$ y $|C|$, en la tabla 7-4 se muestra el número medio de cirugías para cada una de las combinaciones.

Una vez ejecutado el programa siguiendo las instrucciones descritas en la sección 7.2, se obtiene para

cada instancia un valor de cada una de las variables de interés que se muestran en la tabla 7-5 *óptimo*, *bound* y *gap*, y también se extrae si el modelo ha sido resuelto de forma óptima, factible o infactible, así como el tiempo de ejecución. En el caso de las soluciones factibles, el *lower bound* es la cota inferior y el *best bound* es la mejor solución encontrada hasta el fin de la ejecución. De esta forma, el *gap* ofrece una idea de lo cercana que está la solución obtenida de la cota inferior, siendo 0 en el mejor de los casos en los que se encuentra la solución óptima y 1 cuanto más grande es la diferencia entre ambos valores.

Solo en un caso el modelo ha sido incapaz de encontrar solución, por tanto, en total, se han obtenido 12 soluciones óptimas, 0 infactibles y 307 factibles, es decir, cuyo tiempo de ejecución ha superado el tiempo límite establecido según la fórmula siguiente comentada en la sección 7.2:

$$\text{Tiempo Límite} = I * R * H * 0.025$$

Tabla 7-4: Número medio de cirugías en función de la capacidad

		β	
$ C $	$ J $	1	1.25
2	3	94	118
2	6	150	188
5	3	150	188
5	6	207	258

Tabla 7-5: Valor de las variables tras la llamada a la función

	Es Óptimo	Es Factible	Es Infactible
Óptimo	<i>Valor Óptimo</i>	<i>Best Bound</i>	0
Bound	<i>Valor Óptimo</i>	<i>Lower Bound</i>	0
Gap	0	$\frac{ Bound - \text{Óptimo} }{\text{Óptimo}}$	0

7.4.1 Análisis de los resultados en función de los recursos

En la tabla 7-6 se recogen los datos que resultan de interés para el análisis del modelo, como son la holgura máxima, mínima y media (calculada tal y como aparece en la tabla 7-5), el porcentaje de soluciones óptimas y el tiempo de ejecución medio. Ya que en la literatura se demuestra que el recurso más limitante son los quirófanos (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015), en este estudio se ha decidido extraer dichos valores para la combinación de consultas-quirófanos, por estar consideradas ambas etapas de la planificación en este proyecto. A su vez, se puede apreciar la influencia del parámetro *Us*, denominándose programación avanzada cuando los cirujanos pueden

encontrarse en todos los quirófanos disponibles ($Us=|J|$) o bien, enfoque integrado si la movilidad se reduce a un solo quirófano ($Us=1$) (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015).

En general, se han obtenido buenas soluciones, aunque el porcentaje de optimalidad no es muy elevado (3.75% en total) debido a la complejidad del modelo y al tamaño considerable de las listas de espera (tabla 7-4), la mayoría de las soluciones factibles se aproximan bastante a su correspondiente cota inferior (*lower bound*), siendo en el peor de los casos la holgura media un 38.3% y llegando a alcanzar valores mínimos cercanos al 0%.

En la tabla 7-7 se aprecia con más detalle la influencia del parámetro Us , observándose una clara mejora en los resultados cuando se relaja la restricción de movilidad de los cirujanos, de tal forma que el gap medio se reduce en un 76.5% y se encuentra la mayor parte de las soluciones óptimas (6.25%), reduciendo así levemente el tiempo de ejecución. En cuanto a éste último, se aprecia un incremento a medida que se añaden recursos al modelo, para 5 recursos se alcanzan entorno a los 60 segundos, para 8 se requieren casi 170 en algunos casos y por último, para 11 recursos en total, se llegan a superar los 300 segundos de ejecución.

Tabla 7-6: Resultados del modelo ILP para resolver el problema de decisión

C	J	Us	Gap Min (%)	Gap Max (%)	Gap Medio (%)	Soluciones óptimas (%)	Tiempo de ejecución (s)
2	3	1	0.05	13.02	1.87	0	66.34
		3	0	4.95	0.45	10	61.89
2	6	1	5.68	75.8	28.9	0	169.11
		6	0.06	23.4	7.91	0	169.14
5	3	1	0	17.62	3.99	5	166.83
		3	0	7.18	1.92	15	160.72
5	6	1	0.09	61.02	38.3	0	320.72
		6	0.018	51.14	7.05	0	319.19

Tabla 7-7: Resultados del modelo ILP en función de Us

Us	Gap Medio (%)	Soluciones óptimas (%)	Tiempo de ejecución (s)
1	18.27	1.25	180.75
J	4.33	6.25	177.74

7.4.2 Influencia del número de pacientes y cirujanos

En esta sección se pretende conocer la influencia que ejerce el tamaño de la lista de espera y la cantidad de cirujanos disponibles en la función objetivo y el tiempo de ejecución, con la intención de esclarecer el posible alcance de este modelo exacto en un entorno hospitalario real.

Antes de analizar el valor de la función objetivo, hay que distinguir los dos factores de los que ésta se compone. Por un lado, el valor de la tardanza total (*tardiness*) de los pacientes puede extraerse truncando el valor de la función, mientras que el resto, un valor entre 0 y 1, indica la proporción de pacientes que han pasado por quirófano, siendo 0 el mejor de los casos en el que han sido planificadas todas las cirugías pendientes dentro del horizonte disponible, y 1 en caso contrario. Con estos valores, puede extraerse el porcentaje de pacientes que han sido operados dentro del horizonte de planificación con la capacidad disponible.

En la tabla 7-8 se observa una leve mejora en la tardanza total si se consideran 2 cirujanos por recurso al día, sin embargo, no tiene apenas influencia en el resto de valores, excepto un ligero incremento en el tiempo de ejecución y en el porcentaje de pacientes operados. Por el contrario, es notable la influencia del tamaño de la lista de espera. Incrementando en tan solo un 25% el número de pacientes en función de la capacidad disponible, la holgura media empeora, la tardanza total media se incrementa en 20 días aproximadamente para $\alpha=1.5$ y unos 17 días para $\alpha=2$, y a la vez se reduce el número de pacientes intervenidos en un 10% aproximadamente. Además, se corrobora la relación del número de cirugías con un aumento considerable en el tiempo de ejecución.

Tabla 7-8: Resultados del modelo ILP en función de β y α

β	α	Gap Medio (%)	Tardiness	Pacientes planificados (%)	Tiempo de ejecución (s)
1	1.5	8.9	38.58	51.25	158.24
	2	10.89	38.35	51.37	159.89
1.25	1.5	12.53	58.39	41.34	197.63
	2	12.9	55.4	41.52	200

7.5 Caso práctico: valor que toman las variables

En esta sección se detalla una solución obtenida al ejecutar el programa dada una instancia pequeña propuesta arbitrariamente con la intención de mostrar su efectividad en la creación de una planificación que se ajuste a los requisitos establecidos previamente. Para ello se ha creado un código que ejecute la misma estructura pero cuya finalidad es mostrar por pantalla el valor de las variables en forma de matrices dada una sola instancia. Los datos a tener en cuenta se muestran en la tabla 7-9 y la tabla 7-10. En cuanto al parámetro δ_{ijn} , se ha generado de forma aleatoria que el paciente 2, por ser una cirugía especial, solo puede ser operado en el quirófano 1 el día 2 (al igual que se planteó en la sección 7.3.2).

Para facilitar la comprobación de las restricciones se ha establecido un número fijo tanto de pacientes como de cirujanos y se han considerado las siguientes simplificaciones: respecto a los primeros el tiempo mínimo antes de la operación es 1 día para todos, y en cuanto a los cirujanos se consideran disponibles todos los días del horizonte de planificación.

Las capacidades son 8 horas para recursos y cirujanos como se explicó en el capítulo 4, descripción del problema.

Tabla 7-9: Datos propuestos

<i>C</i>	<i>J</i>	<i>I</i>	<i>S</i>	<i>H</i>	<i>Us</i>
2	3	7	6	4	1

Tabla 7-10: Datos de pacientes generados

Paciente	d_i	w_i	p_i	γ_i	t_{1i}	t_{2i}
0	4	1	1	-	25	115
1	4	1	1	-	15	107
2	1	5	0	5	16	120
3	1	1	1	-	28	117
4	3	1	1	-	30	101
5	4	1	1	-	25	115
6	3	2	1	-	30	98

*Se indica con un guion que para los pacientes tipo 1 no es relevante el cirujano pues se asigna en consulta

Una vez planteados los datos necesarios, se muestra a continuación el valor que han tomado las variables una vez ejecutado el programa, estableciendo un tiempo límite de 20 segundos:

1. Valor de X_{ich} :

Tal y como se observa en la tabla 7-11, todos los pacientes tipo 1 han pasado una vez por consulta durante los dos primeros días del horizonte de planificación, mientras que el paciente 2 que es el único de tipo 0 no es atendido en la primera etapa. Para el resto de días todas las variables toman el valor 0.

Tabla 7-11: Valor de X_{ich}

		Pacientes						
		0	1	2	3	4	5	6
Consultas por día	DÍA 0							
	0	0	1	0	0	1	1	0
	1	1	0	0	1	0	0	0
	DÍA 1							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	1
	DÍA 2							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	DÍA 3							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0

2. Valor de Z_{sch} :

En la siguiente tabla 7-12 se puede comprobar que toda consulta está ocupada por un cirujano al día. Además, una vez analizadas estas dos primeras tablas, podemos extraer como conclusión quién ha sido el cirujano asignado a cada paciente tipo 1, siendo el cirujano 5 el encargado del paciente 1, 4, 5; el cirujano 3 de los pacientes 0 y 3; y por último el cirujano 0 operará al paciente 6.

Tabla 7-12: Valor de Z_{sch}

		Cirujanos					
		0	1	2	3	4	5
Consultas por día	DÍA 0						
	0	0	0	0	0	0	1
	1	0	0	0	1	0	0
	DÍA 1						
	0	0	0	0	0	1	0
	1	1	0	0	0	0	0
	DÍA 2						
	0	0	0	0	0	0	1
	1	0	0	1	0	0	0
	DÍA 3						
	0	1	0	0	0	0	0
	1	0	1	0	0	0	0

3. Valor de Y_{isch} :

Esta variable es necesaria para contabilizar la carga de trabajo de los cirujanos en la primera etapa y puede ser contrastada con los datos extraídos de las tablas anteriores, de forma que solo tome el valor 1 cuando coincidan en la misma consulta y día, un determinado un cirujano y su paciente. Para simplificar el contenido de la tabla 7-13 solo se muestra el valor de los dos primeros días del horizonte de planificación, ya que el resto de días ningún paciente pasa por consulta por lo que su valor es 0. Como puede apreciarse solo se activan aquellos cirujanos que pasan consulta dicho día, y toman el valor 1 para los pacientes que atienden. Se puede comprobar con facilidad que se respeta la asignación deducida de las tablas anteriores.

A continuación, con estos valores, es sencillo comprobar que no se excede el límite de capacidad de trabajo del cirujano. Extrayendo de la tabla 7-10 los tiempos estimados para la primera etapa (t_{1i}), se sabe que en el día 0, el cirujano de la consulta 0 trabaja 70 min y el de la consulta 1 acumula un total de 28 minutos, mientras que el día siguiente solo pasa por la consulta 1 el paciente 6 con una duración aproximada de 30 minutos. En cualquier caso, este tiempo es mucho inferior a los 480 minutos (8 horas) que podría estar disponible un cirujano, esto es debido al tamaño tan pequeño del entorno que se ha tomado como ejemplo. En un caso real, las restricciones de capacidad, tanto de los recursos como de los cirujanos, sí son limitantes.

Tabla 7-13: Valor de Y_{isch}

Pacientes		DÍA 0						DÍA 1							
		0	1	2	3	4	5	6	0	1	2	3	4	5	6
Cirujanos por consulta	C=0														
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	1	0	0	1	1	0	0	0	0	0	0	0	0
	C=1														
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	1	0	0	1	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

4. Valor de X'_{ij} :

Esta variable es igual a la del punto 1 pero haciendo referencia al paso por quirófano de los pacientes. Según los valores obtenidos, se respeta un día como mínimo entre el paso por consulta (o el instante 0 en caso del paciente 2, tipo 0) y la fecha asignada para la cirugía, como se puede ver el día 0 no se lleva a cabo ninguna operación por este motivo. Como se esperaba en este problema por su tamaño reducido, todos los pacientes pasan por quirófano y cumpliendo con la fecha límite estimada tal y como se puede comprobar en la tabla 7-14.

Tabla 7-14: Valor de X'_{ijh}

		Pacientes						
		0	1	2	3	4	5	6
Quirófanos por día	DÍA 0							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0
	DÍA 1							
	0	0	0	1	0	0	1	0
	1	0	0	0	0	0	0	0
	2	1	0	0	1	0	0	0
	DÍA 2							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	1
	2	0	0	0	0	0	0	0
	DÍA 3							
	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	

5. Valor de Z'_{sjh} :

De forma similar a lo comentado en el punto 2, esta variable toma el valor 1 si un cirujano se encuentra trabajando en el quirófano j el día h . En este ejemplo un cirujano solo puede encontrarse en un quirófano al día según la variable U_s fijada en la tabla 7-9. Hay que indicar que con una batería de pacientes mayor, se asignará como mínimo un cirujano por quirófano por exigencia de la planificación, aunque en este caso tan pequeño sí puedan darse quirófanos vacíos debido a que no hay restricción que lo impida como sí ocurría en consultas, o cirujanos asignados que no lleven a cabo operación como en el día 2 quirófano 0, sin impedir la consecución del objetivo.

Para comprobar que ningún cirujano que haya sido asignado previamente a una consulta pueda encontrarse en la zona de quirófanos el mismo día, se han marcado en verde en la tabla 7-15 aquellos cirujanos que se encuentran en la primera etapa según la tabla 7-12.

Tabla 7-15: Valor de Z'_{sjh}

		Cirujanos					
		0	1	2	3	4	5
Quirófanos por día	DÍA 0						
	0	0	0	0	0	0	0
	1	0	0	0	0	0	0
	2	0	0	0	0	0	0
	DÍA 1						
	0	0	0	0	0	0	1
	1	0	0	0	0	0	0
	2	0	0	0	1	0	0
	DÍA 2						
	0	0	1	0	1	1	0
	1	1	0	0	0	0	0
	2	0	0	0	0	0	0
DÍA 3							
0	0	0	0	0	0	0	
1	0	0	0	0	0	0	
2	0	0	0	0	0	1	

6. Valor de Y'_{isjh} :

Por último, hay que analizar si se cumple la asignación de cirujanos de la etapa anterior, es decir, si la pareja cirujano-paciente coincide en el mismo quirófano-día. Este valor se verifica a través de la variable Y'_{isjh} , que al igual que en el punto 3, sirve para cuantificar posteriormente las horas de cirugía de cada cirujano. Por simplificar la tabla 7-16, se ha prescindido del día 0, ya que al no desarrollarse ninguna operación toma en todo caso el valor 0.

Una vez analizados los valores de la tabla 7-16 se puede concluir que se cumple lo siguiente:

- Las asignaciones paciente-cirujano establecidas en consulta (punto 2) se mantienen para las cirugías. En el caso particular del paciente 2, tal y como aparece en la 7-10, se respeta su asignación previa al cirujano 5.
- Los cirujanos encargados de las operaciones se encuentran en el quirófano y día adecuados de acuerdo a las tablas 7-14 y 7-15.
- La carga de trabajo por cirujano y quirófano no excede los límites de capacidad de ambos. Durante el día 1, en el quirófano 0 el tiempo total estimado es de 235 min y en el quirófano 2 de 232 min según el tiempo t_{2i} extraído de la tabla 7-10. De igual forma en el día 2- quirófano 1 la única

cirugía es de 98 min y en el día 3 se calcula un total de 208 minutos en el recurso 2. En ningún caso se alcanzan los 480 minutos disponibles por quirófano o cirujano, por el mismo motivo que se explicó anteriormente en el punto 3.

Tabla 7-16: Valor de Y'_{isjh}

Pacientes		DÍA 1							DÍA 2							DÍA 3						
		0	1	2	3	4	5	6	0	1	2	3	4	5	6	0	1	2	3	4	5	6
Cirujanos por quirófano	J=0																					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	J=1																					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	J=2																					
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	

El último paso es analizar el valor de la función objetivo. En este ejemplo ha tomado el valor 0, o lo que es lo mismo todos los pacientes pasan por quirófano sin ninguna tardanza, siendo un valor óptimo del problema de planificación, ya que ha sido encontrado en menos de 20 segundos que se estableció como tiempo límite anteriormente.

8 CONCLUSIONES

A lo largo de este trabajo se ha abordado el problema de decisión que enfrentan los hospitales a la hora de planificar a medio plazo tanto los quirófanos como la etapa previa de consultas, con el objetivo de facilitar una posterior programación a corto plazo. Así, gracias a este nivel previo de decisión, queda establecida de forma eficaz, la asignación paciente-cirujano y las correspondientes consultas y quirófanos en los que estos deben encontrarse en sus respectivas fechas, asegurando un retraso aceptable en las operaciones y un buen uso de los recursos disponibles.

Este es un problema de total actualidad y en pleno desarrollo, ya que repercute directamente en el coste de los hospitales y en la calidad de servicio (en término de tiempo de espera) que éstos ofrecen, especialmente importante en la situación actual a la que se enfrenta el sistema sanitario.

Para llevar a cabo el estudio del problema de decisión se ha enmarcado dentro de la gestión de operaciones, haciendo una comparación entre el sistema industrial y sanitario, para poder concluir que los métodos desarrollados para la programación de la producción en las fábricas se pueden adaptar al entorno hospitalario obteniendo buenos resultados. Así, se ha propuesto una metodología que tiene como objetivo la planificación de la producción a medio plazo, realizándose lo necesario para poder ser aplicada al problema de decisión del hospital.

La revisión de la literatura que abarca problemas de este tipo, especialmente centrados en la planificación de quirófanos, demuestra la efectividad de estos distintos métodos, existiendo aportaciones desde la programación matemática, pasando por el desarrollo de heurísticas constructivas o metaheurísticas hasta simulaciones de distintas naturalezas.

En este caso en particular, se ha optado por resolver el problema a través de un método exacto como un primer acercamiento a la programación conjunta de ambas etapas (consulta-quirófano). Para ello se ha desarrollado el modelo ILP cumpliendo con las restricciones y limitaciones impuestas por el hospital, en principio de forma teórica, para posteriormente programarlo en C# y resolverlo mediante el solver *Gurobi*.

Una vez generado un banco de pruebas adecuado para un análisis lo más cercano posible a la realidad, basado sobre todo en la literatura existente, se ha ejecutado el código obteniendo un conjunto de instancias que han sido analizadas. De este análisis se puede concluir un buen funcionamiento del modelo para el tamaño del banco de datos de entrada generado, aunque el porcentaje de optimalidad no alcanza el 4%, se obtienen buenas soluciones factibles con una holgura cercana al 0% para gran parte de las instancias. Esta holgura o *gap* se ha calculado como la diferencia entre la mejor solución encontrada dentro del tiempo límite establecido y la cota inferior correspondiente. También se aprecia una clara influencia del parámetro Us que se encarga de limitar la movilidad de los cirujanos, resultando ser un factor determinante a la hora de encontrar soluciones óptimas y buenas soluciones factibles.

Una de las principales conclusiones refiere al tiempo de ejecución. Al ser un modelo enumerativo, que implícitamente realiza una enumeración de todas las posibles soluciones para encontrar el óptimo, se aprecia como al incrementar el tamaño del problema, ya sean recursos o número de cirugías, el tiempo de ejecución requerido aumenta de forma considerable, lo que evidencia una limitación de este tipo de metodología exacta a la hora de poder implantarse en un entorno real.

Además de esta complejidad computacional o teórica que demuestra ser no polinomial, se añade la complejidad inherente a los entornos reales de trabajo, que incluyen aspectos como la multiplicidad y diversidad de recursos, la incertidumbre y la ambigüedad en restricciones y objetivos entre otros.

Por último, para demostrar la eficacia del modelo en cuanto al cumplimiento de las restricciones impuestas, se ha generado un pequeño banco de datos gracias al cual se ha podido comprobar fácilmente el valor que toman las distintas variables.

Tras este repaso se puede concluir que el modelo no es capaz de encontrar el óptimo, o incluso alguna solución factible para ciertas instancias, a medida que se incrementa el tamaño del problema, por lo que será necesario explorar otros métodos de resolución para una posible aplicación al hospital real. Esto mismo ocurre con la ampliación del horizonte de planificación. En cuanto a este aspecto, debe aclararse que para este estudio se ha tomado un horizonte teórico mucho más pequeño de lo que en realidad debe considerarse, más propio de un problema de programación a corto plazo. La planificación a medio plazo que se busca optimizar resulta útil cuando se aplica en horizontes de dos a cuatro semanas o más (Molina-Pariente, Hans, Framiñán & Gómez-Cia, 2015), sin embargo, debido a la complejidad que conlleva la consideración de ambas etapas, ha resultado interesante optar por un horizonte reducido de una semana (5 días) para establecer una base sobre la que construir métodos aproximados como se verá en la siguiente sección.

8.1 Líneas de futuro

Aludiendo a las conclusiones extraídas previamente, en esta sección se sugieren posibles caminos de actuación a seguir partiendo del análisis de la solución del modelo exacto.

En primer lugar, sería interesante generar un banco de pruebas basado en datos reales (según experiencia del personal encargado y datos históricos), y poder comprobar así el alcance exacto de esta solución.

A continuación, se podrían construir algoritmos aproximados como heurísticas constructivas o metaheurísticas, que comparándose con los resultados obtenidos de forma exacta, aporten soluciones buenas en un tiempo razonable, dentro de un horizonte de planificación mayor y para una lista de espera de tamaño real.

Con la ayuda de las heurísticas desarrolladas, sería posible evaluar diferentes objetivos de planificación, utilizando varios horizontes de planificación y siguiendo diferentes directrices (priorización de pacientes, reducción de la lista de espera, etc.).

REFERENCIAS

- Anand, R., Aggarwal, D. & Kumar, V. "A comparative analysis of optimization solvers". *Journal of Statistics and Management Systems*, 2017 [en línea], [fecha de consulta: 15 de junio de 2020]. Disponible en: <https://www.tandfonline.com/doi/abs/10.1080/09720510.2017.1395182>
- Cardoen, B., Demeulemeester, E. & Belien, J. *Operating room planning and scheduling: A literature review* [en línea]. Brussel: Katholieke Universiteit Leuven, 2009. Disponible en: <https://core.ac.uk/download/pdf/6347245.pdf>
- Framiñán, J.M., Leisten, R. & Ruiz, R. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools* [en línea]. Londres: Springer London Heidelberg New York Dordrecht, 2014. Disponible en: <https://bib.us.es/ingenieros/>
- GOBIERNO de Aragón, 2020. "Situación de las listas de espera". En: *Listas de espera* [en línea], [consulta: 06 de junio de 2020]. Disponible en: <https://leweb.salud.aragon.es/listaespera/estadis.do>
- GUROBI Optimization, 2020. "Gurobi optimizer". En: *Products* [en línea], [consulta: 16 de junio de 2020]. Disponible en: <https://www.gurobi.com/products/gurobi-optimizer/>
- Jebali, A. & Diabat, A. "A stochastic model for operating room planning under capacity constraints". *International journal of production research*, 2015 [en línea]. Disponible en: <https://www.tandfonline.com/doi/abs/10.1080/00207543.2015.1033500>
- Litvak, E., Prenney, B., Fuda, K., Levtzion-Korach, O. & McGlinchey, P., 2007. *Improving Patient Flow and Throughput in California Hospital Operating Room Services*. Boston university program for management of variability in health care delivery, California Healthcare Foundation. Citado por: Ortiga Fontgivell, Berta. *La dirección de operaciones en un hospital de alta tecnología: medidas para incrementar la productividad y la eficiencia*. [Barcelona]: UAB, 2013, p.50:51
- MICROSOFT, 2020. "Acerca de Visual Studio". En: "Microsoft Docs" [en línea], [consulta: 18 de junio de 2020]. Disponible en: <https://docs.microsoft.com/es-es/visualstudio/get-started/visual-studio-ide?view=vs-2019>
- MICROSOFT, 2020. "Paseo por C#". En: *Documentación de C#* [en línea], [consulta: 18 de junio de 2020]. Disponible en: <https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/>
- MINISTERIO de Sanidad, Consumo y Bienestar Social, 2020. "Sistema de Información de Listas de Espera del Sistema Nacional de Salud". En: *Sanidad* [en línea], [consulta: 09 de junio de 2020]. Disponible en: <https://www.msbs.gob.es/estadEstudios/estadisticas/inforRecopilaciones/listaEspera.htm>
- Molina-Pariente, J.M., Hans, E.W., Framiñán, J.M. & Gómez-Cia, T. "New Heuristics for planning operating rooms". *Computer & Industrial Engineering*, 2015 [en línea]. Disponible en: www.elsevier.com/locate/caie
- Ortiga Fontgivell, Berta, 2013. *La dirección de operaciones en un hospital de alta tecnología: medidas para incrementar la productividad y la eficiencia* [en línea], [consulta: 08 de junio de 2020]. Joan Escarrabill Sanglas, Albert Jovell Fernández, dir. Tesis Doctoral. Universidad Autónoma de Barcelona, Facultad de Medicina, Barcelona. Disponible en: <https://ddd.uab.cat/>
- Perez, P., Framiñán, J.M. & Fernandez-Viagas, V. *Programación de Operaciones*. Universidad de Sevilla: Escuela Técnica Superior de Ingeniería, 2018.
- Reiss, M. *Komplexitätsmanagement: Das Wirtschaftsstudium*, 1993. Citado por: Framiñán, J.M., Leisten, R. & Ruiz, R. *Manufacturing Scheduling Systems: An Integrated View on Models, Methods and Tools*. [London]: Springer, 2014, p. 27
- Vissers, JMH & Beech, R. *Health operations management: patient flow logistics in health care*.

Londres: Routledge health management series, Routledge Taylor & Francis Group, 2005.

Anexo 1: Código en C#

```
using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Gurobi;

namespace Prueba
{
    class Program
    {
        static void Main()
        {
            //Generación de un fichero
            String ruta = AppDomain.CurrentDomain.BaseDirectory;
            ruta = ruta.Replace("\\", "/").ToString();
            DirectoryInfo DIRESC = new DirectoryInfo(ruta + "/temp"); //Añadir System.IO; para generar
            el fichero
            if (!DIRESC.Exists)
            {
                DIRESC.Create();
            }
            String ficBatBorrar = ruta + "temp" + "/Experimentacion.csv";
            FileInfo FicheroEliminar = new FileInfo(ficBatBorrar);
            if (FicheroEliminar.Exists)
            {
                File.Delete(ficBatBorrar);
            }
            String ficCSV = ruta + "temp" + "/Experimentacion.csv";
```

```

//Declaramos los parámetros y les damos un valores posibles
Random generador = new Random();

double[] Beta = new double[] { 1, 1.25 }; //Probaremos con el 100% de la capacidad total y el
125% de ésta
double[] Alfa = new double[] { 1.5, 2 }; //Factor de control para generar el número de cirujanos
int[] J = new int[] { 3, 6 };
int[] C = new int[] { 2, 5 };
int H = 5; //Fijo el horizonte temporal en 5 días
int mds = 4; //los cirujanos trabajan 4 de 5 días a la semana

//Variables para recoger el análisis de las soluciones
int esOptimo=0, esFactible=0, esInfactible=0;
double Optimo=0, bound=0, gap=0;
double tiempo = 0;

//Para el encabezado del fichero
System.IO.StreamWriter FicheroCSV1 = new System.IO.StreamWriter(ficCSV, true);
FicheroCSV1.Write("Beta"+";"+"Alfa" + ";" + "C" + ";" + "J" + ";"+"Us" + ";" + "I" + ";" + "S"
+ ";" + "Iteracion" + ";" + " " + ";"+"Gap"+";"+"Bound"+";"+"Optimo"+";"+"Es Optimo"+";"+"Es
Factible"+";"+"Es Infactible"+";"+"Tiempo"+"\\n");
FicheroCSV1.Close();

for (int b=0; b<Beta.Length; b++)
{
for(int a=0; a<Alfa.Length; a++)
{
for (int c = 0; c < C.Length; c++)
{
for (int j = 0; j < J.Length; j++)
{
int[] J_aux = new int[] { 1, J[j] };

for (int u = 0; u < J_aux.Length; u++)
{

```

```

int R = C[c] + J[j];

//Cálculo de S
int S = 0;
double capacidad_S = R * 480 * H * Alfa[a];

while (capacidad_S > 0)
{
    S++;
    capacidad_S -= (mds * 480); //mds=4 días/sem l=1 sem 8h/día
}

//Ásignación a los cirujanos del límite de quirófanos o bien 1 o bien J
int[] U_s = new int[S];
for (int s = 0; s < S; s++)
{
    U_s[s] = J_aux[u];
}

//Cálculo de I
int I = 0; //Contador que nos da el número de pacientes
double capacidad_I = R * 480 * H * Beta[b];
while (capacidad_I > 0)
{
    I++;
    capacidad_I -= (23 + 105); //Tiempo medio de los pacientes en el sistema
}

//Cálculo del tiempo limite
double tiempoLimite = I * R * H * 0.025; //en segundos

//Iteraciones
for (int it = 0; it < 10; it++)
{

```

```
//Asignación de la disponibilidad de cirujanos
```

//Etapa 1: creación de una matriz aleatoria de cirujanos: por cada día (fila) aparece un cirujano asignado a cada recurso

```
int[,] Cirujanos = new int[H, R];
int aleatorio = 0;
int stop;
int aux;

for (int h = 0; h < H; h++)
{
    Cirujanos[h, 0] = generador.Next(0, S); //Al primer recurso le asigno
cualquiera para cada día
```

```
for (int r = 1; r < R; r++)
{
    stop = 1;
    while (stop == 1)
    {
        aleatorio = generador.Next(0, S);
        aux = 0;
        stop = 0;
        while (aux < r && stop == 0)
        {
            if (Cirujanos[h, aux] == aleatorio)
            {
                stop = 1;
            }
            aux++;
        }
    }

    Cirujanos[h, r] = aleatorio;
}
}
```

```
//Etapa 2: Creación de una matriz que indica la disponibilidad de cirujanos cada
```

día y sumamos el total de días trabajados por cada uno

```
int[,] Disponible = new int[S, H];  
int[] Suma_s = new int[S];
```

```
for (int s = 0; s < S; s++)  
{  
    for (int h = 0; h < H; h++)  
    {  
        int trabaja = 0;  
        for (int r = 0; r < R; r++)  
        {  
            if (Cirujanos[h, r] == s)  
            {  
                trabaja = 1;  
                Suma_s[s]++;  
            }  
        }  
        Disponible[s, h] = trabaja;  
    }  
}
```

disponibilidad //Comparamos el total de días asignados con mds=4 y actualizamos la matriz

```
for (int s = 0; s < S; s++)  
{  
    int diferencia_s = mds - Suma_s[s]; //¿Se podría dar el caso de que la suma  
    sea mayor que mds?  
    for (int h = 0; h < H; h++)  
    {  
        if (Disponible[s, h] == 0 && diferencia_s > 0)  
        {  
            Disponible[s, h] = 1;  
            diferencia_s--;  
        }  
    }  
}
```

//Con la matriz disponibilidad se crea la matriz ash en la que aparecen 480 min
si el cirujano esta disponible para trabajar ese día y 0 si no trabaja

```
int[,] a_sh = new int[S, H];  
for (int h = 0; h < H; h++)  
{  
    for (int s = 0; s < S; s++)  
    {  
        if (Disponibile[s, h] == 1)  
        {  
            a_sh[s, h] = 480;  
        }  
        else  
        {  
            a_sh[s, h] = 0;  
        }  
    }  
}
```

//Valores del resto de datos para las iteraciones

```
int[] w_i = new int[I];  
int[] p_i = new int[I]; //Nuevo parámetro que indica si un paciente debe pasar
```

o no por consulta

```
int[] gamma_i = new int[I]; //Cirujanos asignados  
int[] d_i = new int[I];  
int[] r_i = new int[I];  
int[] t1_i = new int[I];  
int[] t2_i = new int[I];
```

double sem;

```
for (int i = 0; i < I; i++)
```

```
{
```

//Generación de los pesos según unos porcentajes

```
sem = generador.NextDouble();
```

```
if (sem > 0.2)
```

```
{
```

```
    if (sem > 0.5)
```

```
    {
```

```

        w_i[i] = 1; //El 50% de los casos son de prioridad baja (MTBT=360 días)

    }
    else
    {
        w_i[i] = 2; //El 30% de los casos son de prioridad media (MTBT=180
días)

    }
}
else
{
    w_i[i] = 5; //El 20% de los casos son de prioridad alta (MTBT=30 días)
}

p_i[i] = generador.Next(0, 2); //Binario
gamma_i[i] = generador.Next(0, S); //Asigna de 0 a S-1
d_i[i] = generador.Next(1, 11);
r_i[i] = generador.Next(1, 3);
t1_i[i] = generador.Next(15, 31); //Lo he reducido entre un cuarto y media
hora

t2_i[i] = generador.Next(90, 121); //El tiempo de operación es entre 90 y 120
min

}

int[,] m_jh = new int[J[j], H];
for (int jj = 0; jj < J[j]; jj++)
{
    for (int h = 0; h < H; h++)
    {
        m_jh[jj, h] = 480; //8h por quirófano cada día (480 min)
    }
}

//Un 20% de los pacientes solo podrán ser operados en un día y quirófano
determinado

double val;
int[,] delta_ijh = new int[I, J[j], H];
for (int i = 0; i < I; i++)

```

```

    {
        val = generador.NextDouble();
        for (int jj = 0; jj < J[j]; jj++)
        {
            for (int h = 0; h < H; h++)
            {
                if (val > 0.2)
                {
                    delta_ijh[i, jj, h] = 1;
                }
                else
                {
                    if (h == 2 & jj == 1) //Solo se pueden operar el día 2 en el quirófano 1
en el 20% de los casos
                    {
                        delta_ijh[i, jj, h] = 1;
                    }
                    else
                    {
                        delta_ijh[i, jj, h] = 0;
                    }
                }
            }
        }
    }

//Llamamos a la función Optimización
Optimizacion(I, S, J[j], C[c], H, w_i, p_i, gamma_i, d_i, r_i, t1_i, t2_i, U_s,
a_sh, m_jh, delta_ijh, tiempoLimite, ref esOptimo, ref esFactible, ref esInfactible, ref Optimo, ref gap,
ref bound, ref tiempo);

//A la salida de cada optimización.
FicheroCSV1 = new System.IO.StreamWriter(ficCSV, true);

//Escribimos en el fichero los valores de cada iteración
FicheroCSV1.Write(Beta[b] + ";" + Alfa[a] + ";" + C[c] + ";" + J[j]
+ ";" + J_aux[u] + ";" + I + ";" + S + ";" + it + ";" + "-->" + ";" + gap + ";" + bound + ";" + Optimo + ";"

```

```
+ esOptimo + ";" + esFactible + ";" + esInfactible + ";" + tiempo + "\n");
```

```
        FicheroCSV1.Close();
    }
}
}
}
}
}
}
}
```

```
private static void Optimizacion(int I, int S, int J, int C, int H,int[] w_i, int[]p_i, int[] gamma_i,
int[] d_i, int[] r_i, int[] t1_i, int[] t2_i, int[] U_s, int[,] a_sh, int[,] m_jh, int[,] delta_ijh, double
tiempoLimite, ref int esOptimo, ref int esFactible, ref int esInfactible, ref double Optimo, ref double
gap, ref double bound, ref double tiempo)
```

```
{
    try
    {
        GRBEnv env = new GRBEnv("mip1.log");
        GRBModel model = new GRBModel(env);
        model.Set(GRB.StringAttr.ModelName, "Consultas y Quirófanos");

        ////////////////////////////////////////////////////
        //Declaración de variables (GUROBI)//
        ////////////////////////////////////////////////////

        GRBVar[,] X_ich = new GRBVar[I, C, H];
        for (int i = 0; i < I; i++)
        {
            for (int c = 0; c < C; c++)
            {
                for (int h = 0; h < H; h++)
                {
                    X_ich[i, c, h] = model.AddVar(0, 1, 0, GRB.BINARY, "X_" + i + "_" + c + "_" +
h); //Variable Objetivo
                }
            }
        }
    }
}
```

```

    }
  }
}

GRBVar[,] Xp_ijh = new GRBVar[I, J, H];
for (int i = 0; i < I; i++)
{
  for (int j = 0; j < J; j++)
  {
    for (int h = 0; h < H; h++)
    {
      Xp_ijh[i, j, h] = model.AddVar(0, 1, 0, GRB.BINARY, "Xp_" + i + "_" + j + "_" +
h); //Variable objetivo
    }
  }
}

```

```

GRBVar[,] Z_sch = new GRBVar[S, C, H];
for (int s = 0; s < S; s++)
{
  for (int c = 0; c < C; c++)
  {
    for (int h = 0; h < H; h++)
    {
      Z_sch[s, c, h] = model.AddVar(0, 1, 0, GRB.BINARY, "Z_" + s + "_" + c + "_" +
h);
    }
  }
}

```

```

GRBVar[,] Zp_sjh = new GRBVar[S, J, H];
for (int s = 0; s < S; s++)
{
  for (int j = 0; j < J; j++)
  {
    for (int h = 0; h < H; h++)
    {
      Zp_sjh[s, j, h] = model.AddVar(0, 1, 0, GRB.BINARY, "Zp_" + s + "_" + j + "_" +

```

```

h);
    }
    }
}

GRBVar[,,] Y_isch = new GRBVar[I, S, C, H];
for (int i = 0; i < I; i++)
{
    for (int s = 0; s < S; s++)
    {
        for (int c = 0; c < C; c++)
        {
            for (int h = 0; h < H; h++)
            {
                Y_isch[i, s, c, h] = model.AddVar(0, 1, 0, GRB.BINARY, "Y_" + i + "_" + s +
                "_" + c + "_" + h);
            }
        }
    }
}

GRBVar[,,] Yp_isjh = new GRBVar[I, S, J, H];
for (int i = 0; i < I; i++)
{
    for (int s = 0; s < S; s++)
    {
        for (int j = 0; j < J; j++)
        {
            for (int h = 0; h < H; h++)
            {
                Yp_isjh[i, s, j, h] = model.AddVar(0, 1, 0, GRB.BINARY, "Yp_" + i + "_" + s +
                "_" + j + "_" + h);
            }
        }
    }
}

```

```

//Nueva variable para indicar si un cirujano se encuentra en zona de quirófanos el día h
GRBVar[,] M_sh = new GRBVar[S, H];
for (int s = 0; s < S; s++)
{
    for (int h = 0; h < H; h++)
    {
        M_sh[s, h] = model.AddVar(0, 1, 0, GRB.BINARY, "M_" + s + "_" + h);
    }
}

//Variable Tardiness para la función objetivo
GRBVar[] T_i = new GRBVar[I];
for (int i = 0; i < I; i++)
{
    T_i[i] = model.AddVar(0, GRB.INFINITY, 0, GRB.INTEGER, "T_" + i); //Prioridad
clínica 1 para todos los pacientes
}

//Variable Objetivo
GRBVar objetivo = new GRBVar();
objetivo = model.AddVar(0, GRB.INFINITY, 1, GRB.CONTINUOUS, "Objetivo");

////////////////////////////////////
// El OBJETIVO es minimizar(+1) el tardiness en las operaciones maximizando la presencia
en quirófano //
////////////////////////////////////

model.Set(GRB.IntAttr.ModelSense, 1);

////////////////////////////////////
// Actualizar el modelo para integrar nuevas variables//
////////////////////////////////////

model.Update();

////////////////////////////////////

```

```
//RESTRICCIONES DEL MODELO//
```

```
////////////////////////////////////
```

```
//Restricción Objetivo
```

```
double Inv_I = (double)1 / (double)I;
```

```
GRBLinExpr restr_objetivo = 0;
```

```
for (int i = 0; i < I; i++)
```

```
{
```

```
    restr_objetivo += w_i[i] * T_i[i];
```

```
}
```

```
restr_objetivo += 1;
```

```
for (int i = 0; i < I; i++)
```

```
{
```

```
    for (int j = 0; j < J; j++)
```

```
    {
```

```
        for (int h = 0; h < H; h++)
```

```
        {
```

```
            restr_objetivo -= Inv_I * Xp_ijh[i, j, h];
```

```
        }
```

```
    }
```

```
}
```

```
model.AddConstr(restr_objetivo - objetivo == 0, "Restricción Objetivo");
```

```
//Restricción para cálculo del Tardiness pacientes operados en el horizonte H
```

```
for (int i = 0; i < I; i++)
```

```
{
```

```
    GRBLinExpr ladoIzqT_1 = 0;
```

```
    ladoIzqT_1 += T_i[i];
```

```
    for (int j = 0; j < J; j++)
```

```
    {
```

```
        for (int h = 0; h < H; h++)
```

```

    {
        ladoIzqT_1 -= Xp_ijh[i, j, h] * h; //Cálculo del día de operación
    }
}

ladoIzqT_1 += d_i[i];

model.AddConstr(ladoIzqT_1 >= 0, "Tardiness_1");

}

//Restricción para cálculo del Tardiness complementaria para pacientes no operados en H
int Ult_dia = H - 1;
for (int i = 0; i < I; i++)
{
    GRBLinExpr ladoIzqT_2 = 0;

    ladoIzqT_2 += T_i[i];
    ladoIzqT_2 -= Ult_dia;
    ladoIzqT_2 += d_i[i];
    for (int j = 0; j < J; j++)
    {
        for (int h = 0; h < H; h++)
        {
            ladoIzqT_2 += Xp_ijh[i, j, h] * (Ult_dia - d_i[i]); //Cálculo del día de operación
        }
    }

    model.AddConstr(ladoIzqT_2 >= 0, "Tardiness_2");

}

```

//ETAPA 1//

//Restricción 1: fuerza a que cada paciente sea visto en consulta como máximo una vez durante el horizonte de planificación

```

for (int i = 0; i < I; i++)
{
    GRBLinExpr ladoIzqRestriccion1 = 0;

    for (int c = 0; c < C; c++)
    {
        for (int h = 0; h < H; h++)
        {
            ladoIzqRestriccion1 += X_ich[i, c, h];
        }
    }
    ladoIzqRestriccion1 -= p_i[i];

    model.AddConstr(ladoIzqRestriccion1 <= 0, "Restriccion1");
}

```

//Restricción 2: asegura que cada consulta esté ocupada por un cirujano cada día.

```

for (int c = 0; c < C; c++)
{
    for (int h = 0; h < H; h++)
    {
        GRBLinExpr ladoIzqRestriccion2 = 0;

        for (int s = 0; s < S; s++)
        {
            ladoIzqRestriccion2 += Z_sch[s, c, h];
        }
        model.AddConstr(ladoIzqRestriccion2 == 1, "Restriccion2");
    }
}

```

//Restriccion 3.1: asegura que un cirujano trabaje o bien en una consulta, o bien en la zona de quirófanos en un mismo día.

```

for (int s = 0; s < S; s++)
{

```

```

for (int h = 0; h < H; h++)
{
    GRBLinExpr ladoIzqRestriccion3_1 = 0;

    for (int c = 0; c < C; c++)
    {
        ladoIzqRestriccion3_1 += Z_sch[s, c, h];
    }
    ladoIzqRestriccion3_1 += M_sh[s, h];

    model.AddConstr(ladoIzqRestriccion3_1 <= 1, "Restriccion3_1");
}
}

```

//Restricción 3.2: limita la presencia de cada cirujano en un número establecido de quirófanos.

```

for (int s = 0; s < S; s++)
{
    for (int h = 0; h < H; h++)
    {
        GRBLinExpr ladoIzqRestriccion3_2 = 0;
        for (int j = 0; j < J; j++)
        {
            ladoIzqRestriccion3_2 += Zp_sjh[s, j, h];
        }
        ladoIzqRestriccion3_2 -= M_sh[s, h] * U_s[s];

        model.AddConstr(ladoIzqRestriccion3_2 <= 0, "Restriccion3_2");
    }
}
}

```

//Restricción 4.1 y 4.2: asocia qué cirujano atiende a cada paciente en consulta, teniendo en cuenta que ambos se encuentren en la misma consulta el mismo día.

//Restricción 4.1:

```

for (int i = 0; i < I; i++)
{
    for (int c = 0; c < C; c++)

```

```

{
  for (int s = 0; s < S; s++)
  {
    for (int h = 0; h < H; h++)
    {
      GRBLinExpr ladoIzqRestriccion4_1 = 0;

      ladoIzqRestriccion4_1 += X_ich[i, c, h];
      ladoIzqRestriccion4_1 += Z_sch[s, c, h];
      ladoIzqRestriccion4_1 -= Y_isch[i, s, c, h];

      model.AddConstr(ladoIzqRestriccion4_1 <= 1, "Restriccion4_1");
    }
  }
}

```

//Restricción 4.2:

```

for (int i = 0; i < I; i++)
{
  for (int c = 0; c < C; c++)
  {
    for (int s = 0; s < S; s++)
    {
      for (int h = 0; h < H; h++)
      {
        GRBLinExpr ladoIzqRestriccion4_2 = 0;

        ladoIzqRestriccion4_2 += X_ich[i, c, h];
        ladoIzqRestriccion4_2 += Z_sch[s, c, h];
        ladoIzqRestriccion4_2 -= 2 * Y_isch[i, s, c, h];

        model.AddConstr(ladoIzqRestriccion4_2 >= 0, "Restriccion4_2");
      }
    }
  }
}

```

```
}
```

//Restricción 4.3: prohíbe que un cirujano en consulta sea asignado a algún paciente en lista de espera para quirófano

```
for (int i = 0; i < I; i++)
{
    if (p_i[i] == 0)
    {
        GRBLinExpr ladoIzqRestriccion4_3 = 0;

        for (int s = 0; s < S; s++)
        {
            for (int c = 0; c < C; c++)
            {
                for (int h = 0; h < H; h++)
                {
                    ladoIzqRestriccion4_3 += Y_isch[i, s, c, h];
                }
            }
        }

        model.AddConstr(ladoIzqRestriccion4_3 == 0, "Restriccion4_3");
    }
}
```

//Restricción 5: prohíbe que la cantidad total de tiempo asignado a cada cirujano en la etapa 1 sea mayor que su capacidad en cualquier día.

```
for (int s = 0; s < S; s++)
{
    for (int h = 0; h < H; h++)
    {

        GRBLinExpr ladoIzqRestriccion5 = 0;

        for (int i = 0; i < I; i++)
        {
            for (int c = 0; c < C; c++)
```

```

        {
            ladoIzqRestriccion5 += t1_i[i] * Y_isch[i, s, c, h];
        }
    }
    ladoIzqRestriccion5 -= a_sh[s, h];

    model.AddConstr(ladoIzqRestriccion5 <= 0, "Restriccion5");

}
}

//ETAPA 2//

```

//Restricción 6.1: obliga a que la programación del día de operación de un paciente respete el tiempo mínimo desde que fue atendido en consulta.

```

for (int i = 0; i < I; i++)
{
    GRBLinExpr ladoIzqRestriccion6_1 = 0;

    for (int j = 0; j < J; j++)
    {
        for (int h = 0; h < H; h++)
        {
            ladoIzqRestriccion6_1 += Xp_ijh[i, j, h] * h;
        }
    }
    int numeroAlto = 1000;
    for (int j = 0; j < J; j++)
    {
        for (int h = 0; h < H; h++)
        {
            ladoIzqRestriccion6_1 -= numeroAlto * Xp_ijh[i, j, h];
        }
    }
    for (int c = 0; c < C; c++)
    {
        for (int h = 0; h < H; h++)

```

```

    {
        ladoIzqRestriccion6_1 -= X_ich[i, c, h] * h;
    }
}
ladoIzqRestriccion6_1 -= r_i[i];

model.AddConstr(ladoIzqRestriccion6_1 >= -1 * numeroAlto, "Restriccion6_1");
}

```

//Restricción 6.2: prohíbe que si un paciente no es visto en consulta éste pueda ser operado en la siguiente etapa (solo válido para pacientes nuevos)

```

for (int i = 0; i < I; i++)
{
    if (p_i[i] == 1)
    {
        GRBLinExpr ladoIzqRestriccion6_2 = 0; //Quitar punto entre GRB LinExpr

        for (int c = 0; c < C; c++)
        {
            for (int h = 0; h < H; h++)
            {
                ladoIzqRestriccion6_2 += X_ich[i, c, h];
            }
        }
        for (int j = 0; j < J; j++)
        {
            for (int h = 0; h < H; h++)
            {
                ladoIzqRestriccion6_2 -= Xp_ijh[i, j, h];
            }
        }

        model.AddConstr(ladoIzqRestriccion6_2 >= 0, "Restriccion6_2");
    }
}
}

```

//Restricción 7: asegura que cada cirugía sea llevada a cabo en un quirófano y día adecuado

al tipo de operación.

```
for (int i = 0; i < I; i++)
{
    for (int j = 0; j < J; j++)
    {
        for (int h = 0; h < H; h++)
        {
            if (delta_ijh[i, j, h] == 0)
            {
                GRBLinExpr ladoIzqRestriccion7 = 0;

                ladoIzqRestriccion7 += Xp_ijh[i, j, h];

                model.AddConstr(ladoIzqRestriccion7 == 0, "Restriccion7");
            }
        }
    }
}
```

//Restricción 8.1: asegura que un paciente que se encuentre en un quirófano en un determinado día sea operado por alguno de los cirujanos.

```
for (int i = 0; i < I; i++)
{
    for (int j = 0; j < J; j++)
    {
        for (int h = 0; h < H; h++)
        {
            GRBLinExpr ladoIzqRestriccion8_1 = 0;

            ladoIzqRestriccion8_1 += Xp_ijh[i, j, h];
            for (int s = 0; s < S; s++)
            {
                ladoIzqRestriccion8_1 -= Yp_isjh[i, s, j, h];
            }

            model.AddConstr(ladoIzqRestriccion8_1 <= 0, "Restriccion8_1");
        }
    }
}
```

```

    }
  }
}

```

//Restricción 8.2: complementa a la anterior limitando que sea como máximo un cirujano el que lo opere.

```

for (int i = 0; i < I; i++)
{
  for (int j = 0; j < J; j++)
  {
    for (int h = 0; h < H; h++)
    {
      GRBLinExpr ladoIzqRestriccion8_2 = 0;

      for (int s = 0; s < S; s++)
      {
        ladoIzqRestriccion8_2 += Yp_isjh[i, s, j, h];
      }

      model.AddConstr(ladoIzqRestriccion8_2 <= 1, "Restriccion8_2");
    }
  }
}

```

//Restricción 8.3: obliga a que el cirujano asignado se encuentre en dicho quirófano el día de la intervención.

```

for (int i = 0; i < I; i++)
{
  for (int j = 0; j < J; j++)
  {
    for (int s = 0; s < S; s++)
    {
      for (int h = 0; h < H; h++)
      {
        GRBLinExpr ladoIzqRestriccion8_3 = 0;

```

```

ladoIzqRestriccion8_3 += Xp_ijh[i, j, h];
ladoIzqRestriccion8_3 += Zp_sjh[s, j, h];
ladoIzqRestriccion8_3 = 2 * Yp_isjh[i, s, j, h];

model.AddConstr(ladoIzqRestriccion8_3 >= 0, "Restriccion8_3");
}
}
}
}

```

//Restricción 9.1: prohíbe que la cantidad total de tiempo asignado a cada cirujano en la etapa 2 sea mayor que su capacidad en cualquier día.

```

for (int s = 0; s < S; s++)
{
    for (int h = 0; h < H; h++)
    {
        GRBLinExpr ladoIzqRestriccion9_1 = 0;

        for (int i = 0; i < I; i++)
        {
            for (int j = 0; j < J; j++)
            {
                ladoIzqRestriccion9_1 += t2_i[i] * Yp_isjh[i, s, j, h];
            }
        }
        ladoIzqRestriccion9_1 -= a_sh[s, h];

        model.AddConstr(ladoIzqRestriccion9_1 <= 0, "Restriccion9_1");
    }
}

```

//Restricción 9.2: se asegura de no sobrepasar el tiempo máximo disponible de cada quirófano en cada día.

```

for (int j = 0; j < J; j++)
{
    for (int h = 0; h < H; h++)

```

```

{
  GRBLinExpr ladoIzqRestriccion9_2 = 0;

  for (int i = 0; i < I; i++)
  {
    ladoIzqRestriccion9_2 += t2_i[i] * Xp_ijh[i, j, h];
  }
  ladoIzqRestriccion9_2 -= m_jh[j, h];

  model.AddConstr(ladoIzqRestriccion9_2 <= 0, "Restriccion9_2");
}
}

```

//Restricción 10.1: controla que un paciente sea visto en consulta y operado por el mismo cirujano (pacientes nuevos)

```

for (int i = 0; i < I; i++)
{
  for (int s = 0; s < S; s++)
  {
    if (p_i[i] == 1)
    {
      GRBLinExpr ladoIzqRestriccion10_1 = 0;

      for (int h = 0; h < H; h++)
      {
        for (int c = 0; c < C; c++)
        {
          ladoIzqRestriccion10_1 += Y_isch[i, s, c, h];
        }
      }
      for (int h = 0; h < H; h++)
      {
        for (int j = 0; j < J; j++)
        {
          ladoIzqRestriccion10_1 -= Yp_isjh[i, s, j, h];
        }
      }
    }
  }
}

```

```

        model.AddConstr(ladoIzqRestriccion10_1 >= 0, "Restriccion10_1");
    }
}
}

```

//Restricción 10.2: para pacientes en LE para quirófano, prohíbe que éstos sean asignados a un cirujano que no les corresponde

```

for (int i = 0; i < I; i++)
{
    if (p_i[i] == 0)
    {
        for (int s = 0; s < S; s++)
        {
            if (gamma_i[i] != s)
            {
                GRBLinExpr ladoIzqRestriccion10_2 = 0;

                for (int h = 0; h < H; h++)
                {
                    for (int j = 0; j < J; j++)
                    {
                        ladoIzqRestriccion10_2 -= Yp_isjh[i, s, j, h];
                    }
                }

                model.AddConstr(ladoIzqRestriccion10_2 == 0, "Restriccion10_2");
            }
        }
    }
}
}
}
}

```

//////////

//OPTIMIZAR//

//////////

```

model.GetEnv().Set(GRB.DoubleParam.TimeLimit, tiempoLimite);

model.Write("ejemplo.lp");

//Optimizamos
model.Optimize();

//Tiempo de ejecución en segundos
tiempo = model.Get(GRB.DoubleAttr.Runtime);

//Recogemos el análisis de las soluciones
int optimstatus = model.Get(GRB.IntAttr.Status);

if (optimstatus == GRB.Status.INF_OR_UNBD) //O bien es infactible o ilimitado, para
obtener conclusiones más definitivas se reoptimiza
{
    model.GetEnv().Set(GRB.IntParam.Presolve, 0);
    model.Optimize();
    tiempo = model.Get(GRB.DoubleAttr.Runtime);
    optimstatus = model.Get(GRB.IntAttr.Status);
}

if (optimstatus == GRB.Status.OPTIMAL) //Si es óptimo
{
    Optimo = model.Get(GRB.DoubleAttr.ObjVal);
    esOptimo = 1;
    esFactible = 0;
    esInfactible = 0;
    bound = Optimo;
    gap = 0;
}
else if (optimstatus == GRB.Status.INFEASIBLE)
{
    //Optimo = model.Get(GRB.DoubleAttr.ObjVal);
    esFactible = 0;
    esInfactible = 1;
    Optimo = 0;
}

```

```

        bound = 0;
        gap = 0;
    }
    else if (optimstatus == 9) //Es factible, el tiempo de ejecución es mayor que el especificado
    {
        esOptimo = 0;
        esFactible = 1;
        esInfactible = 0;
        Optimo = model.Get(GRB.DoubleAttr.ObjVal); //Best Bound
        bound = model.Get(GRB.DoubleAttr.ObjBound); //Lower bound
        gap = Math.Abs(bound - Optimo) / Math.Abs(Optimo); //Holgura
    }

}

catch (GRBException e)
{
    Console.WriteLine("Error code: " + e.ErrorCode + ". " + e.Message);
}
}
}
}

```

Anexo 2: Resultados volcados en el fichero

β	α	C	J	U	I	S	It	Gap	Bound	Opt	O	F	I	T_ejec
1	1,5	2	3	1	94	10	0	0,00104822	30,4148936	30,4468085	0	1	0	58,851429
1	1,5	2	3	1	94	10	1	0,0017311	18,4042553	18,4361702	0	1	0	58,7687263
1	1,5	2	3	1	94	10	2	0,00137836	15,4148936	15,4361702	0	1	0	58,8092594
1	1,5	2	3	1	94	10	3	0,00183038	17,4042553	17,4361702	0	1	0	58,9799576
1	1,5	2	3	1	94	10	4	0,00104822	30,4148936	30,4468085	0	1	0	59,0762558
1	1,5	2	3	1	94	10	5	0,00503145	8,41489362	8,45744681	0	1	0	58,7625065
1	1,5	2	3	1	94	10	6	0,00784314	5,38297872	5,42553191	0	1	0	59,0960541
1	1,5	2	3	1	94	10	7	0,00074766	28,4361702	28,4574468	0	1	0	58,7768326
1	1,5	2	3	1	94	10	8	0,00256191	12,4255319	12,4574468	0	1	0	58,779129
1	1,5	2	3	1	94	10	9	0,00056802	37,4361702	37,4574468	0	1	0	58,7924595
1	1,5	2	3	3	94	10	0	0,00221076	14,4042553	14,4361702	0	1	0	58,8662033
1	1,5	2	3	3	94	10	1	0,001221	17,4042553	17,4255319	0	1	0	58,808094
1	1,5	2	3	3	94	10	2	0,00130605	24,4042553	24,4361702	0	1	0	58,8080807
1	1,5	2	3	3	94	10	3	0,00137931	15,4042553	15,4255319	0	1	0	58,7937889
1	1,5	2	3	3	94	10	4	0,00129534	16,4042553	16,4255319	0	1	0	58,7924767
1	1,5	2	3	3	94	10	5	0,00719424	4,40425532	4,43617021	0	1	0	58,7848701
1	1,5	2	3	3	94	10	6	0,00099305	21,4042553	21,4255319	0	1	0	58,7835922
1	1,5	2	3	3	94	10	7	0,00063674	33,393617	33,4148936	0	1	0	58,79245
1	1,5	2	3	3	94	10	8	0,00503145	8,41489362	8,45744681	0	1	0	58,8248634
1	1,5	2	3	3	94	10	9	0,00305499	10,4148936	10,4468085	0	1	0	58,8080788
1	1,5	2	6	1	150	15	0	0,29032944	22,26	31,3666667	0	1	0	150,139034
1	1,5	2	6	1	150	15	1	0,1505418	29,2666667	34,4533333	0	1	0	150,068165
1	1,5	2	6	1	150	15	2	0,26004832	22,46	30,3533333	0	1	0	150,070488
1	1,5	2	6	1	150	15	3	0,06405907	31,2666667	33,4066667	0	1	0	150,083754
1	1,5	2	6	1	150	15	4	0,6094037	21,2666667	54,4466667	0	1	0	150,099558
1	1,5	2	6	1	150	15	5	0,46853575	28,94	54,4533333	0	1	0	150,088432
1	1,5	2	6	1	150	15	6	0,2274162	34,2666667	44,3533333	0	1	0	150,086725

1	1,5	2	6	1	150	15	7	0,16831819	40,2866667	48,44	0	1	0	150,083715
1	1,5	2	6	1	150	15	8	0,07553551	38,2666667	41,3933333	0	1	0	150,085173
1	1,5	2	6	1	150	15	9	0,14899437	35,26	41,4333333	0	1	0	150,083849
1	1,5	2	6	6	150	15	0	0,00232964	34,26	34,34	0	1	0	150,217447
1	1,5	2	6	6	150	15	1	0,05234564	73,26	77,3066667	0	1	0	150,086187
1	1,5	2	6	6	150	15	2	0,00158144	25,2533333	25,2933333	0	1	0	150,161901
1	1,5	2	6	6	150	15	3	0,00185682	32,2533333	32,3133333	0	1	0	150,077932
1	1,5	2	6	6	150	15	4	0,0017735	26,2666667	26,3133333	0	1	0	150,080154
1	1,5	2	6	6	150	15	5	0,08420187	22,26	24,3066667	0	1	0	150,03734
1	1,5	2	6	6	150	15	6	0,15298347	39,2733333	46,3666667	0	1	0	150,082954
1	1,5	2	6	6	150	15	7	0,00589275	11,2466667	11,3133333	0	1	0	150,114935
1	1,5	2	6	6	150	15	8	0,15469332	33,26	39,3466667	0	1	0	150,107111
1	1,5	2	6	6	150	15	9	0,07253886	26,2533333	28,3066667	0	1	0	150,131758
1	1,5	5	3	1	150	15	0	0,0652229	49,2066667	52,64	0	1	0	150,168045
1	1,5	5	3	1	150	15	1	0,17627797	30,1866667	36,6466667	0	1	0	150,178171
1	1,5	5	3	1	150	15	2	0,00014607	45,6333333	45,64	0	1	0	150,036146
1	1,5	5	3	1	150	15	3	0,02715949	31,76	32,6466667	0	1	0	150,375654
1	1,5	5	3	1	150	15	4	0	35,6466667	35,6466667	1	0	0	66,4168835
1	1,5	5	3	1	150	15	5	0,0008016	16,62	16,6333333	0	1	0	150,052544
1	1,5	5	3	1	150	15	6	0,06902185	30,3933333	32,6466667	0	1	0	150,12175
1	1,5	5	3	1	150	15	7	0,00057159	46,6266667	46,6533333	0	1	0	150,144989
1	1,5	5	3	1	150	15	8	0,12068157	37,5	42,6466667	0	1	0	150,1423
1	1,5	5	3	1	150	15	9	0,00044984	29,6266667	29,64	0	1	0	150,156071
1	1,5	5	3	3	150	15	0	0,00020421	32,64	32,6466667	0	1	0	150,138666
1	1,5	5	3	3	150	15	1	0	25,6266667	25,6266667	1	0	0	142,12598
1	1,5	5	3	3	150	15	2	0,01887007	57,54	58,6466667	0	1	0	150,15366
1	1,5	5	3	3	150	15	3	0,01727862	57,6333333	58,6466667	0	1	0	150,171667
1	1,5	5	3	3	150	15	4	0,00020429	32,6266667	32,6333333	0	1	0	150,195234
1	1,5	5	3	3	150	15	5	0,00039489	50,6266667	50,6466667	0	1	0	150,113737
1	1,5	5	3	3	150	15	6	0,0220599	44,6266667	45,6333333	0	1	0	150,160608

1	1,5	5	3	3	150	15	7	0,02269673	43,6333333	44,6466667	0	1	0	150,557695
1	1,5	5	3	3	150	15	8	0,00018202	36,62	36,6266667	0	1	0	150,167866
1	1,5	5	3	3	150	15	9	0,03202022	30,6333333	31,6466667	0	1	0	150,087093
1	1,5	5	6	1	207	21	0	0,49896885	66,8985507	133,521739	0	1	0	284,75845
1	1,5	5	6	1	207	21	1	0,41215342	61,4541063	104,541063	0	1	0	284,741085
1	1,5	5	6	1	207	21	2	0,30037296	35,3429952	50,5169082	0	1	0	284,717131
1	1,5	5	6	1	207	21	3	0,24903854	42,4492754	56,52657	0	1	0	284,711796
1	1,5	5	6	1	207	21	4	0,59195821	32,4541063	79,5362319	0	1	0	284,719545
1	1,5	5	6	1	207	21	5	0,24880284	41,700515	55,5120773	0	1	0	284,721252
1	1,5	5	6	1	207	21	6	1	56,4541063	1,00E+100	0	1	0	284,672598
1	1,5	5	6	1	207	21	7	0,24401023	31,4009662	41,5362319	0	1	0	284,718616
1	1,5	5	6	1	207	21	8	0,18357367	31,4541063	38,52657	0	1	0	284,732069
1	1,5	5	6	1	207	21	9	0,27553095	61,9613527	85,52657	0	1	0	284,722973
1	1,5	5	6	6	207	21	0	0,00052987	36,4492754	36,468599	0	1	0	285,249456
1	1,5	5	6	6	207	21	1	0,00031178	77,4492754	77,47343	0	1	0	284,810141
1	1,5	5	6	6	207	21	2	0,0716423	52,4589372	56,5072464	0	1	0	284,721821
1	1,5	5	6	6	207	21	3	0,00045496	42,4541063	42,47343	0	1	0	284,830488
1	1,5	5	6	6	207	21	4	0,13996471	49,4541063	57,5024155	0	1	0	284,763437
1	1,5	5	6	6	207	21	5	0,16191005	36,4589372	43,5024155	0	1	0	284,722361
1	1,5	5	6	6	207	21	6	0,00061056	47,4444444	47,47343	0	1	0	284,76351
1	1,5	5	6	6	207	21	7	0,03763505	77,4541063	80,4830918	0	1	0	284,723694
1	1,5	5	6	6	207	21	8	0,00061192	39,4492754	39,47343	0	1	0	284,793112
1	1,5	5	6	6	207	21	9	0,00047744	40,4541063	40,47343	0	1	0	284,818172
1	2	2	3	1	94	13	0	0,00137836	15,4148936	15,4361702	0	1	0	59,1037235
1	2	2	3	1	94	13	1	0,00130548	24,4148936	24,4468085	0	1	0	58,7974834
1	2	2	3	1	94	13	2	0,0942928	19,4148936	21,4361702	0	1	0	58,7961483
1	2	2	3	1	94	13	3	0,00316706	13,393617	13,4361702	0	1	0	58,9272022
1	2	2	3	1	94	13	4	0,0030581	10,4042553	10,4361702	0	1	0	58,8078213
1	2	2	3	1	94	13	5	0,12547051	7,41489362	8,4787234	0	1	0	58,8125362
1	2	2	3	1	94	13	6	0,13018598	6,46808511	7,43617021	0	1	0	58,8080959

1	2	2	3	1	94	13	7	0,00131441	40,4148936	40,4680851	0	1	0	58,8080769
1	2	2	3	1	94	13	8	0,00092678	34,4042553	34,4361702	0	1	0	58,8109913
1	2	2	3	1	94	13	9	0,00429185	7,40425532	7,43617021	0	1	0	58,7809067
1	2	2	3	3	94	13	0	0,001221	17,4042553	17,4255319	0	1	0	58,789257
1	2	2	3	3	94	13	1	0,00080483	26,4148936	26,4361702	0	1	0	58,8132935
1	2	2	3	3	94	13	2	0,01310044	2,40425532	2,43617021	0	1	0	58,8093395
1	2	2	3	3	94	13	3	0,00077549	27,4148936	27,4361702	0	1	0	58,9783115
1	2	2	3	3	94	13	4	0,0027881	11,4148936	11,4468085	0	1	0	58,8092995
1	2	2	3	3	94	13	5	0,00090827	23,4042553	23,4255319	0	1	0	58,7791367
1	2	2	3	3	94	13	6	0,0018622	11,4042553	11,4255319	0	1	0	59,045496
1	2	2	3	3	94	13	7	0,00158479	13,4042553	13,4255319	0	1	0	58,7929077
1	2	2	3	3	94	13	8	0,00225734	9,40425532	9,42553191	0	1	0	58,8115177
1	2	2	3	3	94	13	9	0,00061125	17,393617	17,4042553	0	1	0	58,7937584
1	2	2	6	1	150	20	0	0,20950413	38,26	48,4	0	1	0	150,122723
1	2	2	6	1	150	20	1	0,43634497	40,26	71,4266667	0	1	0	150,127195
1	2	2	6	1	150	20	2	0,05681818	18,26	19,36	0	1	0	150,108492
1	2	2	6	1	150	20	3	0,42175091	41,26	71,3533333	0	1	0	150,107037
1	2	2	6	1	150	20	4	0,37559071	20,26	32,4466667	0	1	0	150,099733
1	2	2	6	1	150	20	5	0,10958415	33,26	37,3533333	0	1	0	150,100121
1	2	2	6	1	150	20	6	0,25023518	21,2533333	28,3466667	0	1	0	150,082514
1	2	2	6	1	150	20	7	0,2474695	38,66	51,3733333	0	1	0	150,146227
1	2	2	6	1	150	20	8	0,06574112	15,2533333	16,3266667	0	1	0	150,099386
1	2	2	6	1	150	20	9	0,16325536	57,2333333	68,4	0	1	0	150,099075
1	2	2	6	6	150	20	0	0,00354072	26,2666667	26,36	0	1	0	150,19311
1	2	2	6	6	150	20	1	0,05206037	37,2666667	39,3133333	0	1	0	150,115105
1	2	2	6	6	150	20	2	0,04680365	22,2666667	23,36	0	1	0	150,146204
1	2	2	6	6	150	20	3	0,07069789	27,2533333	29,3266667	0	1	0	150,134766
1	2	2	6	6	150	20	4	0,11594937	23,28	26,3333333	0	1	0	150,177538
1	2	2	6	6	150	20	5	0,00207254	19,26	19,3	0	1	0	150,208927
1	2	2	6	6	150	20	6	0,22600897	17,26	22,3	0	1	0	150,083729

1	2	2	6	6	150	20	7	0,0051713	10,26	10,3133333	0	1	0	150,093933
1	2	2	6	6	150	20	8	0,00135639	39,2666667	39,32	0	1	0	150,116116
1	2	2	6	6	150	20	9	0,11600811	23,2666667	26,32	0	1	0	150,113729
1	2	5	3	1	150	20	0	0,01643143	31,1266667	31,6466667	0	1	0	150,366007
1	2	5	3	1	150	20	1	0,00061249	32,6333333	32,6533333	0	1	0	150,061022
1	2	5	3	1	150	20	2	0,00065246	30,6333333	30,6533333	0	1	0	150,053598
1	2	5	3	1	150	20	3	0,02781547	19,1066667	19,6533333	0	1	0	150,052511
1	2	5	3	1	150	20	4	0,02962678	33,6266667	34,6533333	0	1	0	150,052574
1	2	5	3	1	150	20	5	0,05406811	24,26	25,6466667	0	1	0	150,059452
1	2	5	3	1	150	20	6	0,00032803	40,6333333	40,6466667	0	1	0	150,051283
1	2	5	3	1	150	20	7	0,0253267	56,1866667	57,6466667	0	1	0	150,08374
1	2	5	3	1	150	20	8	0,06944198	35,02	37,6333333	0	1	0	150,052551
1	2	5	3	1	150	20	9	0,07463472	46,8666667	50,6466667	0	1	0	150,060768
1	2	5	3	3	150	20	0	0,00029214	45,6266667	45,64	0	1	0	151,208635
1	2	5	3	3	150	20	1	0,01424797	69,6466667	70,6533333	0	1	0	150,209768
1	2	5	3	3	150	20	2	0,0431601	42,7133333	44,64	0	1	0	150,062365
1	2	5	3	3	150	20	3	0	31,6466667	31,6466667	1	0	0	57,1220264
1	2	5	3	3	150	20	4	0,02136139	81,8533333	83,64	0	1	0	150,099375
1	2	5	3	3	150	20	5	0,07108886	24,74	26,6333333	0	1	0	150,070242
1	2	5	3	3	150	20	6	0,00672064	43,3533333	43,6466667	0	1	0	150,051249
1	2	5	3	3	150	20	7	0,00100503	6,62666667	6,63333333	0	1	0	150,214188
1	2	5	3	3	150	20	8	0,00071505	18,6333333	18,6466667	0	1	0	150,10037
1	2	5	3	3	150	20	9	0,00012663	52,64	52,6466667	0	1	0	150,22447
1	2	5	6	1	207	28	0	0,49261507	48,4589372	95,5072464	0	1	0	284,725042
1	2	5	6	1	207	28	1	0,51955808	23,3188406	48,5362319	0	1	0	284,729887
1	2	5	6	1	207	28	2	0,46550472	35,5555556	66,5217391	0	1	0	284,72541
1	2	5	6	1	207	28	3	0,36985466	29,9516908	47,531401	0	1	0	284,745504
1	2	5	6	1	207	28	4	0,41828654	46,8405797	80,5217391	0	1	0	284,957769
1	2	5	6	1	207	28	5	0,42849026	53,4541063	93,531401	0	1	0	284,770859
1	2	5	6	1	207	28	6	0,50081552	56,1835749	112,550725	0	1	0	284,747969

1	2	5	6	1	207	28	7	0,60326759	41,8792271	105,560386	0	1	0	284,746889
1	2	5	6	1	207	28	8	0,4569854	36,6521739	67,4975845	0	1	0	284,772213
1	2	5	6	1	207	28	9	0,26397406	50,4444444	68,5362319	0	1	0	284,745613
1	2	5	6	6	207	28	0	0,00111037	43,4589372	43,5072464	0	1	0	284,791204
1	2	5	6	6	207	28	1	0,11635784	42,8502415	48,4927536	0	1	0	284,776793
1	2	5	6	6	207	28	2	0,11728896	52,5362319	59,5169082	0	1	0	284,903286
1	2	5	6	6	207	28	3	0,00061044	47,4541063	47,4830918	0	1	0	284,842535
1	2	5	6	6	207	28	4	0,03639666	64,0772947	66,4975845	0	1	0	284,792431
1	2	5	6	6	207	28	5	0,00081688	35,4541063	35,4830918	0	1	0	284,739998
1	2	5	6	6	207	28	6	0,00079439	36,4589372	36,4879227	0	1	0	288,95068
1	2	5	6	6	207	28	7	0,00046389	62,4541063	62,4830918	0	1	0	284,752985
1	2	5	6	6	207	28	8	0,09615385	47,4541063	52,5024155	0	1	0	284,778017
1	2	5	6	6	207	28	9	0,00107411	31,4492754	31,4830918	0	1	0	286,473379
1,25	1,5	2	3	1	118	10	0	0,00114679	29,5254237	29,559322	0	1	0	73,8227215
1,25	1,5	2	3	1	118	10	1	0,03197206	30,5338983	31,5423729	0	1	0	73,8083649
1,25	1,5	2	3	1	118	10	2	0,00069061	24,5254237	24,5423729	0	1	0	73,8227234
1,25	1,5	2	3	1	118	10	3	0,00163577	15,5169492	15,5423729	0	1	0	74,0727024
1,25	1,5	2	3	1	118	10	4	0,02730761	36,5254237	37,5508475	0	1	0	73,7693195
1,25	1,5	2	3	1	118	10	5	0,04173429	35,0254237	36,5508475	0	1	0	73,8070927
1,25	1,5	2	3	1	118	10	6	0,02156478	46,5254237	47,5508475	0	1	0	73,7793674
1,25	1,5	2	3	1	118	10	7	0,0148327	24,2033898	24,5677966	0	1	0	73,7914696
1,25	1,5	2	3	1	118	10	8	0,00052356	48,5338983	48,559322	0	1	0	73,7929649
1,25	1,5	2	3	1	118	10	9	0,00095785	26,5169492	26,5423729	0	1	0	73,7914715
1,25	1,5	2	3	3	118	10	0	0,03948242	24,5338983	25,5423729	0	1	0	73,7914696
1,25	1,5	2	3	3	118	10	1	0,00020392	41,5508475	41,559322	0	1	0	73,7927322
1,25	1,5	2	3	3	118	10	2	0,00020903	40,5338983	40,5423729	0	1	0	73,7960072
1,25	1,5	2	3	3	118	10	3	0,00031928	26,5338983	26,5423729	0	1	0	73,8852215
1,25	1,5	2	3	3	118	10	4	0	12,5169492	12,5169492	1	0	0	18,5767555
1,25	1,5	2	3	3	118	10	5	0,00083195	30,5338983	30,559322	0	1	0	73,7914715
1,25	1,5	2	3	3	118	10	6	0,00144858	17,5254237	17,5508475	0	1	0	73,7982216

1,25	1,5	2	3	3	118	10	7	0,00088731	9,54237288	9,55084746	0	1	0	73,8070946
1,25	1,5	2	3	3	118	10	8	0	28,5254237	28,5254237	1	0	0	52,2149239
1,25	1,5	2	3	3	118	10	9	0	22,5254237	22,5254237	1	0	0	20,4216366
1,25	1,5	2	6	1	188	15	0	0,35457558	40,4042553	62,6010638	0	1	0	188,106958
1,25	1,5	2	6	1	188	15	1	0,35890959	32,3989362	50,537234	0	1	0	188,191872
1,25	1,5	2	6	1	188	15	2	0,45704973	27,4680851	50,5904255	0	1	0	188,233692
1,25	1,5	2	6	1	188	15	3	0,2661688	55,4042553	75,5	0	1	0	188,103197
1,25	1,5	2	6	1	188	15	4	0,2942652	43,3989362	61,4946809	0	1	0	188,081335
1,25	1,5	2	6	1	188	15	5	0,39356325	43,3989362	71,5638298	0	1	0	188,09693
1,25	1,5	2	6	1	188	15	6	0,24473871	50,9680851	67,4840426	0	1	0	188,096998
1,25	1,5	2	6	1	188	15	7	0,15220017	78,3989362	92,4734043	0	1	0	188,105425
1,25	1,5	2	6	1	188	15	8	0,15802572	32,393617	38,4734043	0	1	0	188,128277
1,25	1,5	2	6	1	188	15	9	0,52389656	37,4095745	78,5744681	0	1	0	188,098045
1,25	1,5	2	6	6	188	15	0	0,23407775	39,4042553	51,4468085	0	1	0	188,10457
1,25	1,5	2	6	6	188	15	1	0,0015625	20,393617	20,4255319	0	1	0	188,112576
1,25	1,5	2	6	6	188	15	2	0,11404562	31,4042553	35,4468085	0	1	0	188,143816
1,25	1,5	2	6	6	188	15	3	0,00078475	47,4095745	47,4468085	0	1	0	188,073467
1,25	1,5	2	6	6	188	15	4	0,13596649	38,3989362	44,4414894	0	1	0	188,096867
1,25	1,5	2	6	6	188	15	5	0,04151877	47,3989362	49,4521277	0	1	0	188,20628
1,25	1,5	2	6	6	188	15	6	0,1290762	54,4095745	62,4734043	0	1	0	188,215488
1,25	1,5	2	6	6	188	15	7	0,14656371	29,393617	34,4414894	0	1	0	188,179995
1,25	1,5	2	6	6	188	15	8	0,15719893	43,4042553	51,5	0	1	0	188,11257
1,25	1,5	2	6	6	188	15	9	0,10348694	42,5319149	47,4414894	0	1	0	188,112556
1,25	1,5	5	3	1	188	15	0	0,13251962	37,0478723	42,7074468	0	1	0	188,175055
1,25	1,5	5	3	1	188	15	1	0	64,712766	64,712766	1	0	0	180,347471
1,25	1,5	5	3	1	188	15	2	0,03069124	55,9414894	57,712766	0	1	0	188,050159
1,25	1,5	5	3	1	188	15	3	0,00866116	65,143617	65,712766	0	1	0	188,155441
1,25	1,5	5	3	1	188	15	4	0,02109276	81,9574468	83,7234043	0	1	0	188,204559
1,25	1,5	5	3	1	188	15	5	0,03092975	111,159574	114,707447	0	1	0	188,234003
1,25	1,5	5	3	1	188	15	6	0,05006319	51,9787234	54,7180851	0	1	0	188,242437

1,25	1,5	5	3	1	188	15	7	0,0478394	71,143617	74,7180851	0	1	0	188,050056
1,25	1,5	5	3	1	188	15	8	0,05818013	43,0531915	45,712766	0	1	0	188,206312
1,25	1,5	5	3	1	188	15	9	0,07608622	72,7287234	78,7180851	0	1	0	188,221931
1,25	1,5	5	3	3	188	15	0	0	58,712766	58,712766	1	0	0	138,836889
1,25	1,5	5	3	3	188	15	1	0,00688147	99,0265957	99,712766	0	1	0	188,144775
1,25	1,5	5	3	3	188	15	2	0,0353724	82,6808993	85,712766	0	1	0	188,049971
1,25	1,5	5	3	3	188	15	3	0,03203285	75,2234043	77,712766	0	1	0	188,050068
1,25	1,5	5	3	3	188	15	4	0,03745873	44,9680851	46,7180851	0	1	0	188,050013
1,25	1,5	5	3	3	188	15	5	0,07179362	40,5744681	43,712766	0	1	0	188,06559
1,25	1,5	5	3	3	188	15	6	0,02645646	57,1542553	58,7074468	0	1	0	188,159502
1,25	1,5	5	3	3	188	15	7	0	96,7180851	96,7180851	1	0	0	149,037313
1,25	1,5	5	3	3	188	15	8	0,02198455	52,5319149	53,712766	0	1	0	188,300133
1,25	1,5	5	3	3	188	15	9	0	53,7180851	53,7180851	1	0	0	125,209591
1,25	1,5	5	6	1	258	21	0	0,30075429	67,5503876	96,6046512	0	1	0	354,851521
1,25	1,5	5	6	1	258	21	1	0,51298255	41,2209302	84,6395349	0	1	0	355,012819
1,25	1,5	5	6	1	258	21	2	0,48691717	62,3992248	121,616279	0	1	0	354,870792
1,25	1,5	5	6	1	258	21	3	0,57026124	26,9069767	62,6124031	0	1	0	354,885765
1,25	1,5	5	6	1	258	21	4	0,43617484	75,3449612	133,631783	0	1	0	355,000051
1,25	1,5	5	6	1	258	21	5	0,42104915	33,9224806	58,5930233	0	1	0	354,866674
1,25	1,5	5	6	1	258	21	6	0,00091921	71,5543214	71,620155	0	1	0	354,883089
1,25	1,5	5	6	1	258	21	7	0,24612772	70,5542636	93,5891473	0	1	0	354,891949
1,25	1,5	5	6	1	258	21	8	0,35858767	55,5542636	86,6124031	0	1	0	354,897718
1,25	1,5	5	6	1	258	21	9	0,48327203	56,6317829	109,596899	0	1	0	354,92577
1,25	1,5	5	6	6	258	21	0	0,05262326	69,7093023	73,5813953	0	1	0	354,862459
1,25	1,5	5	6	6	258	21	1	0,00025641	75,5620155	75,5813953	0	1	0	354,860899
1,25	1,5	5	6	6	258	21	2	0,22508695	64,7713178	83,5852713	0	1	0	354,878132
1,25	1,5	5	6	6	258	21	3	0,00018586	62,5503876	62,5620155	0	1	0	355,017622
1,25	1,5	5	6	6	258	21	4	0,0530967	100,918605	106,577519	0	1	0	354,913975
1,25	1,5	5	6	6	258	21	5	0,00029221	79,5620155	79,5852713	0	1	0	354,891216
1,25	1,5	5	6	6	258	21	6	0,02040816	96,5581395	98,5697674	0	1	0	354,874306

1,25	1,5	5	6	6	258	21	7	0,18480024	42,8643411	52,5813953	0	1	0	354,876335
1,25	1,5	5	6	6	258	21	8	0,01889858	70,2248062	71,5775194	0	1	0	354,868448
1,25	1,5	5	6	6	258	21	9	0,51143114	31,5581395	64,5930233	0	1	0	355,079254
1,25	2	2	3	1	118	13	0	0,00083195	30,5338983	30,559322	0	1	0	73,812788
1,25	2	2	3	1	118	13	1	0,0022531	22,5169492	22,5677966	0	1	0	73,7854462
1,25	2	2	3	1	118	13	2	0,00196618	21,5084746	21,5508475	0	1	0	73,8313255
1,25	2	2	3	1	118	13	3	0,00069061	24,5254237	24,5423729	0	1	0	73,7876511
1,25	2	2	3	1	118	13	4	0,00400641	10,5338983	10,5762712	0	1	0	73,8154335
1,25	2	2	3	1	118	13	5	0,02052335	48,5338983	49,5508475	0	1	0	73,8242283
1,25	2	2	3	1	118	13	6	0,00118659	28,5338983	28,5677966	0	1	0	73,8024139
1,25	2	2	3	1	118	13	7	0,03495702	28,5423729	29,5762712	0	1	0	73,7762966
1,25	2	2	3	1	118	13	8	0,12184633	25,9576271	29,559322	0	1	0	73,8354168
1,25	2	2	3	1	118	13	9	0,02900107	30,6440678	31,559322	0	1	0	73,8756027
1,25	2	2	3	3	118	13	0	0,00026867	31,5338983	31,5423729	0	1	0	73,8483562
1,25	2	2	3	3	118	13	1	0,03583278	24,6271186	25,5423729	0	1	0	73,8866501
1,25	2	2	3	3	118	13	2	0,00082508	20,5254237	20,5423729	0	1	0	73,817461
1,25	2	2	3	3	118	13	3	0,04950495	19,5254237	20,5423729	0	1	0	73,8364582
1,25	2	2	3	3	118	13	4	0,00096665	17,5169492	17,5338983	0	1	0	73,989996
1,25	2	2	3	3	118	13	5	0	27,5338983	27,5338983	1	0	0	26,8944893
1,25	2	2	3	3	118	13	6	0,00073475	11,5254237	11,5338983	0	1	0	73,8098145
1,25	2	2	3	3	118	13	7	0,000297	28,5254237	28,5338983	0	1	0	73,7853355
1,25	2	2	3	3	118	13	8	0,000594	28,5169492	28,5338983	0	1	0	73,8204136
1,25	2	2	3	3	118	13	9	0,00030788	27,5169492	27,5254237	0	1	0	73,8104134
1,25	2	2	6	1	188	20	0	0,27129187	32,4042553	44,4680851	0	1	0	188,128502
1,25	2	2	6	1	188	20	1	0,34550767	42,893617	65,537234	0	1	0	188,142477
1,25	2	2	6	1	188	20	2	0,75809272	27,2287234	112,558511	0	1	0	188,050598
1,25	2	2	6	1	188	20	3	0,23621297	55,3989362	72,5319149	0	1	0	188,129547
1,25	2	2	6	1	188	20	4	0,35098192	39,9042553	61,4840426	0	1	0	188,125751
1,25	2	2	6	1	188	20	5	0,58045903	17,4042553	41,4840426	0	1	0	188,116594
1,25	2	2	6	1	188	20	6	0,20625103	51,1968085	64,5	0	1	0	188,120407

1,25	2	2	6	1	188	20	7	0,39569848	15,393617	25,4734043	0	1	0	188,117378
1,25	2	2	6	1	188	20	8	0,2273127	51,4042553	66,5265957	0	1	0	188,227808
1,25	2	2	6	1	188	20	9	0,18722729	65,393617	80,4574468	0	1	0	188,126865
1,25	2	2	6	6	188	20	0	0,16529792	25,4095745	30,4414894	0	1	0	188,13023
1,25	2	2	6	6	188	20	1	0,00065898	48,3989362	48,4308511	0	1	0	188,108204
1,25	2	2	6	6	188	20	2	0,05415755	18,393617	19,4468085	0	1	0	188,171446
1,25	2	2	6	6	188	20	3	0,22892255	57,4042553	74,4468085	0	1	0	188,136566
1,25	2	2	6	6	188	20	4	0,00152306	24,4095745	24,4468085	0	1	0	188,114981
1,25	2	2	6	6	188	20	5	0,05114274	112,404255	118,462766	0	1	0	188,307234
1,25	2	2	6	6	188	20	6	0,06174699	46,393617	49,4468085	0	1	0	188,154623
1,25	2	2	6	6	188	20	7	0,09743261	37,3989362	41,4361702	0	1	0	188,247927
1,25	2	2	6	6	188	20	8	0,1322125	46,3989362	53,4680851	0	1	0	188,140512
1,25	2	2	6	6	188	20	9	0,13793103	25,3989362	29,462766	0	1	0	188,327742
1,25	2	5	3	1	188	20	0	0,01372213	51,9946809	52,7180851	0	1	0	188,084789
1,25	2	5	3	1	188	20	1	0,04951941	41,5531915	43,7180851	0	1	0	188,06035
1,25	2	5	3	1	188	20	2	0,04956751	82,4095745	86,7074468	0	1	0	188,128767
1,25	2	5	3	1	188	20	3	0,0003736	42,6968085	42,712766	0	1	0	188,08313
1,25	2	5	3	1	188	20	4	0,07733664	23,7340426	25,7234043	0	1	0	188,071531
1,25	2	5	3	1	188	20	5	0,01251977	79,712766	80,7234043	0	1	0	188,106104
1,25	2	5	3	1	188	20	6	0,01646811	59,7234043	60,7234043	0	1	0	188,09523
1,25	2	5	3	1	188	20	7	0,10481152	31,0744681	34,712766	0	1	0	188,070129
1,25	2	5	3	1	188	20	8	0,02969028	72,4893617	74,7074468	0	1	0	188,069527
1,25	2	5	3	1	188	20	9	0,02713556	42,5265957	43,712766	0	1	0	188,107838
1,25	2	5	3	3	188	20	0	0,05014287	54,8138298	57,7074468	0	1	0	188,318499
1,25	2	5	3	3	188	20	1	0,02712477	62,9574468	64,712766	0	1	0	188,068897
1,25	2	5	3	3	188	20	2	0,019447	56,5904255	57,712766	0	1	0	188,069155
1,25	2	5	3	3	188	20	3	0,00013689	77,7021277	77,712766	0	1	0	188,150351
1,25	2	5	3	3	188	20	4	0,02452419	68,9734043	70,7074468	0	1	0	188,900217
1,25	2	5	3	3	188	20	5	0	33,712766	33,712766	1	0	0	101,239632
1,25	2	5	3	3	188	20	6	0,05247209	63,2021277	66,7021277	0	1	0	188,491705

1,25	2	5	3	3	188	20	7	0,01944044	50,7074468	51,712766	0	1	0	188,081512
1,25	2	5	3	3	188	20	8	0,04730539	93,0904255	97,712766	0	1	0	188,188282
1,25	2	5	3	3	188	20	9	0,02202412	49,6010638	50,7180851	0	1	0	188,118111
1,25	2	5	6	1	258	28	0	0,33617254	32,9263566	49,6007752	0	1	0	355,04166
1,25	2	5	6	1	258	28	1	0,28583207	60,4302326	84,6162791	0	1	0	354,908312
1,25	2	5	6	1	258	28	2	0,40527029	69,3643862	116,631783	0	1	0	354,911661
1,25	2	5	6	1	258	28	3	0,21012179	56,5620155	71,6085271	0	1	0	354,896318
1,25	2	5	6	1	258	28	4	0,61029437	44,6937984	114,686047	0	1	0	354,907269
1,25	2	5	6	1	258	28	5	0,36150414	64,8914729	101,631783	0	1	0	354,905376
1,25	2	5	6	1	258	28	6	0,45690288	77,4418605	142,593023	0	1	0	354,896013
1,25	2	5	6	1	258	28	7	0,59022377	46,5620155	113,627907	0	1	0	354,890713
1,25	2	5	6	1	258	28	8	0,31090991	65,1937984	94,6085271	0	1	0	355,138687
1,25	2	5	6	1	258	28	9	0,36328884	74,8875969	117,616279	0	1	0	356,037174
1,25	2	5	6	6	258	28	0	0,04822325	94,7829457	99,5852713	0	1	0	355,017588
1,25	2	5	6	6	258	28	1	0,1242497	56,5503876	64,5736434	0	1	0	357,411947
1,25	2	5	6	6	258	28	2	0,07181434	116,573643	125,593023	0	1	0	354,917128
1,25	2	5	6	6	258	28	3	0,14116683	36,5736434	42,5852713	0	1	0	354,942646
1,25	2	5	6	6	258	28	4	0,09368444	48,5581395	53,5775194	0	1	0	355,00482
1,25	2	5	6	6	258	28	5	0,09368444	48,5581395	53,5775194	0	1	0	355,00482
1,25	2	5	6	6	258	28	6	0,09368444	48,5581395	53,5775194	0	1	0	355,00482
1,25	2	5	6	6	258	28	7	0,09368444	48,5581395	53,5775194	0	1	0	355,00482
1,25	2	5	6	6	258	28	8	0,09368444	48,5581395	53,5775194	0	1	0	355,00482
1,25	2	5	6	6	258	28	9	0,09368444	48,5581395	53,5775194	0	1	0	355,00482