

Trabajo Fin de Grado

Ingeniería de las Tecnologías Industriales

Diseño de la distribución de quirófanos y consultas mediante algoritmos aproximados

Autor: María Eugenia Fernández Ramírez

Tutor: Víctor Fernández-Viagas Escudero

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

Sevilla, 2023



Trabajo Fin de Grado
Ingeniería de las Tecnologías Industriales

Diseño de la distribución de quirófanos y consultas mediante algoritmos aproximados

Autor:

María Eugenia Fernández Ramírez

Tutor:

Víctor Fernández-Viagas Escudero

Profesor titular

Dpto. de Organización Industrial y Gestión de Empresas I

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2023

Trabajo Fin de Grado: Diseño de la distribución de quirófanos y consultas mediante algoritmos aproximados

Autor: María Eugenia Fernández Ramírez

Tutor: Víctor Fernández-Viagas Escudero

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2023

El Secretario del Tribunal

Agradecimientos

A mi familia, a Ramón y a Eu..

El objetivo de este trabajo es abordar tanto la programación como la planificación de un hospital de referencia real, con el fin de minimizar una función objetivo. Esta función representa el retraso total en la atención de los pacientes con respecto a unos tiempos máximos de respuesta. Mediante la minimización de la función objetivo, se minimiza el retraso promedio de la atención hospitalaria y se promueve, en consecuencia, un aumento de la calidad de vida de la población.

Para programar la producción, es decir, establecer las fechas en las que cada paciente será atendido, los cirujanos que lo harán, y las consultas y quirófanos en que se realizará la consulta o intervención, se propone un Algoritmo Genético que, dados unos datos del problema, proporciona una secuencia de intervenciones satisfactoria.

Para planificar la producción, concretamente, hallar una adecuada distribución de quirófanos y consultas entre las unidades hospitalarias, se propone un Algoritmo Extendido, que realiza una búsqueda similar a la del algoritmo de mejora local.

Abstract

The objective of this project is to address both the programming and planning of a real reference hospital in order to minimize an objective function. This function represents the total delay in patient care compared to maximum response times. By minimizing the objective function, the average delay in hospital care is reduced, consequently promoting an improvement in the quality of life of the population.

To schedule production, that is, to establish the dates on which each patient will be treated, the surgeons who will perform the procedures, and the consultation rooms or operating rooms where the consultation or surgery will take place, a Genetic Algorithm is proposed. Given certain problem data, this algorithm provides a satisfactory sequence of interventions.

For production planning, specifically finding an appropriate distribution of operating rooms and consultation rooms among hospital units, an Extended Algorithm is proposed. This algorithm conducts a search similar to that of the local improvement algorithm.

Índice

Agradecimientos	vii
Resumen	ix
Abstract	xi
Índice	xii
Índice de Tablas	xiv
Índice de Figuras	xvi
1 Introducción	1
1.1 Organización de la producción	1
1.2 Los sistemas productivos y el entorno sanitario	2
2 Presentación del Problema	5
2.1 Descripción del entorno	5
2.2 Objetivo del problema	6
2.3 Restricciones del problema	6
2.4 Datos del problema	7
2.4.1 Recursos	7
2.4.2 Trabajos	7
2.5 Notación	8
3 Decodificación de la Solución	13
3.1 Cálculo de los tiempos de finalización	13
3.1.1 Etapa PAE	14
3.1.2 Etapa OR	15
3.1.3 Etapa POST	16
3.2 Cálculo de la función objetivo	17
4 Metaheurísticas y Algoritmos Genéticos	20
4.1 Métodos heurísticos o aproximados	20
4.2 Algoritmos Evolutivos y Algoritmos Genéticos	21
4.2.1 Población inicial	22
4.2.2 Grado de adaptación de los individuos o <i>fitness</i>	24
4.2.3 Estrategias de selección	24
4.2.4 Cruce	28
4.2.5 Mutación	31
4.2.6 Proceso de reemplazo	31
4.2.7 Criterios de terminación	32

5	Reglas de Despacho Elaboradas	33
5.1	<i>Regla de Despacho 1</i>	33
5.2	<i>Regla de Despacho 2</i>	33
5.3	<i>Regla de Despacho 3</i>	33
5.4	<i>Regla de Despacho 4</i>	34
5.5	<i>Regla de Despacho 5</i>	34
6	Algoritmos Propuestos	35
6.1	<i>Algoritmo Genético 1 o AG1</i>	35
6.1.1	Población inicial del AG1	35
6.1.2	Estrategia de selección del AG1	35
6.1.3	Cruce del AG1	36
6.1.4	Mutación del AG1	36
6.1.5	Proceso de reemplazo del AG1	36
6.1.6	Criterio de terminación del AG1	36
6.2	<i>Algoritmo Genético 2 o AG2</i>	36
6.2.1	Población inicial del AG2	36
6.2.2	Estrategia de selección del AG2	36
6.2.3	Cruce del AG2	36
6.2.4	Mutación del AG2	37
6.2.5	Proceso de reemplazo del AG2	37
6.2.6	Criterio de terminación del AG2	37
6.3	<i>Algoritmo Genético 3 o AG3</i>	37
6.3.1	Población inicial del AG3	37
6.3.2	Estrategia de selección del AG3	37
6.3.3	Cruce del AG3	37
6.3.4	Mutación del AG3	38
6.3.5	Proceso de reemplazo del AG3	38
6.3.6	Criterio de terminación del AG3	38
6.4	<i>Algoritmo Genético 4 o AG4</i>	38
6.4.1	Población inicial del AG4	38
6.4.2	Estrategia de selección del AG4	38
6.4.3	Cruce del AG4	38
6.4.4	Mutación del AG4	38
6.4.5	Proceso de reemplazo del AG4	39
6.4.6	Criterio de terminación del AG4	39
6.5	<i>Resumen de los Algoritmos Genéticos propuestos</i>	40
6.6	<i>Algoritmo Extendido</i>	40
7	Resultados Computacionales	47
7.1	<i>Generación del banco de pruebas</i>	47
7.1.1	Parámetros	47
7.1.2	Quirófanos y consultas virtuales	48
7.1.3	Generación de las intervenciones	48
7.1.4	Generación de los cirujanos	50
7.1.5	Instancias generadas	53
7.2	<i>Comparación de Algoritmos Genéticos</i>	53
7.2.1	Resultados obtenidos	53
8	Conclusiones	64
9	Líneas de Futura Investigación	65
	Referencias	66

ÍNDICE DE TABLAS

Tabla 2-1. Índices del problema.	8
Tabla 2-2. Variables del problema.	9
Tabla 2-3. Parámetros del problema.	10
Tabla 6-1. Resumen de los Algoritmos Genéticos propuestos.	40
Tabla 6-2. Número de soluciones iniciales aleatorias para cada conjunto de instancias.	42
Tabla 6-3. Número de soluciones iniciales para cada conjunto de instancias en cada fase.	43
Tabla 6-4. Resultados del Algoritmo Extendido.	44
Tabla 7-1. Instancias generadas.	53
Tabla 7-2. Comparación de los Algoritmos Genéticos.	55
Tabla 7-3. Comparación de los Reglas de Despacho.	59
Tabla 7-4. Resultados de una instancia.	61
Tabla 7-5. Comparación de los valores de la función objetivo obtenidos al partir de una solución inicial satisfactoria o de una solución inicial aleatoria.	61

ÍNDICE DE FIGURAS

Figura 1-1. Diagrama de Gantt de la producción de un sistema híbrido (Framinan et al., 2014).	3
Figura 1-2. Diagrama de Gantt de la producción de un sistema híbrido con dos tipos de máquinas.	4
Figura 3-1. Esquema del funcionamiento de la decodificación de la solución.	13
Figura 4-1. Exploración y explotación del espacio de búsqueda de una metaheurística.	21
Figura 4-2. Escapada de un óptimo local mediante una metaheurística.	21
Figura 4-3. Carácter iterativo del Algoritmo Genético.	22
Figura 4-4. A la izquierda, población inicial de un problema cuya solución es una cadena de 7 celdas azules o naranjas. A la derecha, población inicial de un problema cuya solución es una cadena de 7 caracteres que contienen los números del 1 al 7.	22
Figura 4-5. Ejemplo de las reglas de despacho EDD y SPT.	23
Figura 4-6. Ejemplo de vecinos por Intercambio Adyacente, Intercambio General e Inserción.	24
Figura 4-7. Ejemplo de la selección por torneo determinista de un padre en un problema de minimización.	25
Figura 4-8. Ejemplo de la selección por torneo probabilístico de un padre en un problema de minimización.	26
Figura 4-9. Ejemplo de selección por ruleta en un problema de maximización.	27
Figura 4-10. Ejemplo de selección por ruleta en un problema de minimización.	28
Figura 4-11. Arriba, ejemplo del operador de cruce monopunto. Abajo, ejemplo del operador de cruce multipunto.	29
Figura 4-12. Ejemplo inválido del operador de cruce monopunto.	29
Figura 4-13. Ejemplo del operador de cruce para representación de orden.	30
Figura 4-14. Ejemplo del operador de cruce para representación de orden con dos puntos de cruce.	30
Figura 4-15. A la izquierda, ejemplo de mutación por intercambio de pares. A la derecha, ejemplo de mutación por inserción.	31
Figura 6-1. Esquema del funcionamiento del Algoritmo Genético.	35
Figura 6-2. Ejemplo de solución inicial.	41
Figura 7-1. Primera parte de la Hoja de Pacientes de una instancia en formato .xlsx.	51
Figura 7-2. Segunda parte de la Hoja de Pacientes de una instancia en formato .xlsx.	52
Figura 7-3. Hoja de Cirujanos de una instancia en formato .xlsx.	52
Figura 7-4. Hoja de Datos de una instancia en formato .xlsx.	52
Figura 7-5. Excel con los resultados obtenidos.	54

1 INTRODUCCIÓN

Este trabajo afronta un problema vigente en el entorno sanitario y lo aborda como si se tratara de un problema de organización de la producción. De entrada, podría no verse claramente la relación existente entre un sistema de producción y un hospital. En este capítulo, se expondrán las similitudes que hacen comparables a ambos sistemas, para posteriormente aplicar fundamentadamente la teoría de los sistemas de producción al hospital.

1.1 Organización de la producción

Se entiende por organización de la producción el proceso de toma de decisiones destinadas a garantizar la distribución de productos con máxima calidad, mínimo coste, y mínimo tiempo de elaboración (en inglés, *lead time*) (Framinan et al., 2014). Dichas decisiones se clasifican en tres grupos según el impacto que tendrán en la empresa y el período de tiempo durante el cual se aplicarán.

- Decisiones estratégicas o de planificación: aquellas que sirven para establecer la capacidad de producción y de distribución de los recursos (Framinan et al., 2014). Estas decisiones, tomadas generalmente por los altos directivos de la empresa, tienen un enorme impacto en la misma, y solo pueden ser revisadas en un horizonte temporal de años, debido a su gran envergadura. La selección de productos a fabricar y de proveedores, y la localización y el diseño de la planta son algunas decisiones estratégicas.
- Decisiones tácticas: aquellas que sirven para repartir eficientemente los recursos limitados a medio plazo. La información clave para tomar estas decisiones puede estimarse gracias a los datos históricos y a la propia experiencia. El impacto de las decisiones tácticas sobre la empresa es medio, menor que el de las decisiones estratégicas y mayor que el de las operativas, y el horizonte temporal en el que pueden revisarse es de meses. Algunas decisiones tácticas son la elección de contratos con los proveedores, la mano de obra requerida y la distribución en planta (Morera et al., 2012).
- Decisiones operacionales: aquellas que sirven para repartir eficientemente los recursos limitados a corto plazo. Al igual que en el caso de las decisiones tácticas, la información utilizada para poder tomar estas decisiones se estima a partir de los datos históricos y de la experiencia. Sin embargo, es evidente que la calidad de la estimación disminuye con el nivel de detalle (Framinan et al., 2014). Por ejemplo, la estimación de las ventas anuales de un cierto modelo de coche será más acertada que la estimación de las ventas semanales de dicho modelo. Por consiguiente, la toma de decisiones tácticas podrá fundamentarse en la información estimada con mayor confianza que la toma de decisiones operacionales, ya que estas últimas programan la producción en el corto plazo. Las decisiones operativas tienen menor impacto que las estratégicas y las tácticas, y pueden ser revisadas en un horizonte temporal de días. Algunos ejemplos pueden ser la asignación de trabajos a los recursos disponibles y el inventario.

Se ha hecho referencia en el primer párrafo a la persecución de la máxima calidad, el mínimo coste y el mínimo tiempo de elaboración. Efectivamente, el decisor racional identifica y enumera las alternativas

posibles, analiza las consecuencias derivadas de cada una, y valora y compara dichas consecuencias (Morera et al., 2012), para elegir finalmente la alternativa óptima. Sin embargo, a la hora de gestionar la producción, es en muchos casos más eficiente aceptar una solución sencillamente satisfactoria, en lugar de perseguir aquella óptima. La razón es que la cantidad de alternativas es tan vasta, que hallar la óptima significaría invertir un tiempo desproporcionado para obtener un resultado con una mejora del objetivo poco significativa.

En el presente trabajo se presentará un método para abordar la toma de decisiones operacionales y tácticas de un hospital. Para comenzar, es necesario evidenciar las similitudes del hospital, y en general del entorno sanitario, con los sistemas productivos.

1.2 Los sistemas productivos y el entorno sanitario

Un sistema productivo es una estructura organizativa cuyo principal objetivo es la creación de valor. Para conseguirlo, necesita utilizar y combinar una serie de factores productivos que dan lugar a los productos necesarios para alcanzar el objetivo propuesto. En el sistema sanitario, la creación de valor se traduce en una mejora de la salud de la población o un aumento de la calidad de vida del paciente (Gorgemans & Urbina, 2017).

En un sistema productivo, entran los trabajos (en inglés, *jobs*) en forma de producto inicial y, tras ser sometidos a un proceso de transformación llevado a cabo por las máquinas (en inglés, *machines*), salen del sistema en forma de producto final. Más concretamente, en un entorno híbrido, que es el que más se asemeja al sistema de un hospital, cada trabajo debe pasar por un número de etapas o fases, y en cada una de ellas hay un número de máquinas que realizan la transformación.

En el sistema sanitario, los trabajos se identifican con los pacientes, que entran al sistema en forma de pacientes enfermos y salen de él como pacientes sanos. Las fases del entorno híbrido son las etapas por las que debe pasar el paciente en el proceso médico-quirúrgico: la consulta inicial, la intervención quirúrgica y la consulta postoperatoria. Las máquinas que intervienen en cada etapa para llevar a cabo el proceso de recuperación del paciente se identifican con los recursos necesarios para tratar al enfermo. Es evidente que entre estos recursos se encuentran los médicos y cirujanos, pero también las consultas y los quirófanos en los que se debe atender al paciente.

Los sistemas de producción se caracterizan también por la necesidad de planificación y programación. La programación de la producción implica la toma de decisiones operacionales con el objetivo de establecer en el corto plazo cuándo debe comenzar cada máquina a procesar cada trabajo (Pérez et al., 2018). La información cuantitativa proporcionada por un programa de la producción puede ser representada de manera gráfica, a través de un Diagrama de Gantt. De este modo, se facilita la visualización de la secuenciación, simultaneidad y relaciones de duración entre trabajos, así como el equilibrio presente en la utilización de los recursos. En la Figura 1-1 se puede ver un Diagrama de Gantt, que es resultado de la programación de un entorno híbrido, en el que cada fase está compuesta por máquinas paralelas. En él se especifica en qué momento (eje X) debe cada trabajo empezar a ser tratado por cada máquina (eje Y).

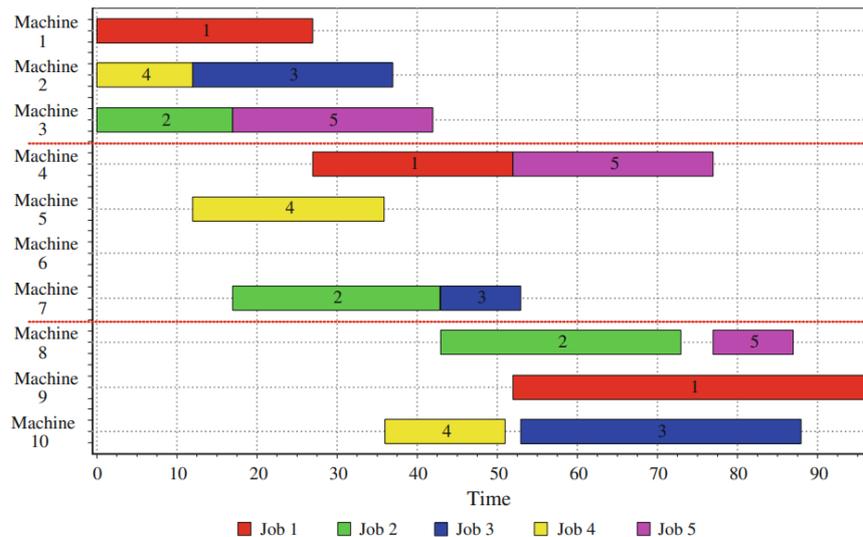


Figura 1-1. Diagrama de Gantt de la producción de un sistema híbrido (Framinan et al., 2014).

Del mismo modo, en el sistema sanitario es necesario programar la atención a los pacientes. La programación en este caso incluye establecer las fechas concretas en que cada paciente será atendido en cada etapa, así como el cirujano que llevará a cabo esta consulta o intervención, y la consulta médica o quirófano en que se realizará.

Tanto en un sistema productivo como en el sistema sanitario, la programación debe realizarse con una finalidad, que viene definida por una función objetivo a minimizar o maximizar. Si no existiera una función objetivo, la programación perdería importancia, ya que cualquier secuencia viable de trabajos sería igualmente válida.

En un sistema de producción, la función objetivo puede consistir, por ejemplo, en minimizar el tiempo total de flujo de trabajos (en inglés, *total flowtime*), es decir, el tiempo total que los trabajos están en el sistema, o en minimizar el tiempo máximo de terminación de los trabajos (en inglés, *makespan*), o sea, el instante en el que termina el último trabajo.

En un sistema sanitario, la función objetivo puede ser muy similar. Por ejemplo, minimizar el tiempo total de flujo de trabajos podría ser la función objetivo de un hospital cuya máxima fuera que cada paciente estuviera el menor tiempo posible enfermo. En cambio, minimizar el tiempo máximo de terminación de los trabajos podría ser la función objetivo de un hospital cuya finalidad es terminar de atender a los pacientes lo antes posible para poder dedicar el resto del tiempo a otros asuntos. En cualquier caso, la función objetivo debe estar definida para poder realizar una programación con el propósito de minimizarla (o maximizarla si fuera el caso).

También la planificación de un sistema de producción debe realizarse con el objetivo de optimizar el valor de la función objetivo. La planificación, a diferencia de la programación, consiste en la toma de decisiones tácticas y estratégicas a largo plazo para definir los recursos necesarios y la capacidad de producción. Si bien se entiende fácilmente que estrategias como invertir en maquinaria especializada pueden mejorar funciones objetivo relacionadas con los tiempos de producción, otras decisiones pueden ser más complicadas de tomar.

Por ejemplo, en un entorno híbrido en el que el número total de máquinas paralelas que puede haber en cada fase se conoce, y cada trabajo solo puede ser tratado por máquinas de un tipo concreto, es necesario decidir cuántas máquinas de cada tipo debe haber en cada fase. Esta decisión se toma al planificar la producción, y no al programarla, ya que es inalterable en el medio o largo plazo. Para decidir esta cuestión, se utiliza de nuevo la función objetivo, y se toma como mejor decisión aquella que la optimice. En la Figura 1-2 puede verse un ejemplo de la programación de la producción realizada una vez tomadas las decisiones tácticas y estratégicas que comprenden la planificación.

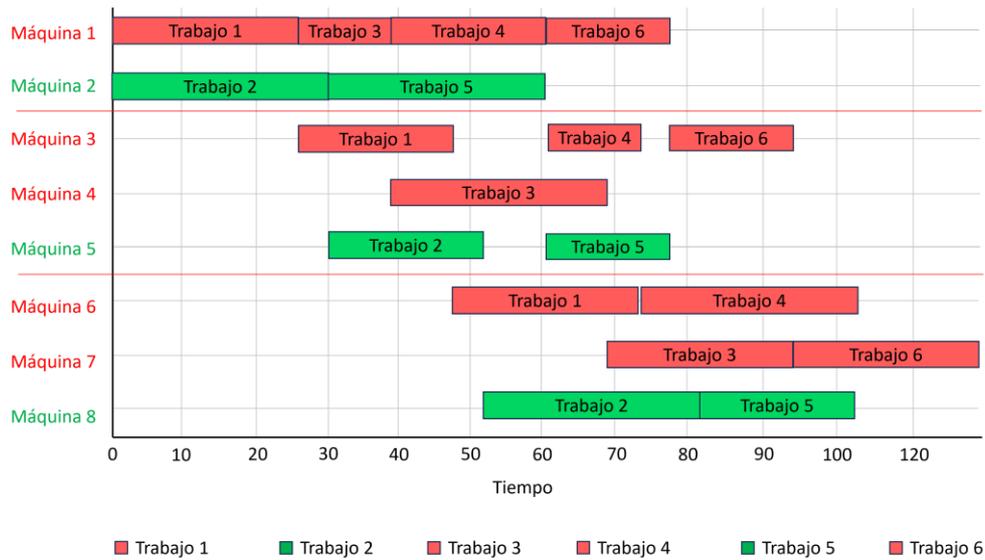


Figura 1-2. Diagrama de Gantt de la producción de un sistema híbrido con dos tipos de máquinas.

En el sistema sanitario también hay que planificar la atención a los pacientes en el largo plazo. Más allá de establecer una fecha, un médico y una consulta o quirófano para el enfermo, hay que tomar decisiones tácticas y estratégicas de carácter mucho más permanente. Por ejemplo, en línea con el ejemplo anterior de planificación de la producción, el hospital cuenta con un número de consultas y quirófanos concretos, y corresponde a la planificación asignar cada uno de ellos a las distintas unidades hospitalarias. Una distribución de quirófanos y consultas será mejor que otra si ofrece un mejor resultado de la función objetivo.

En conclusión, la necesidad común de programación y planificación para optimizar una función objetivo, la identificación de trabajos como pacientes y la equiparación de máquinas con cirujanos, consultas y quirófanos, permite tratar al sistema sanitario como un sistema de producción.

2 PRESENTACIÓN DEL PROBLEMA

El presente trabajo trata un problema actual de un hospital de referencia real cuyo nombre no se especifica por cuestiones de confidencialidad. El objetivo de este proyecto es presentar una metodología para facilitar la toma de decisiones operacionales y tácticas. Concretamente, se propondrán una serie de Algoritmos Genéticos que permitan secuenciar satisfactoriamente las intervenciones de una lista de espera (decisión operacional), así como un Algoritmo Extendido para encontrar una distribución aceptable de consultas y quirófanos a cada una de las unidades médicas del hospital (decisión táctica). Aplicando estos algoritmos en la toma de decisiones del hospital, se puede acabar con los desequilibrios en la adjudicación de recursos a las distintas unidades, con el fin de conseguir una mejor atención a todos los pacientes. Esta mejora en la atención se verá reflejada en la minimización de una función objetivo concreta, que se presentará posteriormente.

2.1 Descripción del entorno

Cada día, el hospital en cuestión atiende a pacientes con afecciones y síntomas muy diversos. Como la mayoría de los hospitales, posee distintas unidades, entendidas como secciones hospitalarias donde se concentran aparatos y personal especializados para el tratamiento de enfermos con dolencias de carácter similar. Por ejemplo, el hospital tiene una unidad de pediatría, una de cardiología, una de dermatología, etc., donde se trata a los pacientes que lo necesitan.

Cuando un paciente que afirma necesitar un médico entra al sistema del hospital, se le asigna una unidad responsable y se incorpora a una lista de espera que le conducirá a una sucesión de tres etapas: consulta PAE, intervención quirúrgica y consulta postoperatoria.

- Consulta PAE (Primera Atención Especializada): se trata de una consulta inicial realizada por un cirujano de la unidad para evaluar al paciente y confirmar la necesidad de intervención quirúrgica. Existe un tiempo máximo de respuesta genérico, desde la entrada al sistema hasta la ejecución del PAE, que no se debe superar. A lo largo del trabajo, se hará referencia a esta etapa como PAE.
- Intervención quirúrgica: llevada a cabo en un quirófano por un cirujano responsable y un cirujano asistente de la unidad. Existe un tiempo máximo de respuesta genérico entre el PAE y la intervención quirúrgica que no se debe superar. En posteriores alusiones a esta etapa, se utilizará el término OR, debido a la necesidad de llevar a cabo esta fase en un quirófano, que en inglés es *operating room*.
- Consulta postoperatoria: se trata de una consulta final en la que un cirujano de la unidad revisa el resultado de la intervención y comprueba que no hay imprevistos. Una vez finalizada esta consulta, el paciente está curado y abandona el sistema del hospital. Existe un tiempo máximo de respuesta genérico entre la intervención y el POST que no se debe superar. Para hacer referencia a esta etapa, se utilizará el término POST.

Por otro lado, pueden entrar al sistema pacientes que solo requieren una consulta genérica, es decir, una

consulta única que no va seguida de intervención quirúrgica. La programación y planificación de estas consultas queda fuera del alcance de este trabajo, sin embargo, es esencial tenerlas en cuenta, ya que afectan a la disponibilidad de los cirujanos. Para modelar las consecuencias de esta faceta del hospital, se reducirá la disponibilidad de los cirujanos.

2.2 Objetivo del problema

El objetivo de este trabajo es minimizar el retraso total (en inglés, *total tardiness*) del problema, que es la suma de los días de demora de todas las etapas de cada intervención en el horizonte de planificación. La minimización de los días totales de retraso se traduce en la minimización de la media aritmética de los retrasos.

Los días de demora se calculan a partir del tiempo máximo de respuesta de cada una de las etapas. Es importante recalcar que este dato no es una fecha de entrega, sino un intervalo de tiempo entre una etapa y la precedente que no se debe superar. Es decir, si el tiempo máximo de respuesta del OR es de 10 días, y el PAE se realizó en el día 4, el OR no debería realizarse después del día 14. A partir del día 15, el retraso de esa intervención y de esa etapa se hace positivo e incrementa con cada día que pase.

En el subcapítulo 3.2. *Cálculo de la función objetivo*, se presentará el cálculo matemático de la función objetivo, para el que hay que tener en cuenta factores como la etapa en la que la intervención entra al sistema y el horizonte de planificación. Como idea principal, el retraso de la etapa i de la intervención j es simplemente el número de días que dicha etapa se ha retrasado con respecto a su tiempo máximo de respuesta.

Una vez se haya definido claramente la función objetivo, esta se tratará de minimizar mediante la aplicación de algoritmos a la programación y la planificación del sistema.

2.3 Restricciones del problema

Hay algunas consideraciones genéricas que se han asumido a la hora de elaborar el modelo del hospital, para la correspondencia correcta de este con la realidad. Son las siguientes:

- Todo paciente tiene que pasar por el proceso médico-quirúrgico PAE-OR-POST.
- El tiempo de respuesta del PAE se inicia cuando el paciente entra al sistema.
- El tiempo de respuesta del OR se inicia en el instante de finalización del PAE.
- El tiempo de respuesta del POST se inicia en el instante de finalización del OR.
- Existen consultas genéricas a las que cada cirujano debe dedicar parte de su disponibilidad.
- Todas las consultas PAE tienen la misma duración y el mismo tiempo máximo de respuesta genérico.
- Todas las consultas POST tienen la misma duración y el mismo tiempo máximo de respuesta genérico.
- Todos los pacientes deben esperar el mismo tiempo mínimo genérico requerido entre la consulta PAE y la intervención OR.
- Todos los pacientes deben esperar el mismo tiempo mínimo genérico requerido entre la intervención OR y la consulta POST.
- No hay condicionante para decidir qué cirujano realizará las consultas PAE y POST.
- El OR lo realizan un cirujano responsable y un cirujano asistente que poseen el nivel de habilidad requerido.
- Cada cirujano puede estar asignado en un día a consultas o a quirófanos, pero no a ambos.
- Cuando un cirujano está asignado a consultas, puede realizar tanto consultas PAE como POST.
- Cada cirujano pertenece a una unidad.

- Cada intervención está asignada a una unidad, y solo pueden ser realizada por un cirujano de esta.
- Cada consulta y cada quirófano pertenece a una unidad, y en ellos solo se pueden realizar consultas e intervenciones de esta.

2.4 Datos del problema

Los datos del problema son toda aquella información que el hospital posee y que es necesaria para que el modelo funcione. Para analizarlos, a continuación se distingue entre los recursos y los trabajos.

2.4.1 Recursos

Los recursos son los medios disponibles para resolver una necesidad. En este problema, los recursos de los que se dispone para llevar a cabo las tres etapas (PAE, OR y POST) son los cirujanos, las consultas y los quirófanos. Los dos últimos son los que pueden asignarse a una unidad u otra para minimizar la función objetivo.

Cada cirujano pertenece a una unidad, por lo que solo podrá participar en el PAE, OR y POST de pacientes de esta. Es decir, un cirujano de la unidad de traumatología no puede atender a un paciente que ingresa en la unidad de oncología. Por otro lado, los cirujanos tienen un nivel de habilidad que les permite realizar unas intervenciones quirúrgicas u otras. Además, cada cirujano posee un calendario de disponibilidad diaria en el horizonte de planificación.

Por otro lado, existe un número determinado de consultas y quirófanos, *num_cons* y *num_quir*, cada uno de los cuales pertenece a una unidad médica y posee una disponibilidad diaria en el horizonte de planificación.

Por tanto, los datos de entrada de los recursos son:

Cirujano:

- Identificador.
- Unidad a la que pertenece.
- Nivel de habilidad que posee.
- Disponibilidad diaria en el horizonte de planificación.

Consulta:

- Identificador.
- Unidad a la que pertenece.
- Disponibilidad diaria en el horizonte de planificación.

Quirófano:

- Identificador.
- Unidad a la que pertenece.
- Disponibilidad diaria en el horizonte de planificación.

2.4.2 Trabajos

Los trabajos son aquellos productos que van a pasar por el proceso de producción. Entendiendo el proceso de producción como la secuencia médico-quirúrgica PAE-OR-POST, los trabajos son los pacientes. Puesto que a cada paciente se le asocia una intervención, se hablará a partir de ahora de intervenciones en lugar de pacientes. Por ejemplo, el paciente número 1 tiene asociada la intervención número 1, pero en este trabajo se hará referencia únicamente a las intervenciones.

Las intervenciones se asignan a una unidad. Para cada intervención, es necesario conocer en qué etapa se encuentra al momento de realizar la planificación. Algunas intervenciones entran a la primera etapa, PAE, pero

muchas otras ya han realizado dicha etapa y entran a OR o a POST. Además, cada intervención requiere un nivel de habilidad mínimo para el cirujano responsable del OR, y un nivel de habilidad mínimo para el cirujano asistente del OR. Por último, cada intervención tiene un tiempo de espera acumulado en la etapa en la que entra, una duración concreta del OR y un tiempo máximo de respuesta quirúrgica del OR. Los tiempos máximos de respuesta del PAE y del POST, así como sus duraciones, son comunes para todas las intervenciones. También son datos constantes para todas las intervenciones el tiempo mínimo requerido entre PAE y OR y el tiempo mínimo requerido entre OR y POST.

Por tanto, los datos de entrada de los trabajos son:

Intervención:

- Identificador.
- Unidad.
- Etapa inicial.
- Nivel de habilidad mínimo para el cirujano responsable del OR.
- Nivel de habilidad mínimo para el cirujano asistente del OR.
- Tiempo de espera acumulado.
- Tiempo de duración del PAE (común para todas las intervenciones).
- Tiempo de duración del OR.
- Tiempo de duración del POST (común para todas las intervenciones).
- Tiempo máximo de respuesta genérico del PAE (común para todas las intervenciones).
- Tiempo máximo de respuesta genérico del OR.
- Tiempo máximo de respuesta genérico del POST (común para todas las intervenciones).
- Tiempo mínimo genérico requerido entre el PAE y el OR (común para todas las intervenciones).
- Tiempo mínimo genérico requerido entre el OR y el POST (común para todas las intervenciones).

2.5 Notación

La notación utilizada a lo largo de este proyecto se define en las siguientes tablas.

Tabla 2-1. Índices del problema

Índices	
$j \in J$	Índice de la intervención.
$ciruj \in S$	Índice del cirujano.
$u \in n_{ud}$	Índice de la unidad.
$cons \in n_{cons}$	Índice de la consulta.
$quir \in n_{quir}$	Índice del quirófano.
$d \in H$	Índice del día.

$i = 0, 1, 2$	Índice de la etapa. Toma el valor 0 si es PAE, 1 si es OR, 2 si es POST.
---------------	--

Tabla 2-2. Variables del problema

Variables	
ud_j	Unidad de la intervención j .
$etapa_inic_j$	Etapa inicial de la intervención j . Toma el valor 0 si es PAE, 1 si es OR, 2 si es POST.
hab_resp_j	Nivel de habilidad mínimo para el cirujano responsable de la intervención j .
hab_asist_j	Nivel de habilidad mínimo para el cirujano asistente de la intervención j .
$tiempo_acum_j$	Tiempo acumulado en días de la intervención j en la lista de espera.
dur_or_j	Duración del OR de la intervención j en horas.
$tresp_or_j$	Tiempo máximo de respuesta genérico de la etapa OR de la intervención j . Hace referencia al número máximo de días que pueden pasar entre el PAE y el OR de la intervención.
$tresp_pae_rel_j$	Tiempo máximo de respuesta relativo en días de la etapa PAE de la intervención j . Se explica más detalladamente al final del capítulo.
$tresp_or_rel_j$	Tiempo máximo de respuesta relativo en días de la etapa OR de la intervención j . Se explica más detalladamente al final del capítulo.
$tresp_post_rel_j$	Tiempo máximo de respuesta relativo en días de la etapa POST de la intervención j . Se explica más detalladamente al final del capítulo.
$t_p_o_rel_j$	Tiempo mínimo relativo en días requerido entre PAE y OR para la intervención j . Se explica más detalladamente al final del capítulo.
$t_o_p_rel_j$	Tiempo mínimo relativo en días requerido entre OR y POST para la intervención j . Se explica más detalladamente al final del capítulo.
ud_{ciruj}	Unidad del cirujano $ciruj$.
hab_{ciruj}	Habilidad del cirujano $ciruj$.
$disp_{ciruj,d}$	Disponibilidad en horas del cirujano $ciruj$ el día d .
ud_{cons}	Unidad de la consulta $cons$.
$disp_{cons,d}$	Disponibilidad en horas de la consulta $cons$ el día d .
ud_{quir}	Unidad del quirófano $quir$.
$disp_{quir,d}$	Disponibilidad en horas del quirófano $quir$ el día d .

P_u	Número de pacientes de la unidad u .
C_u	Número de cirujanos de la unidad u .
$ciruj_cons_{u,d}$	Lista que contiene los cirujanos de la unidad u asignados a consultas el día d .
$ciruj_quir_{u,d}$	Lista que contiene los cirujanos de la unidad u asignados a quirófanos el día d .
$ct_{i,j}$	Tiempo de finalización de la etapa i de la intervención j . Es el día que se lleva a cabo dicha etapa.
$T_{i,j}$	Días de retraso de la etapa i de la intervención j .

Tabla 1-3. Parámetros del problema

Parámetros	
J	Número de intervenciones.
S	Número de cirujanos
n_{ud}	Número de unidades.
n_{cons}	Número de consultas.
n_{quir}	Número de quirófanos.
H	Número de días en el horizonte de planificación.
dur_{pae}	Duración en horas de una consulta PAE.
dur_{post}	Duración en horas de una consulta POST.
$tresp_{pae}$	Tiempo máximo de respuesta genérico de una consulta PAE. Hace referencia al número máximo de días que pueden pasar desde la entrada al sistema hasta la realización del PAE.
$tresp_{post}$	Tiempo máximo de respuesta genérico de una consulta POST. Hace referencia al número máximo de días que pueden pasar entre el OR y el POST.
t_{p-o}	Tiempo mínimo genérico requerido entre el PAE y el OR. Hace referencia al mínimo número de días que deben pasar entre el PAE y el OR.
t_{o-p}	Tiempo mínimo genérico requerido entre el OR y el POST. Hace referencia al mínimo número de días que deben pasar entre el OR y el POST.

Se puede observar que se ha especificado el carácter genérico o relativo de los tiempos máximos de respuesta y los tiempos mínimos requeridos presentados. Se habla de tiempo máximo de respuesta genérico para hacer referencia al tiempo máximo que puede pasar entre una etapa y la anterior. En el caso de la consulta PAE, el tiempo máximo de respuesta genérico se refiere al número máximo de días que puede pasar desde la entrada

del paciente al sistema hasta la ejecución de la consulta PAE; en el caso de la intervención quirúrgica OR, el tiempo máximo de respuesta genérico hace referencia al número máximo de días que pueden pasar desde la realización de la consulta PAE hasta la ejecución de la intervención quirúrgica OR; en el caso de la consulta POST, el tiempo máximo de respuesta genérico se refiere al número máximo de días que pueden pasar desde la realización del OR hasta la realización de la consulta POST.

Distinto es el concepto de tiempo máximo de respuesta relativo, que es el que se utilizará en los cálculos en posteriores capítulos. El tiempo máximo de respuesta relativo tiene en consideración el tiempo de espera acumulado de cada intervención y la etapa inicial de esta. Es decir:

- Si $etapa_inic_j = 0$
 - $t_resp_pae_rel_j = t_resp_pae - tiempo_acum_j$
 - $t_resp_or_rel_j = t_resp_or_j$
 - $t_resp_post_rel_j = t_resp_post$
- Si $etapa_inic_j = 1$
 - $t_resp_pae_rel_j = t_resp_pae$ (Este dato no se utilizará, ya que la etapa inicial es OR)
 - $t_resp_or_rel_j = t_resp_or_j - tiempo_acum_j$
 - $t_resp_post_rel_j = t_resp_post$
- Si $etapa_inic_j = 2$
 - $t_resp_pae_rel_j = t_resp_pae$ (Este dato no se utilizará, ya que la etapa inicial es POST)
 - $t_resp_or_rel_j = t_resp_or_j$ (Este dato no se utilizará, ya que la etapa inicial es POST)
 - $t_resp_post_rel_j = t_resp_post - tiempo_acum_j$

A continuación, se pone un ejemplo para ilustrar lo anterior. El tiempo máximo de respuesta genérico de todas las consultas POST es de 40 días. Hay una intervención que tiene un tiempo de espera acumulado para el POST de 35 días, por lo que su tiempo máximo de respuesta relativo es de 5 días ($40 - 35 = 5$). Sin embargo, otra intervención que tiene un tiempo de espera acumulado para el POST de tres días, tiene un tiempo máximo de respuesta relativo de 37 días ($40 - 3 = 37$). Aunque el tiempo máximo de respuesta genérico de todas las consultas POST es el mismo, el tiempo máximo de respuesta relativo es característico de cada intervención.

Lo mismo sucede con el tiempo mínimo requerido entre PAE y OR, y el tiempo mínimo requerido entre OR y POST.

- Si $etapa_inic_j = 0$
 - $t_p_o_rel_j = t_p_o$
 - $t_o_p_rel_j = t_o_p$
- Si $etapa_inic_j = 1$
 - $t_p_o_rel_j = \begin{cases} t_p_o - tiempo_acum_j & \text{si } t_p_o_rel_j \geq 0 \\ 0 & \text{e. c. c} \end{cases}$
 - $t_o_p_rel_j = t_o_p$
- Si $etapa_inic_j = 2$
 - $t_p_o_rel_j = t_p_o$ (Este dato no se utilizará, ya que la etapa inicial es POST)
 - $t_o_p_rel_j = \begin{cases} t_o_p - tiempo_acum_j & \text{si } t_o_p_rel_j \geq 0 \\ 0 & \text{e. c. c} \end{cases}$

A continuación, se pone un ejemplo para ilustrar lo anterior. El tiempo mínimo genérico requerido entre OR y POST es de 20 días. Hay una intervención que tiene un tiempo de espera acumulado para el POST de 15 días,

por lo que su tiempo mínimo relativo requerido entre OR y POST es de 5 días. ($20 - 15 = 5 > 0$). Esa intervención no podrá realizar el POST hasta que pasen 5 días más. Sin embargo, otra intervención que tiene un tiempo de espera acumulado para el POST de 30 días, tiene un tiempo mínimo relativo de 0 días ($20 - 30 = -10 < 0$). Aunque el tiempo mínimo genérico requerido entre OR y POST es el mismo, el tiempo mínimo relativo es característico de cada intervención.

Hay que recalcar que los tiempos relativos solo difieren de los genéricos cuando involucran a la etapa inicial de la intervención. Por ejemplo, para una intervención que entra en PAE y que lleva 20 días en lista de espera, su tiempo máximo de respuesta genérico del OR es igual al relativo, y ambos hacen referencia al número máximo de días que pueden pasar entre PAE y OR.

3 DECODIFICACIÓN DE LA SOLUCIÓN

Para resolver el problema que ocupa a este trabajo, es decir, programar las intervenciones y planificar la distribución de quirófanos y consultas de forma que se minimice la función objetivo, es necesario comenzar por una decodificación de la solución. En otras palabras, el primer paso es realizar un modelo que sea capaz de calcular el valor de dicha función objetivo para unos datos del problema y una solución dados.

Conocidos ya los datos del problema, que se han expuesto en el apartado 2.4. *Datos del problema*, es necesario recalcar que, para esta parte del trabajo, se debe partir también de una solución, es decir, una determinada secuencia de intervenciones y una distribución concreta de quirófanos y consultas a las distintas unidades del hospital. Por ejemplo, en una dimensión reducida del problema en la que solo existen 5 intervenciones, 2 unidades hospitalarias, 3 consultas y 2 quirófanos, una solución podría ser la secuencia de intervenciones 1, 2, 3, 4, 5 y la distribución de 2 consultas y 1 quirófano a la primera unidad, y 1 consulta y 1 quirófano a la segunda. Con estos datos de entrada, y disponiendo de la decodificación de la solución apropiada, se obtiene el valor de la función objetivo.



Figura 0-1. Esquema del funcionamiento de la decodificación de la solución.

En este capítulo se explicará detalladamente cómo obtener el valor de la función objetivo a partir de los datos de entrada.

3.1 Cálculo de los tiempos de finalización

Para calcular los tiempos de finalización (en inglés, *completion times*) de cada etapa de cada intervención, es necesario considerar una a una las intervenciones de la secuencia introducida como dato de entrada. Cada una de ellas será analizada con el procedimiento descrito a continuación.

Para empezar, se identifica la etapa en la que se encuentra la intervención, ya que la asignación de cirujanos y consultas o quirófanos se basa en distintos criterios según se encuentre en PAE, OR o POST.

3.1.1 Etapa PAE

En primer lugar, $ud_j = \text{unidad}$ y $x = 0$.

En el caso de que la intervención entre a PAE, se elige el primer día a partir de x en el que alguna de las consultas asignadas a la unidad en cuestión tiene disponibilidad suficiente para albergar el PAE, que tiene una duración definida. Es decir, el primer día en el que se cumple que:

$$ud_{cons} = \text{unidad} \quad \text{y} \quad disp_{cons,d} \geq dur_{pae} \quad \forall cons$$

Cuando esto se cumple, $dia = d$ y $consulta = cons$

Una vez elegido el día dia y la consulta $consulta$, se comprueba cuáles de los cirujanos asignados a esa unidad están disponibles ese día. La disponibilidad implica que el cirujano no esté asignado a quirófanos en esa jornada (es decir, que esté asignado a consultas o que todavía no haya sido asignado a nada) y que tenga tiempo suficiente para realizar el PAE.

$$ciruj_disp_{unidad,dia} \subset ciruj \quad \text{si} \quad \begin{cases} ud_{ciruj} = \text{unidad} \\ ciruj_quir_{unidad,dia} \not\subset ciruj \\ disp_{ciruj,dia} \geq dur_{pae} \end{cases} \quad \forall ciruj$$

De entre los cirujanos disponibles, se selecciona a aquellos que están asignados a consultas y se elige al que posee menor habilidad.

$$ciruano_elegido = ciruj \quad \text{si} \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad \forall ciruj, ciruj' \subset ciruj_cons_{unidad,dia}, \quad ciruj' \neq ciruj$$

Si no hubiera cirujanos asignados a consultas de entre los disponibles, se selecciona al cirujano de menor habilidad y se le asigna a consultas.

$$ciruano = ciruj \quad \text{si} \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad ciruj' \neq ciruj$$

Añadir $ciruj$ a $ciruj_cons_{unidad,dia}$

Si no hubiera cirujanos disponibles en absoluto, es necesario buscar el siguiente día en el que hay consultas disponibles. Es decir, se vuelve al primer paso con $x = dia + 1$, siempre y cuando $x < H$.

Una vez elegidos la consulta y el cirujano, se reduce la disponibilidad de ambos, y se establece el día elegido como el tiempo de finalización de la etapa PAE de la intervención.

$$disp_{cons_{consulta,dia}} = disp_{cons_{consulta,dia}} - dur_{pae}$$

$$disp_{ciruj_{cirujano,dia}} = disp_{ciruj_{cirujano,dia}} - dur_{pae}$$

$$ct_{0,j} = dia$$

A continuación, se envía la intervención a la siguiente etapa, el OR.

Si no se pudiera realizar el PAE ningún día del horizonte de planificación, se devuelve 'No time' para la etapa actual PAE y para las siguientes (OR y POST).

$$ct_{0,j} = 'No time'$$

$$ct_{1,j} = 'No\ time'$$

$$ct_{2,j} = 'No\ time'$$

3.1.2 Etapa OR

En primer lugar, $ud_j = unidad$.

Si la intervención entra a OR, hay que tener en cuenta el tiempo mínimo requerido entre la consulta PAE y la intervención quirúrgica, durante el cual no se puede realizar la intervención.

$$x = t_p_o_rel_j$$

Se elige, a partir de la conclusión de este tiempo x , el primer día en el que alguno de los quirófanos asignados a la unidad en cuestión tiene disponibilidad suficiente para albergar el OR, que tiene una duración definida. Es decir, el primer día en el que se cumple que:

$$ud_{quir} = unidad \quad y \quad disp_{quir,d} \geq dur_or_j \quad \forall quir$$

Cuando esto se cumple, $dia = d$ y $quirofano = quir$

Una vez elegido el día dia y el quirófano $quirofano$, se comprueba cuáles de los cirujanos asignados a esa unidad están disponibles ese día para ser asistentes en la intervención quirúrgica. Estos son los cirujanos que, teniendo un nivel de habilidad suficiente para ser asistentes en la intervención, no están asignados a consultas durante el día elegido y tienen suficiente tiempo para realizar el OR.

$$ciruj_disp_{unidad,dia} \subset ciruj \quad si \quad \begin{cases} ud_{ciruj} = unidad \\ ciruj_cons_{unidad,dia} \not\subset ciruj \\ hab_{ciruj} \geq hab_asist_j \\ disp_{ciruj,dia} \geq dur_or \end{cases} \quad \forall ciruj$$

De entre ellos, se selecciona como cirujano asistente al que menor habilidad tiene.

$$cirujano_asistente = ciruj \quad si \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad ciruj' \neq ciruj$$

Para elegir al cirujano responsable de la intervención, se parte de la lista de cirujanos disponibles para ser asistentes, eliminando de la misma al cirujano asistente y a aquellos cirujanos que no tienen habilidad suficiente para ser responsables.

$$Eliminar \quad cirujano_asistente \quad de \quad ciruj_disp_{unidad,dia}$$

$$Eliminar \quad ciruj \quad de \quad ciruj_disp_{unidad,dia} \quad si \quad hab_{ciruj} < hab_resp_j$$

De la lista resultante, se comprueba si hay cirujanos asignados a quirófanos, en cuyo caso se selecciona a aquel con menor habilidad.

$$cirujano_responsable = ciruj \quad si \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad \forall ciruj, ciruj' \subset ciruj_quir_{unidad,dia}, \quad ciruj' \neq ciruj$$

Si no hubiera cirujanos asignados a quirófanos, se selecciona al cirujano de menor habilidad y se le asigna a quirófanos durante ese día. También se asigna a quirófanos al cirujanos asistente si no lo estaba ya.

$$cirujano_responsable = ciruj \text{ si } hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad ciruj' \neq ciruj$$

Añadir *cirujano_responsable* a *ciruj_quir*_{unidad,dia}

Añadir *cirujano_asistente* a *ciruj_quir*_{unidad,dia} si *ciruj_quir*_{unidad,dia} \notin *cirujano_asistente*

Si no hubiera cirujanos disponibles, se busca el siguiente día en el que hay quirófanos disponibles. Es decir, se vuelve al primer paso con $x = dia + 1$, siempre y cuando $x < H$.

Una vez elegidos el quirófano y los dos cirujanos, se reduce su disponibilidad y se establece el día elegido como el tiempo de finalización de la etapa OR de la intervención.

$$disp_quir_{quirofano,dia} = disp_quir_{quirofano,dia} - dur_or_j$$

$$disp_ciruj_{cirujano_asistente,dia} = disp_ciruj_{cirujano_asistente,dia} - dur_or_j$$

$$disp_ciruj_{cirujano_responsable,dia} = disp_ciruj_{cirujano_responsable,dia} - dur_or_j$$

$$ct_{1,j} = dia$$

A continuación, se envía a la siguiente etapa, el POST.

Si no fuera posible realizar el OR ningún día del horizonte de planificación, se devuelve 'No time' para dicha etapa y para la siguiente (POST).

$$ct_{1,j} = 'No time'$$

$$ct_{2,j} = 'No time'$$

3.1.3 Etapa POST

En primer lugar, $ud_j = unidad$.

Si la intervención entra en POST, hay que tener en cuenta el tiempo mínimo requerido entre la intervención quirúrgica y la consulta POST, durante el cual no se puede realizar la consulta.

$$x = t_o_p_rel_j$$

Se elige, a partir de la conclusión de este tiempo x , el primer día en el que alguna de las consultas asignadas a la unidad en cuestión tiene disponibilidad suficiente para albergar el POST.

$$ud_{cons} = unidad \text{ y } disp_{cons,d} \geq dur_or \quad \forall cons$$

Cuando esto se cumple, $dia = d$ y $consulta = cons$

Una vez elegido el día *dia* y la consulta *consulta*, se comprueba cuáles de los cirujanos asignados a esa unidad están disponibles ese día. La disponibilidad implica que el cirujano no esté asignado a quirófanos en esa jornada (es decir, que esté asignado a consultas o que todavía no haya sido asignado a nada) y que tenga tiempo suficiente para realizar el POST.

$$ciruj_disp_{unidad,dia} \subset ciruj \quad si \quad \begin{cases} ud_{ciruj} = unidad \\ ciruj_quir_{unidad,dia} \notin ciruj \\ disp_{ciruj,dia} \geq dur_post \end{cases} \quad \forall ciruj$$

De entre los cirujanos disponibles, se selecciona a aquellos que están asignados a consultas y se elige al que posee menor habilidad.

$$ciruano_elegido = ciruj \quad si \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad \forall ciruj, ciruj' \subset ciruj_cons_{unidad,dia}, \quad ciruj' \neq ciruj$$

Si no hubiera cirujanos asignados a consultas de entre los disponibles, se selecciona al cirujano de menor habilidad y se le asigna a consultas.

$$ciruano = ciruj \quad si \quad hab_{ciruj} \leq hab_{ciruj'}$$

$$\forall ciruj, ciruj' \subset ciruj_disp_{unidad,dia}, \quad ciruj' \neq ciruj$$

$$\text{Añadir } ciruj \text{ a } ciruj_cons_{unidad,dia}$$

Si no hubiera cirujanos disponibles en absoluto, es necesario buscar el siguiente día en el que hay consultas disponibles. Es decir, se vuelve al primer paso con $x = dia + 1$, siempre y cuando $x < H$.

Una vez elegidos la consulta y el cirujano, se reduce la disponibilidad de ambos, y se establece el día elegido como el tiempo de finalización de la etapa POST de la intervención.

$$disp_cons_{consulta,dia} = disp_cons_{consulta,dia} - dur_post$$

$$disp_ciruj_{cirujano,dia} = disp_ciruj_{cirujano,dia} - dur_post$$

$$ct_{2,j} = dia$$

Si no se pudiera realizar el POST ningún día, se devuelve 'No time' para dicha etapa.

$$ct_{2,j} = 'No time'$$

Por último, es necesario recalcar que, si una intervención tiene como etapa inicial el OR o el POST, las etapas anteriores que no se contemplan en la planificación en cuestión, tienen como tiempo de finalización '-'.

$$ct_{0,j} = ' - ' \quad si \quad etapa_inic_j = 1$$

$$ct_{0,j} = ct_{1,j} = ' - ' \quad si \quad etapa_inic_j = 2$$

3.2 Cálculo de la función objetivo

La función objetivo que se pretende minimizar es el sumatorio del retraso de todas las etapas de todas las intervenciones. Es una función objetivo sencilla, donde la única complejidad reside en la existencia de distintas maneras de calcular los días de retraso, en función del tiempo de finalización y de la etapa inicial de la intervención.

Para simplificar la explicación de este cálculo, va a hacerse una pequeña modificación con respecto a la notación anterior. Los tiempos máximos relativos de respuesta se van a recoger en una única variable $tresp_rel_{i,j}$ que tiene dos índices, uno para la etapa y otra para la intervención, de forma que:

$$tresp_rel_{0,j} = tresp_pae_rel_j$$

$$tresp_rel_{1,j} = tresp_or_rel_j$$

$$tresp_rel_{2,j} = tresp_post_rel_j$$

La función objetivo queda:

$$FO = \sum_{j=1}^J \sum_{i=0}^2 T_{i,j}$$

$$1) T_{i,j} = \text{máx} \{0, ct_{i,j} - tresp_rel_{i,j}\}$$

si $i = etapa_inic_j$ y $ct_{i,j} \subset H$

$$2) T_{i,j} = \text{máx} \{0, (ct_{i,j} - ct_{i-1,j}) - tresp_rel_{i,j}\}$$

si $i \neq etapa_inic_j$ y $ct_{i,j} \subset H$

$$3) T_{i,j} = \text{máx} \{0, H - tresp_rel_{i,j}\}$$

si $i = etapa_inic_j$ y $ct_{i,j} = 'No\ time'$

$$4) T_{i,j} = \text{máx} \{0, (H - ct_{i-1,j}) - tresp_rel_{i,j}\}$$

si $i \neq etapa_inic_j$, $ct_{i,j} = 'No\ time'$ y $ct_{i-1,j} \subset H$

$$5) T_{ij} = 0$$

si $ct_{ij}, ct_{i-1j} = 'No\ time'$ o $ct_{i,j} = '-'$

- 1) Se calcula el retraso de la etapa inicial de una intervención ($i = etapa_inic_j$), sabiendo que el tiempo de finalización de esta etapa de la intervención está dentro del horizonte temporal de planificación ($ct_{i,j} \subset H$).

El retraso es el tiempo de finalización menos el tiempo máximo de respuesta relativo si este resultado es positivo, o cero si el resultado anterior es negativo.

Por ejemplo, si una intervención cuya etapa inicial es OR tiene como tiempo máximo de respuesta relativo del OR 10 días, y su tiempo de finalización es el día 13, el retraso de la etapa OR de la intervención es igual a 3 ($13 - 10 = 3 > 0$). Si en cambio el tiempo de finalización es el día 5, el retraso de la etapa en cuestión es cero ($5 - 10 = -5 < 0$).

- 2) Se calcula el retraso de una etapa distinta a la inicial de la intervención ($i \neq etapa_inic_j$), sabiendo que tiempo de finalización de esta etapa de la intervención está dentro del horizonte temporal de planificación ($ct_{i,j} \subset H$).

El retraso es la diferencia entre, por un lado, el número de días que han pasado entre la realización de la etapa anterior hasta la realización de la etapa en cuestión, y, por otro lado, el tiempo máximo de respuesta genérico de la etapa en cuestión (que coincide con el genérico). Este es el retraso si el resultado de esa operación es positivo. Si el resultado fuera negativo, el retraso es cero.

Por ejemplo, una intervención entra en OR, y se calcula un tiempo de finalización del OR de 3 y un tiempo de finalización del POST de 25. El número de días entre OR y POST es de 22 ($25 - 3 = 22$).

Sabiendo que el máximo tiempo de respuesta relativo del POST es de 20, el retraso del POST es de 2 ($22 - 20 = 2$).

- 3) Se calcula el retraso de la etapa inicial de una intervención ($i = etapa_inic_j$), sabiendo que el tiempo de finalización de esta etapa de la intervención está fuera del horizonte temporal de planificación ($ct_{i,j} = 'No\ time'$).

El retraso es el número de días en el horizonte temporal de planificación menos el tiempo máximo de respuesta relativo de la etapa si el resultado es positivo, y cero si el resultado es negativo.

Por ejemplo, si una intervención entra en POST con un tiempo máximo de respuesta relativo de 10, y al planificar los tiempos de finalización se descubre que no da tiempo a realizar el POST en el horizonte temporal de 14 días, se considera que el retraso del POST es igual a 4 ($14 - 10 = 4 > 0$). En cambio, si el tiempo máximo de respuesta relativo fuera de 20 días, el retraso del POST sería igual a cero ($14 - 20 = -6 < 0$). Consiste simplemente en imaginar que $ct_{i,j} = H$.

- 4) Se calcula el retraso de una etapa i distinta a la inicial de la intervención ($i \neq etapa_inic_j$), sabiendo que el tiempo de finalización de la etapa en cuestión está fuera del horizonte temporal de planificación ($ct_{i,j} = 'No\ time'$), y que el tiempo de finalización de la etapa anterior $i - 1$ está dentro del horizonte temporal de planificación ($ct_{i-1,j} \subset H$).

El retraso es la diferencia entre, por un lado, el número de días desde el tiempo de finalización de la etapa anterior hasta el final del horizonte temporal, y, por otro lado, el tiempo máximo de respuesta relativo de la etapa en cuestión (que coincide con el genérico). Este es el retraso si el resultado de esa operación es positivo. En caso de que sea negativo, el retraso es cero.

Por ejemplo, una intervención entra en PAE, y tiene un tiempo de finalización del PAE de 3, y un tiempo de finalización del OR indefinido porque no da tiempo a realizarlo en el horizonte de planificación de 14 días. Si el tiempo máximo de respuesta genérico del OR es de 10, el retraso del OR es 1 ($(14 - 3) - 10 = 1 > 0$). Consiste simplemente en imaginar que $ct_{i,j} = H$.

- 5) Se calcula el retraso de una etapa i sabiendo que el tiempo de finalización de la etapa en cuestión está fuera del horizonte temporal de planificación ($ct_{i,j} = 'No\ time'$), y que el tiempo de finalización de la etapa anterior $i - 1$ también está fuera del horizonte de planificación ($ct_{i-1,j} = 'No\ time'$).

El retraso es cero.

Este también es el retraso de las etapas anteriores a la etapa inicial de la intervención ($ct_{i-1,j} = ' - '$).

Por ejemplo, una intervención entra en PAE, y tiene un tiempo de finalización del PAE fuera del horizonte temporal y, evidentemente, un tiempo de finalización del OR y del POST fuera del horizonte temporal también. El retraso del PAE será el correspondiente, y el retraso de la etapa OR y POST será 0.

4 METAHEURÍSTICAS Y ALGORITMOS GENÉTICOS

Mediante la decodificación de la solución, se obtiene el valor de la función objetivo para unos datos del problema y una solución dados, entendiendo como solución la secuencia de intervenciones y la distribución de quirófanos y consultas en las unidades hospitalarias. A continuación, es necesario aplicar un método heurístico para, a partir de unos datos del problema y una distribución de quirófanos y consultas, es decir, una solución inicial, dados, hallar una secuencia de intervenciones satisfactoria. Antes de presentar los métodos heurísticos propuestos, se va a introducir los conceptos de heurística, metaheurística y, más detalladamente, el funcionamiento de un Algoritmo Genético.

4.1 Métodos heurísticos o aproximados

Se considera heurístico a un procedimiento para el que se tiene un alto grado de confianza en que encuentre soluciones de alta calidad con un coste computacional razonable (Melián et al., 2003). A diferencia de los métodos exactos, mediante los cuales se obtienen resultados óptimos, los heurísticos aportan soluciones satisfactorias.

Los algoritmos heurísticos o aproximados ofrecen buenas soluciones en un tiempo razonable a problemas de difícil solución, que suelen requerir agrupamientos, ordenaciones o asignaciones de un conjunto discreto de objetos que satisfagan ciertas restricciones (Herrera, 2017). Para estos problemas de gran complejidad computacional, denominados Problemas NP (De Antonio Suárez, 2013), los algoritmos exactos, como el Branch and Bound, la programación matemática, la programación dinámica o los algoritmos constructivos exactos (Pérez et al., 2018), son ineficientes o imposibles de aplicar.

Entre los métodos heurísticos o aproximados más frecuentemente utilizados, se encuentran las heurísticas constructivas, las heurísticas de búsqueda o mejora local y las metaheurísticas. Estas últimas han sido utilizadas en el presente trabajo, pero a continuación se expondrá brevemente las principales características de las otras dos heurísticas.

Las heurísticas constructivas, como su propio nombre indica, construyen una solución en un número finito de pasos, añadiendo trabajos sin secuenciar en secuencias parciales (Pérez et al., 2022). Son métodos deterministas que se basan en la mejor elección en cada iteración (De Antonio Suárez, 2013). Por otro lado, las heurísticas de búsqueda o mejora local comienzan con una solución del problema y la mejoran progresivamente, de modo que, en cada paso, se realiza un movimiento de una solución a otra mejor (De Antonio Suárez, 2013).

Por último, las metaheurísticas son algoritmos aproximados de propósito general, con frecuencia de carácter iterativo, que guían una heurística subordinada mediante un balance adecuado entre la exploración (búsqueda en el espacio actual) y explotación (búsqueda en regiones distantes del espacio) (Herrera, 2017). Estos algoritmos tienen gran éxito en la práctica, en aquellas situaciones en las que no se requiere una solución óptima, sino que basta con una de buena calidad.

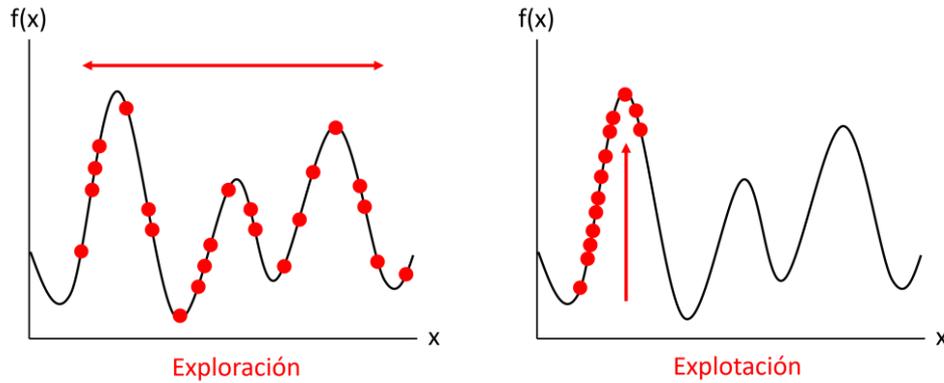


Figura 4-1. Exploración y explotación del espacio de búsqueda de una metaheurística.

Algunas metaheurísticas ampliamente difundidas son el método de búsqueda descendente simple, el Recocido Simulado (en inglés, *Simulated Annealing*), la Búsqueda Iterada Codiciosa (en inglés, *Iterated Greedy*) y el Algoritmo Genético (Pérez et al., 2022). Este último ha sido el elegido para llevar a cabo el presente trabajo. A continuación, se presenta el concepto de Algoritmo Genético y sus principales características.

4.2 Algoritmos Evolutivos y Algoritmos Genéticos

Los Algoritmos Genéticos son un tipo de Algoritmo Evolutivo muy popular. Los Algoritmos Evolutivos son una técnica de resolución de problemas de búsqueda y optimización inspirada en la teoría de la evolución de las especies y la selección natural (Araujo & Cervigón, 2009). Estos algoritmos combinan la búsqueda aleatoria (contribuyendo a la exploración) con la búsqueda dirigida según la selección de los individuos más adaptados (contribuyendo a la explotación).

La búsqueda aleatoria característica de los Algoritmos Evolutivos los hace capaces de escapar de óptimos locales, a diferencia de lo que sucede en el clásico método de escalada. Aplicando esta observación a la Figura 4-2, y considerando que la función objetivo es de maximización, si se parte del punto p y se realiza una técnica de escalada, solo se podrá llegar al punto a , y se dará esta solución por buena. Una vez alcanzado el punto a , es imposible escapar de dicho máximo local. En cambio, si se utiliza un Algoritmo Evolutivo, la búsqueda aleatoria puede conducir a otro punto q que, mediante la búsqueda dirigida por selección, lleve al b .

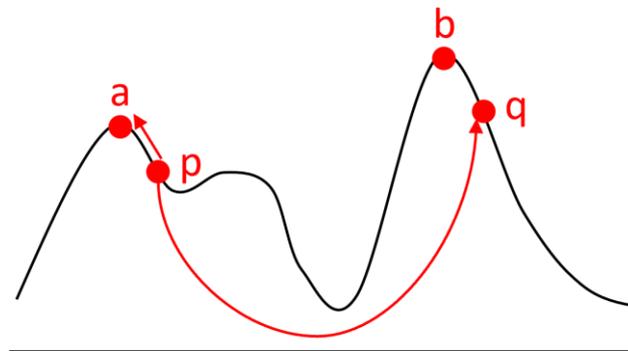


Figura 4-2. Escapada de un óptimo local mediante una metaheurística.

Los Algoritmos Evolutivos se basan en las premisas del modelo de evolución natural propuesto por Charles Darwin en 1859. Según la teoría de Darwin, la evolución de las especies se debe al principio de selección

natural, que favorece la supervivencia y multiplicación de aquellas especies que están mejor adaptadas a las condiciones de su entorno. Además, Darwin señaló como fundamental para la evolución el papel de las mutaciones, o pequeñas variaciones entre padres e hijos (Araujo & Cervigón, 2009).

Como ya se ha comentado, el Algoritmo Genético es uno de los Algoritmos Evolutivos más conocidos y utilizados. A continuación, se exponen las claves para comprender su metodología y funcionamiento.

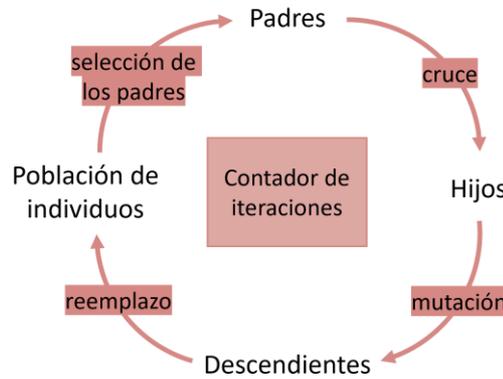


Figura 4-3. Carácter iterativo del Algoritmo Genético.

4.2.1 Población inicial

La población inicial consiste en un conjunto de individuos que son candidatos a ser solución del problema. Es necesario definir previamente la representación de los individuos, ya que, dependiendo del problema y de la solución buscada, los individuos pueden representarse como cadenas binarias, enteras o de número reales.

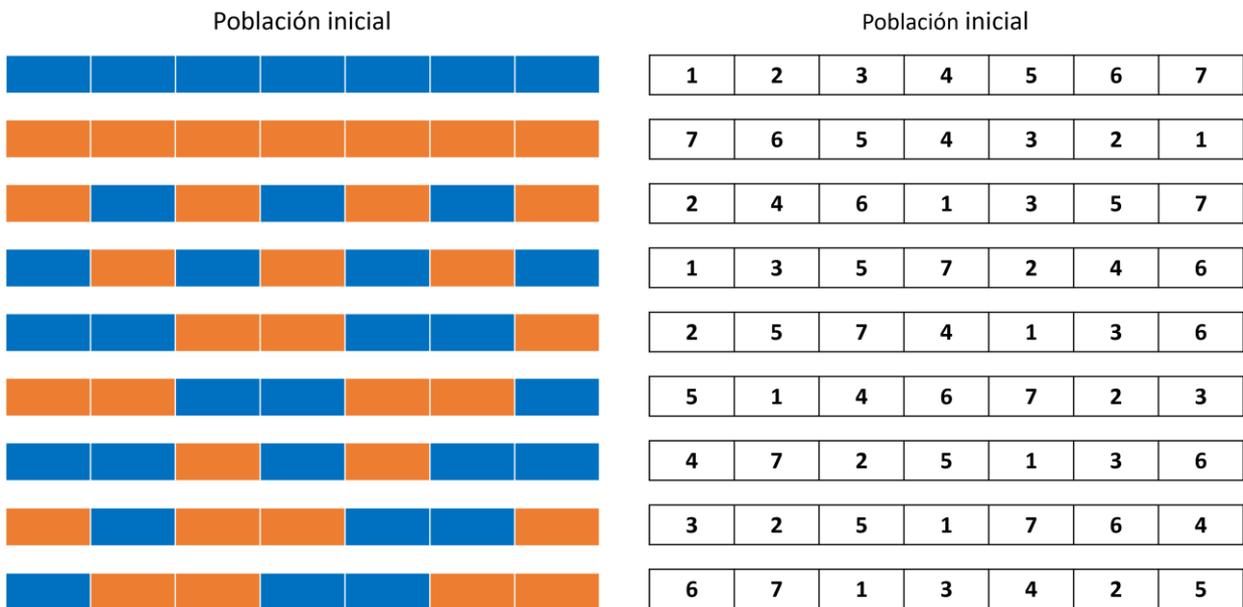


Figura 4-4. A la izquierda, población inicial de un problema cuya solución es una cadena de 7 celdas azules o naranjas. A la derecha, población inicial de un problema cuya solución es una cadena de 7 caracteres que contienen los números del 1 al 7.

El tamaño de la población inicial debe ser lo suficientemente grande para proporcionar un rango de exploración adecuado, y lo suficientemente pequeño para no entorpecer el algoritmo provocando tiempos de ejecución demasiado extensos. La mejor manera de encontrar el tamaño adecuado para un problema concreto es ejecutar el algoritmo y, tras valorar la relación entre los resultados obtenidos y los tiempos empleados, decidir aumentar o reducir el tamaño de la población inicial, hasta llegar a uno apropiado. En general, la población no debe ser demasiado numerosa. Poblaciones iniciales con cientos de individuos quedan absolutamente descartadas.

Los individuos de la población inicial pueden generarse de diversas maneras: de forma aleatoria, mediante reglas de despacho, con heurísticas constructivas o haciendo uso de las vecindades (Pérez et al., 2022). Puesto que las reglas de despacho y las vecindades no han sido explicadas anteriormente, y han intervenido en el Algoritmo Genético desarrollado en el presente trabajo, se explican a continuación.

Las reglas de despacho son reglas que permiten clasificar y ordenar los trabajos de diferente forma, teniendo en cuenta la prioridad establecida (Sipper & Bulfin, 1998). Algunas reglas de despacho muy típicas son la “Regla de la Fecha de Entrega más Temprana” o EDD (en inglés, *Earliest Due Date*), por la que los trabajos se ordenan de menor a mayor fecha de entrega, la “Regla del Tiempo de Proceso más Corto” o SPT (en inglés, *Shortest Processing Time*), por la que los trabajos se ordenan de menor a mayor tiempo de proceso, la “Regla de Primeras Entradas, Primeras Salidas” o FIFO (en inglés, *First In, First Out*), por la que los trabajos se ordenan según el orden de llegada, y muchas otras más.

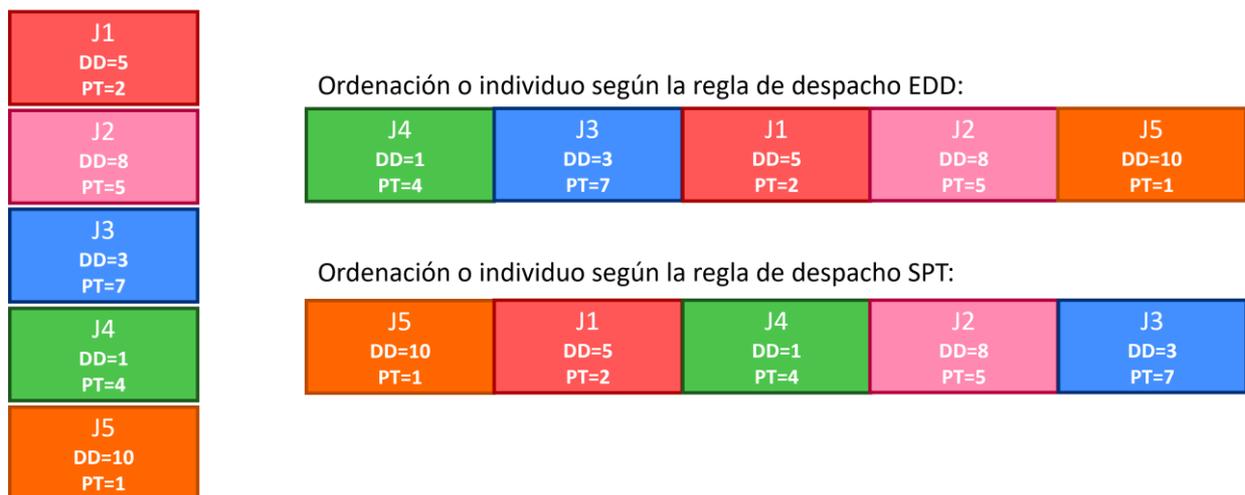


Figura 4-5. Ejemplo de las reglas de despacho EDD y SPT.

Por otro lado, se entiende como vecindad de una solución inicial al conjunto de todas las posibles soluciones que se pueden alcanzar mediante un pequeño movimiento local en la misma (Pérez et al., 2022). Los vecinos de una solución inicial pueden obtenerse mediante el Intercambio Adyacente (en inglés, *Adjacent Swap*), el Intercambio General (en inglés, *General Swap*) o la Inserción (en inglés, *Insertion*).

El Intercambio Adyacente consiste en el intercambio entre posiciones adyacentes, y da lugar a $J - 1$ vecinos (siendo J el número de trabajos de la solución). El Intercambio General consiste en el intercambio entre dos posiciones, para todas las parejas de posiciones posibles, y da lugar a $J \times (J - 1) \div 2$ vecinos. La Inserción consiste en el desplazamiento de cada trabajo a todas las posiciones, y da lugar a $(J - 1)^2$ vecinos.

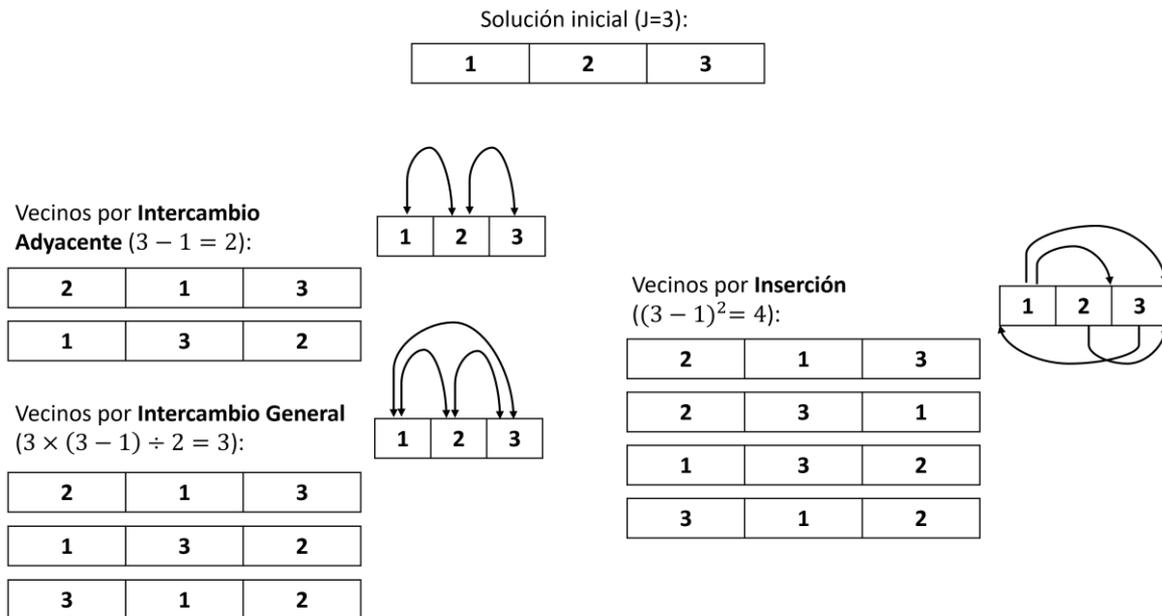


Figura 4-6. Ejemplo de vecinos por Intercambio Adyacente, Intercambio General e Inserción.

4.2.2 Grado de adaptación de los individuos o *fitness*

La evolución de la población depende de la calidad relativa de los individuos que compiten por reproducirse (Araujo & Cervigón, 2009), es decir, su grado de adaptación o adecuación a ser solución del problema. En un problema de maximización de la función objetivo, el grado de adaptación puede ser sencillamente el valor de la función objetivo de cada individuo, de modo que, a mayor valor de la función objetivo, mayor grado de adaptación.

Hay que tener en cuenta que, en un problema de minimización de la función objetivo, el individuo con mayor grado de adaptación es aquel que da lugar a un menor valor de la función objetivo. Para evitar confusiones, se utiliza el término *fitness* (Pérez et al., 2022), que tiene la misma función que el grado de adaptación: evaluar a los individuos de la población. La diferencia es que el *fitness* no es por definición mejor cuanto mayor sea su valor, como ocurre con el grado de adaptación. Es responsabilidad del usuario decidir en qué sentido mejora el *fitness*: al aumentar o al disminuir.

De este modo, en un problema de minimización en el que el *fitness* es el valor de la función objetivo, este mejora a medida que disminuye. En cambio, en un problema de maximización en el que el *fitness* es el valor de la función objetivo, este mejora cuando aumenta. En resumen, en general, el *fitness* mejora en el sentido en que la función objetivo mejora.

Es necesario mencionar que en ocasiones se mide el *fitness* de un individuo con funciones que no solo dependen del valor de la función objetivo de este, sino también de las restricciones o del valor de la función objetivo del resto de individuos (Pérez et al., 2022). No obstante, a partir de este punto del trabajo, cuando se hable del *fitness* del individuo, se hará referencia al valor de la función objetivo que genera.

4.2.3 Estrategias de selección

De entre la población, se deben seleccionar dos (o más) padres que serán encargados de reproducirse, permitiendo a la población renovarse. Se debe garantizar que los individuos más adaptados, es decir, aquellos con mejor valor de la función objetivo o mejor *fitness*, tienen una mayor probabilidad de ser padres que aquellos individuos menos buenos. Sin embargo, es importante dar la oportunidad también a los individuos

menos adaptados de ser seleccionados.

Existen diversos mecanismos de selección. Algunos de ellos son tan simples como la mera selección aleatoria de los padres (que pasa por alto la regla de favorecer a los individuos mejor adaptados), la selección del individuo mejor adaptado y de otro aleatorio, o la selección del mejor y peor individuo. Si bien estos mecanismos son válidos y pueden dar lugar a resultados muy buenos, existen otros más elaborados.

En primer lugar, la selección por torneo. Esta técnica, en su versión determinista, consiste en elegir aleatoriamente una pequeña muestra de la población, de dos o tres individuos, y seleccionar de ella al individuo mejor adaptado (Araujo & Cervigón, 2009). Este proceso se repite con tantas muestras de población como padres se quieren seleccionar.

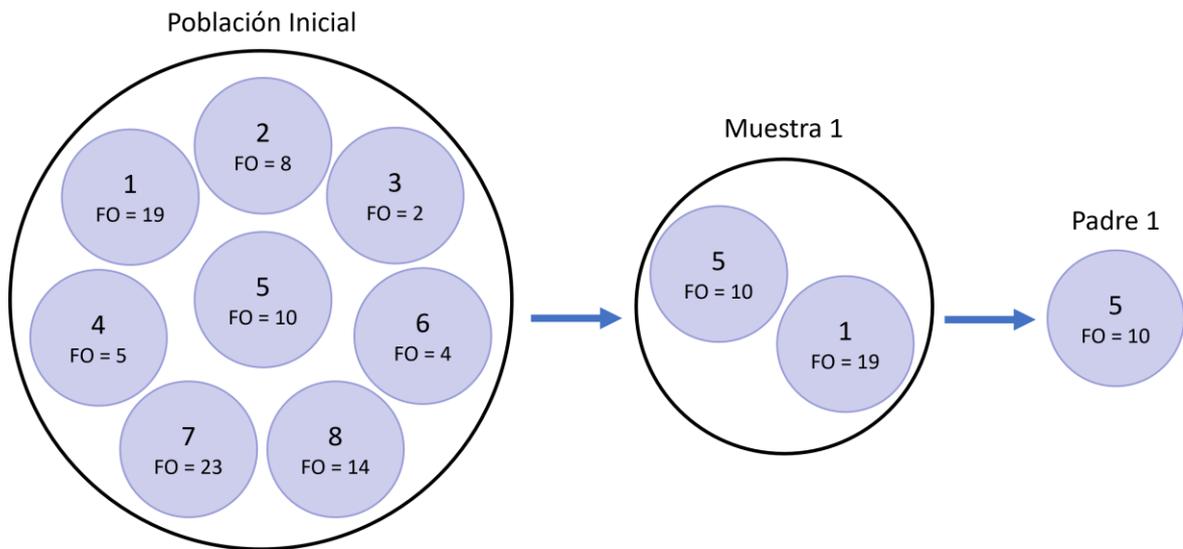


Figura 4-7. Ejemplo de la selección por torneo determinista de un padre en un problema de minimización.

Por otro lado, la selección por torneo probabilístico es muy similar, pero cuenta con una diferencia interesante. En lugar de elegir siempre al mejor individuo de la muestra, la selección se hace con una cierta probabilidad. En primer lugar, se debe especificar un umbral como parámetro del algoritmo. Una vez elegida la muestra, se genera un número aleatorio entre 0 y 1. Si este es mayor que el umbral, se elige al mejor individuo, si es igual o menor que el umbral, se elige al peor (Araujo & Cervigón, 2009).

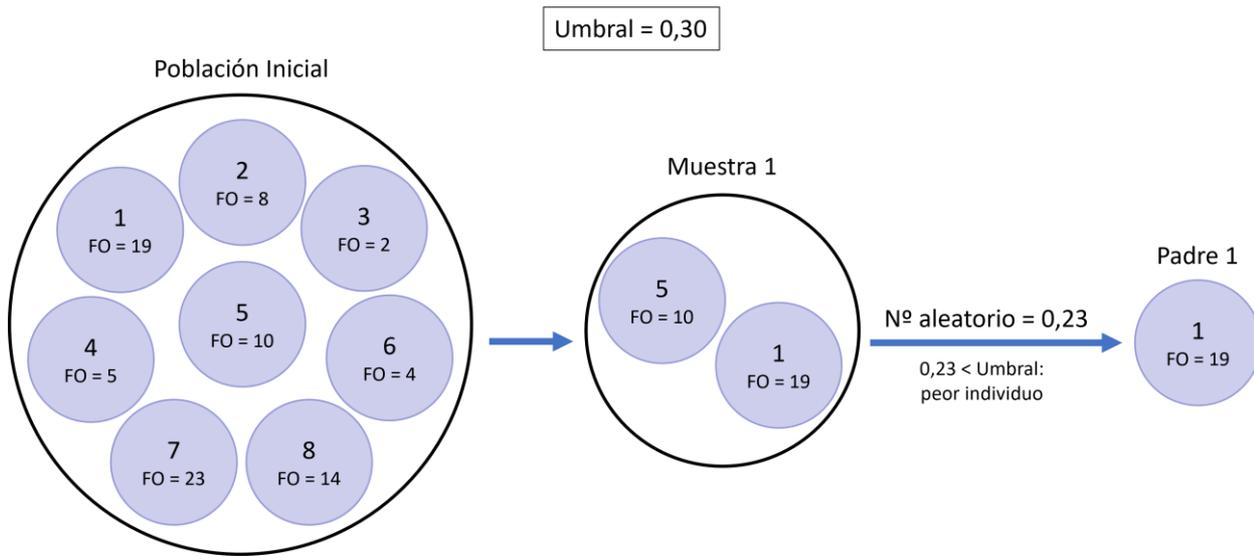


Figura 4-8. Ejemplo de la selección por torneo probabilístico de un padre en un problema de minimización.

Otro conocido mecanismo de selección es la selección por ruleta. En este caso, la probabilidad de cada individuo p_{ind} de ser seleccionado como padre es mayor cuanto mejor sea su *fitness*.

Como explican Araujo y Cervigón (2009), en un problema en el que el *fitness* mejora cuando aumenta (por ejemplo, un problema de maximización en el que el *fitness* de un individuo es su valor de la función objetivo), la probabilidad de cada individuo p_{ind} de ser seleccionado es:

$$p_{ind} = \frac{f(ind)}{\sum f(ind)} \quad (\forall ind = 1 \dots n)$$

siendo $f(ind)$ el *fitness* de cada individuo ind

y n el número total de individuos en la población.

Las puntuaciones acumuladas de cada individuo q_{ind} se definen como:

$$q_0 = 0$$

$$q_{ind} = p_1 + \dots + p_{ind} \quad (\forall ind = 1 \dots n)$$

A continuación, se elige un número aleatorio $a \in [0,1]$, y se selecciona como padre al individuo ind que cumple:

$$q_{ind-1} < a < q_{ind}$$

En la Figura 4-9 se muestra un ejemplo.

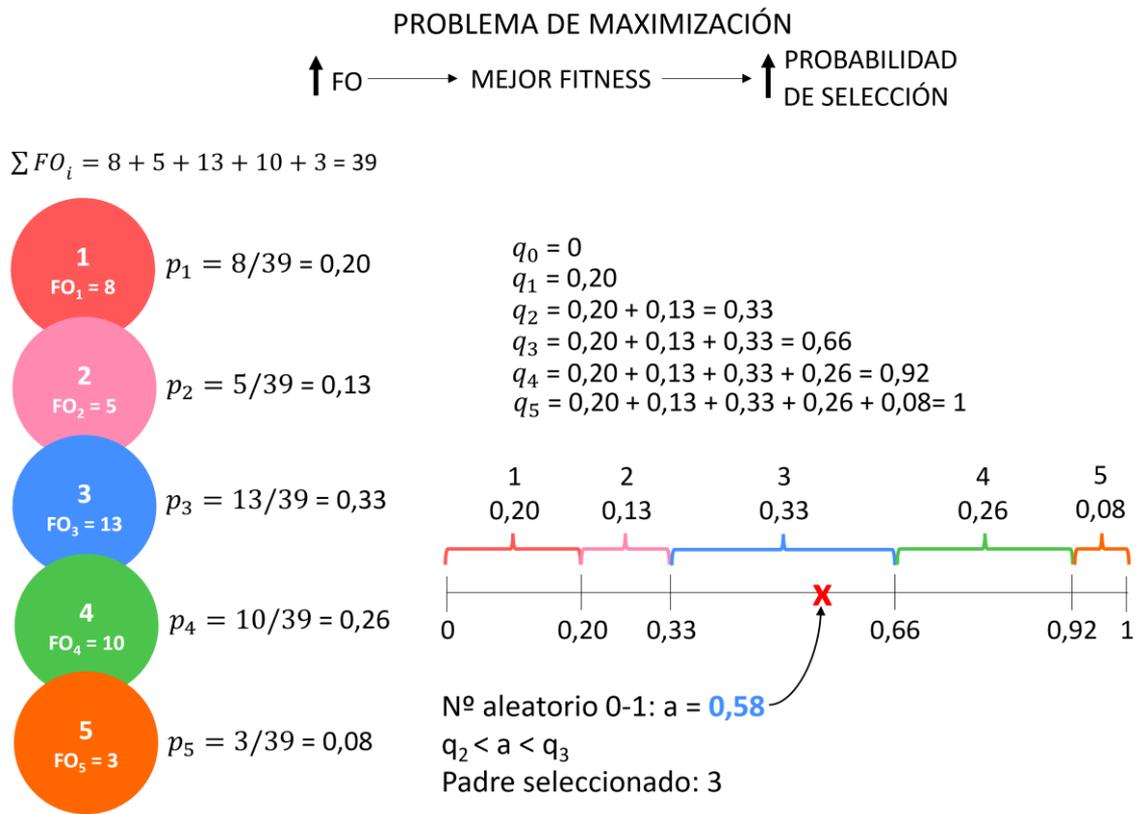


Figura 4-9. Ejemplo de selección por ruleta en un problema de maximización.

El mecanismo cambia cuando se trata de un problema en el que el *fitness* mejora cuando disminuye (por ejemplo, un problema de minimización en el que el *fitness* de un individuo es su valor de la función objetivo). En este caso, las p_{ind} de cada individuo se definen como:

$$p_{ind} = \frac{1}{\frac{f(ind)}{\Sigma \frac{1}{f(ind)}}} \quad (\forall ind = 1 \dots n)$$

Las puntuaciones acumuladas de cada individuo q_{ind} son:

$$q_0 = 0$$

$$q_{ind} = p_1 + \dots + p_{ind} \quad (\forall i = 1 \dots n)$$

A continuación, se elige un número aleatorio $a \in [0,1]$, y se selecciona como padre al individuo i que cumple:

$$q_{ind-1} < a < q_{ind}$$

En la Figura 4-10 se muestra un ejemplo.

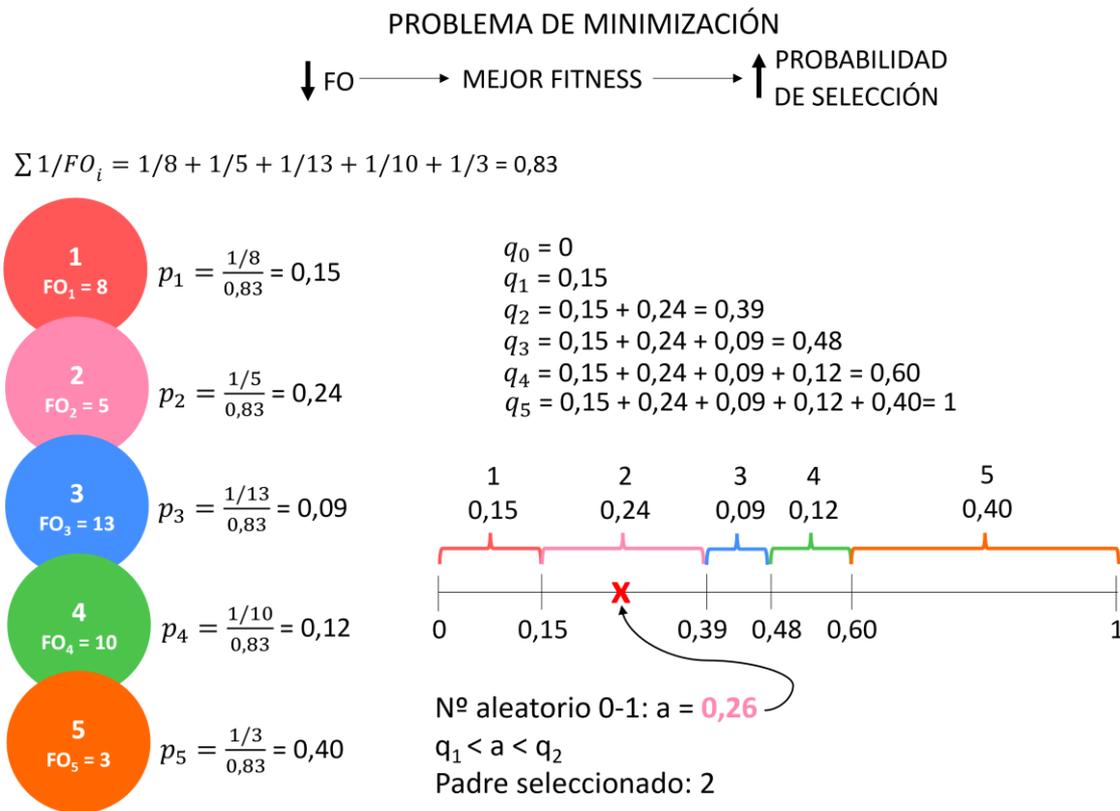


Figura 4-10. Ejemplo de selección por ruleta en un problema de minimización.

Existen más estrategias de selección, como el muestreo estocástico universal o el muestreo por restos (Araujo & Cervigón, 2009), pero estas no van a ser utilizadas en este trabajo, por lo que no se explicará su mecanismo.

4.2.4 Cruce

El cruce o reproducción de los padres es necesario para permitir la evolución de la población y que, de este modo, el algoritmo genético pueda acceder a nuevas regiones del espacio de búsqueda. El operador de cruce combina propiedades de dos individuos de la población anterior (los padres) para crear nuevos individuos (los hijos) (Araujo & Cervigón, 2009).

Se debe establecer una tasa para cada operador genético, ya sea de cruce o de mutación, de manera que el operador solo se aplica si un valor generado aleatoriamente está por debajo de la tasa especificada. En el caso del operador de cruce, la tasa es alta, entre el 60% y el 90%, de modo que hay una probabilidad de entre 0,6 y 0,9 de reproducción de los padres. Si se obtiene un valor por encima de la tasa y, por tanto, el operador no actúa, los propios padres se consideran los descendientes del proceso de recombinación de la pareja (Universidad de Granada, 2013).

Existen muchas posibilidades en cuanto al diseño del operador de cruce. En un problema cuyos individuos poseen una representación binaria, son muy utilizados el operador de cruce monopunto y el operador de cruce multipunto. El cruce monopunto consiste en seleccionar una posición en la cadena de ambos padres e intercambiar las partes de los padres divididas por dicha posición. A su vez, el cruce multipunto consiste en seleccionar dos o más posiciones en la cadena de los padres e intercambiar las partes de los padres divididas por dichos puntos de cruce (Araujo & Cervigón, 2009).

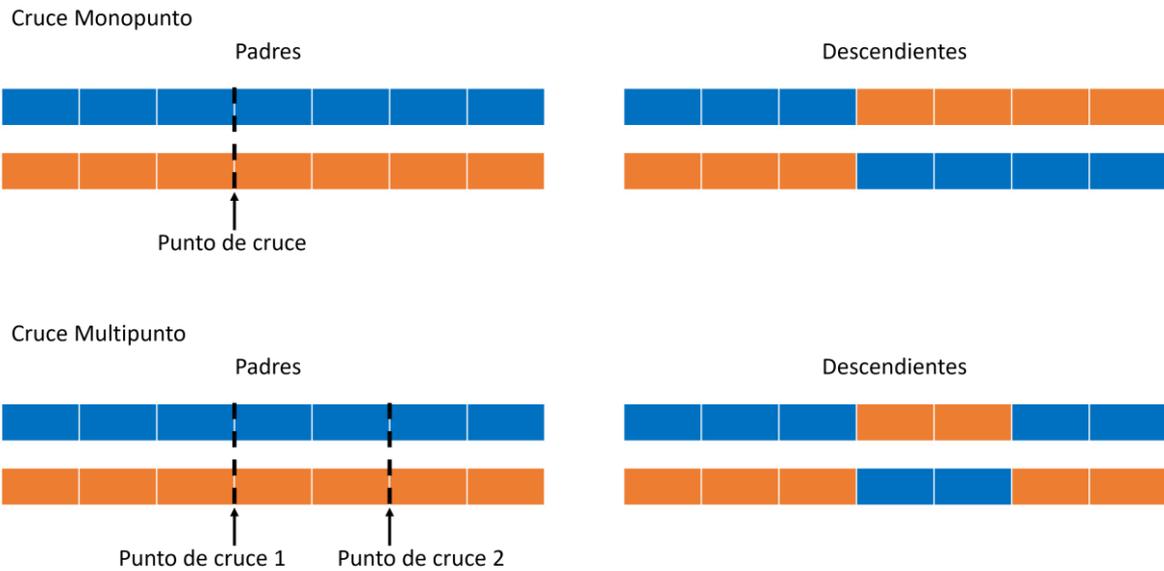


Figura 4-11. Arriba, ejemplo del operador de cruce monopunto. Abajo, ejemplo del operador de cruce multipunto.

Estos operadores de cruce, sin embargo, no son útiles cuando los individuos de la población constituyen una ordenación de trabajos, ya que los descendientes obtenidos serían inválidos, con trabajos repetidos y otros suprimidos. Esto se observa mejor en la siguiente imagen:

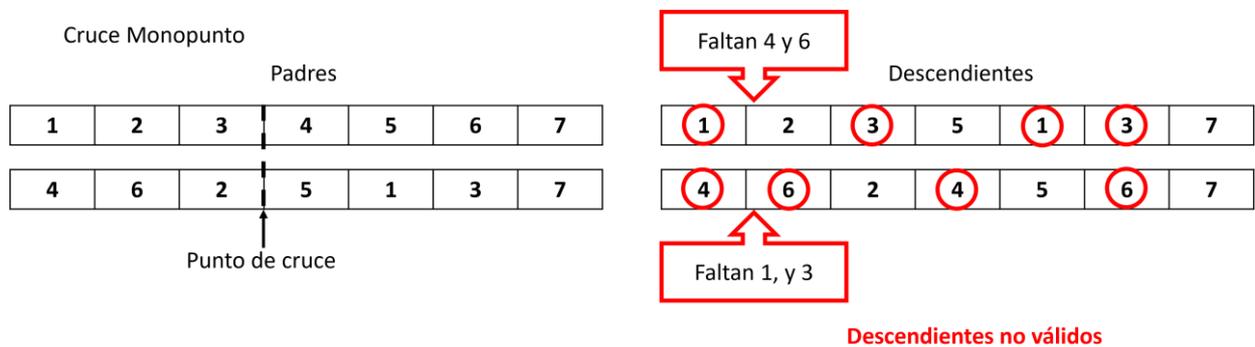


Figura 4-12. Ejemplo inválido del operador de cruce monopunto.

Para aquellos problemas en los que los individuos deben contener siempre los mismos caracteres y solo se diferencian en el orden de estos, existe el operador de cruce para representación de orden. Primero, se selecciona una posición en la cadena de ambos padres. Para el primer descendiente, se toma la parte del padre 1 antes del punto de cruce y se copia en el descendiente 1. El resto de genes se rellenan con los caracteres que faltan en el orden en que aparecen en el padre 2. Para el segundo descendiente, se toma la parte del padre 2 antes del punto de cruce y se copia en el descendiente 2. El resto de genes se rellenan con los caracteres que faltan en el orden en que aparecen en el padre 1.

Cruce para Representación de Orden

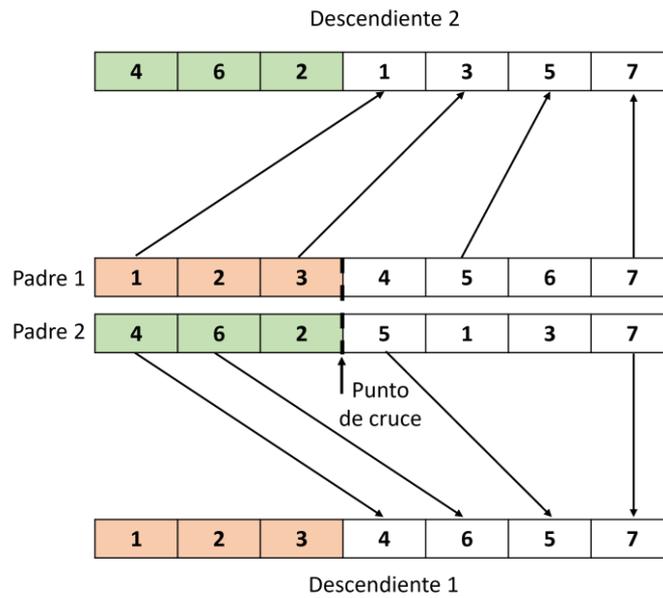


Figura 4-13. Ejemplo del operador de cruce para representación de orden.

Este operador también puede diseñarse con más de un punto de cruce, pero es necesario estar alerta, ya que aumenta la probabilidad de obtener descendientes iguales a los padres, y quizá no sea este el objetivo perseguido. El funcionamiento es el mismo que en el caso de un solo punto de cruce, pero en lugar de una sola parte copiada de padre a hijo, habrá dos partes: una antes del primer punto de cruce y otra después del segundo punto. En la Figura 4-13 puede verse un ejemplo. Precisamente, se puede observar que el Descendiente 1 obtenido es igual al Padre 1.

Cruce para Representación de Orden con dos puntos de cruce

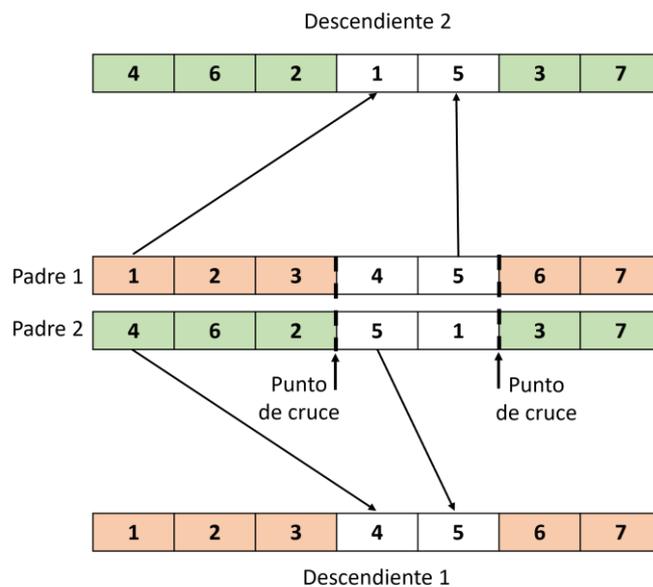


Figura 4-14. Ejemplo del operador de cruce para representación de orden con dos puntos de cruce.

4.2.5 Mutación

El operador de mutación crea un nuevo individuo realizando algún tipo de alteración, usualmente pequeña, en otro individuo (Araujo & Cervigón, 2009). Este operador permite alcanzar cualquier parte del espacio de búsqueda, contribuyendo a mantener la diversidad de la población. Se aplica sobre los descendientes obtenidos tras la fase de cruce, incluidos aquellos que coinciden con los padres porque el operador de cruce no actuó (Universidad de Granada, 2013).

Habitualmente, la tasa de mutación es bastante pequeña comparada con la tasa de cruce. Sin embargo, se puede innovar en este sentido introduciendo una tasa de mutación variable, que comienza siendo alta para ampliar la búsqueda en las generaciones iniciales, y se reduce a medida que avanza la evolución (Araujo & Cervigón, 2009).

Existen diversos operadores de mutación. Entre los más comunes se encuentra el operador de intercambio de pares, por el que se seleccionan aleatoriamente dos posiciones de la cadena del individuo y se intercambian los trabajos que las ocupan. Otro operador de mutación habitualmente utilizado es el de inserción, que consiste en seleccionar aleatoriamente un trabajo e insertarlo en una posición elegida también aleatoriamente (Pérez et al., 2022).

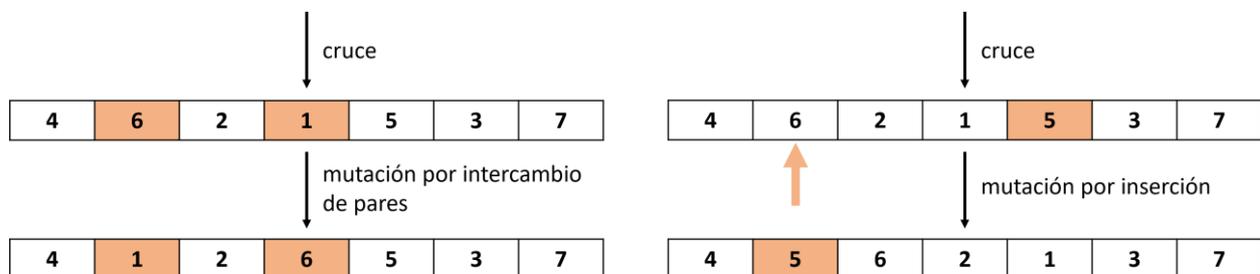


Figura 4-15. A la izquierda, ejemplo de mutación por intercambio de pares. A la derecha, ejemplo de mutación por inserción.

4.2.6 Proceso de reemplazo

En los Algoritmos Genéticos, el tamaño de la población suele mantenerse constante. Por esta razón, es lógico pensar que, para que los nuevos descendientes pasen a formar parte de la población, deben reemplazar a otros individuos de generaciones anteriores. En cuanto a este proceso de reemplazo, los Algoritmos Genéticos pueden clasificarse en generacionales o con estado estacionario.

Los Algoritmos Genéticos generacionales son aquellos en los que la población se renueva completamente de una generación a otra (Araujo & Cervigón, 2009). Esto quiere decir que los descendientes de una generación reemplazan por completo a sus predecesores. Para mantener el tamaño de la población constante, el número de descendientes debe ser en este caso igual al número total de individuos de la generación anterior.

Por otro lado, los Algoritmos Genéticos con estado estacionario son aquellos en los que los descendientes solo reemplazan a algunos individuos de la población anterior, por lo que se conserva parte de la población de generación en generación (Araujo & Cervigón, 2009). En este caso, que es el que concierne a este trabajo, el proceso de reemplazo cuenta con más variables que en el caso generacional, ya que debe decidirse qué individuos predecesores serán los reemplazados. Para tomar esta decisión, se puede optar entre:

1. Reemplazo de los padres: los descendientes sustituyen a sus padres (Araujo & Cervigón, 2009).
2. Reemplazo aleatorio: los individuos de la población anterior que serán reemplazados se eligen aleatoriamente (Araujo & Cervigón, 2009).

3. Reemplazo de los individuos peor adaptados: los individuos que serán reemplazados se eligen aleatoriamente de entre los que tienen peor *fitness* (Araujo & Cervigón, 2009).
4. Reemplazo de los individuos de adaptación similar: cada descendiente reemplaza a un individuo de la población anterior que tiene un *fitness* similar al suyo (Araujo & Cervigón, 2009).

Para el presente trabajo se utilizará el método 3 de reemplazo de los individuos con peor *fitness*. Posteriormente se explicará con más detalle el mecanismo utilizado.

4.2.7 Criterios de terminación

Por último, es necesario concretar las condiciones en las que el algoritmo deja de evolucionar, es decir, los requisitos ante los cuales se deja de seleccionar a los padres y de generar a los descendientes, y la población deja de desarrollarse. Es en este punto cuando se presenta la mejor solución encontrada hasta entonces: el individuo con mejor *fitness*.

Existen distintos criterios que pueden poner fin al algoritmo, y pueden usarse por separado o combinados. Según Araujo y Cervigón (2009), Michalewicz contempla las siguientes opciones:

1. Alcanzar un número máximo de generaciones.
2. Alcanzar un número máximo de llamadas al cálculo del *fitness*.
3. Llegar a una situación con escasas posibilidades de que se produzcan cambios significativos en la generación siguiente.

Los valores concretos de los dos primeros criterios dependen de la complejidad de cada problema, por lo que no tiene sentido dar estimaciones generalizadas. Por otro lado, la materialización del tercer criterio puede basarse en el número de genes que han convergido a un mismo valor, o en el número de generaciones seguidas cuyo mejor *fitness* o *fitness* medio no mejora (o no mejora lo suficiente) (Araujo & Cervigón, 2009).

5 REGLAS DE DESPACHO ELABORADAS

Antes de describir los Algoritmos Genéticos desarrollados, se van a presentar las cinco reglas de despacho utilizadas en algunos de ellos como parte de la población inicial. Estas reglas se han elaborado siguiendo el criterio personal del autor, en base al modelo elaborado. Posteriormente, en el capítulo 7 *Resultados computacionales*, se probarán con algunas instancias para comparar sus resultados.

5.1 Regla de Despacho 1

Consiste en ordenar las intervenciones de menor a mayor tiempo máximo de respuesta relativo de la etapa en la que entran (PAE, OR o POST).

5.2 Regla de Despacho 2

Se seleccionan las intervenciones que entran en la etapa POST y se ordenan de menor a mayor tiempo máximo de respuesta relativo. Después, se ordena el resto de las intervenciones de menor a mayor tiempo máximo de respuesta relativo de la etapa en la que entran (PAE o OR).

5.3 Regla de Despacho 3

Las intervenciones se ordenan según su etapa inicial: primero las que entran en POST, luego en OR y por último PAE. Cada grupo de intervenciones se ordena de menor a mayor pos_j , que se define de la siguiente manera:

$$pos_j = dias_j + 2 * tresp_rel_{i,j}$$

donde $tresp_rel_{i,j}$ es el tiempo máximo de respuesta relativo de la intervención j en su etapa inicial i , según la notación especificada en el apartado 3.2 *Cálculo de la función objetivo*

y $dias_j$ se calcula de forma distinta según cuál sea la etapa inicial:

- Si la etapa inicial de la intervención es PAE, ($etapa_inic_j = 0$), $dias_j$ es el número de días, desde el primer día en que se puede realizar la intervención quirúrgica respetando los tiempos mínimos entre PAE y OR y suponiendo que el PAE se realiza el primer día, hasta el último del horizonte de planificación, que hay cirujanos de la unidad adecuada con tiempo disponible para realizar la intervención quirúrgica.
- Si la etapa inicial de la intervención es OR, $dias_j$ es el número de días, desde el primer día en que se puede realizar la operación respetando los tiempos mínimos entre PAE y OR, hasta el último día del horizonte de planificación, que hay cirujanos de la unidad adecuada con tiempo disponible para

realizar la intervención quirúrgica.

- Si la etapa inicial es POST, $dias_j$ es el número de días que hay desde que se puede realizar el POST respetando los tiempos mínimos entre OR y POST y el último día del horizonte de planificación.

5.4 Regla de Despacho 4

De nuevo, se da prioridad a las intervenciones que entran en POST, y se ordenan de menor a mayor tiempo máximo de respuesta relativo.

A continuación, se ordena las intervenciones que entran en OR. Se calcula la diferencia entre la habilidad requerida por el cirujano responsable de la intervención y el tiempo máximo de respuesta. Esta diferencia es mayor cuando aumenta la habilidad requerida y cuando disminuye el tiempo máximo de respuesta relativo, por lo que la lógica dicta que las intervenciones deben ordenarse de mayor a menor.

Por último, las intervenciones que entran en PAE se ordenan de menor a mayor tiempo máximo de respuesta relativo.

5.5 Regla de Despacho 5

Se ordenan en primer lugar las intervenciones que entran en POST, luego las que entran en OR y por último las que entran en PAE. Cada grupo se ordena de menor a mayor tiempo máximo de respuesta relativo.

6 ALGORITMOS PROPUESTOS

A continuación, se van a presentar los Algoritmos Genéticos desarrollados en este trabajo. Como ya se ha dicho anteriormente, el objetivo de la aplicación de estos algoritmos es la obtención de una secuencia de intervenciones satisfactoria para un problema concreto, formado por unos datos específicos y una distribución de quirófanos y consultas entre las distintas unidades hospitalarias. A esta distribución se le denomina solución inicial, pues es necesario conocerla antes de poder hallar una secuencia de intervenciones adecuada.

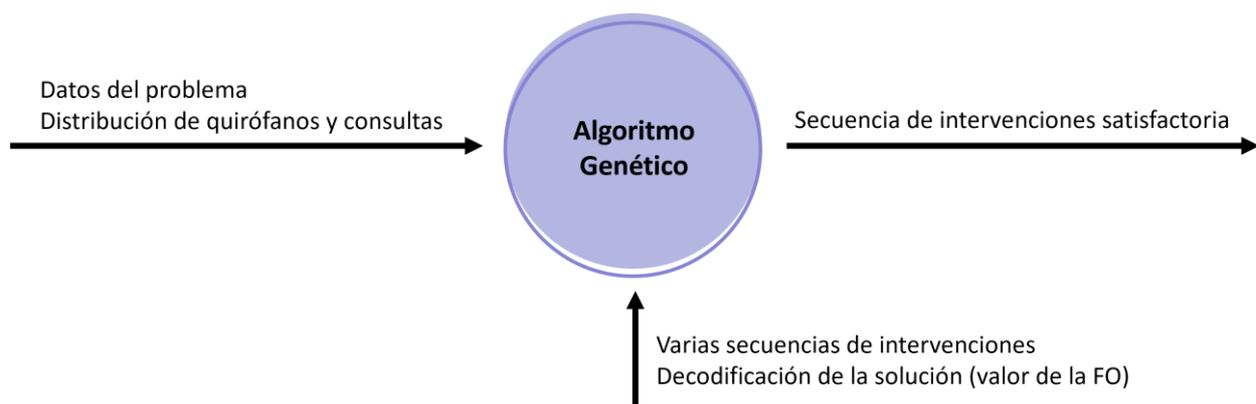


Figura 6-1. Esquema del funcionamiento del Algoritmo Genético.

Finalmente, una vez explicadas las características de los Algoritmos Genéticos elaborados, se presentará un algoritmo extendido para hallar una satisfactoria distribución de los quirófanos y las consultas, es decir, una solución inicial aceptable.

6.1 Algoritmo Genético 1 o AG1

6.1.1 Población inicial del AG1

El Algoritmo Genético 1 tiene una población inicial compuesta por 15 individuos. Dicho tamaño se mantendrá constante a lo largo de las iteraciones. De los 15 individuos, diez son secuencias generadas aleatoriamente y los otros cinco proceden de la aplicación de las Reglas de Despacho 1, 2, 3, 4 y 5.

6.1.2 Estrategia de selección del AG1

Los dos padres son seleccionados mediante dos torneos deterministas. En cada torneo hay dos concursantes, y

resultará ganador el mejor de ellos, es decir, el que tenga mejor *fitness*, convirtiéndose de este modo en uno de los padres.

6.1.3 Cruce del AG1

La tasa de cruce es 0.9, por lo que hay un 90% de probabilidad de que tenga lugar. En caso de que así sea, se genera un solo hijo mediante el operador de cruce para representación de orden. Concretamente, el hijo copia la primera mitad del Padre 1, y ordena el resto de sus componentes según el orden del Padre 2. Si no tiene lugar el cruce, el hijo es, aleatoriamente, una copia del Padre 1 o del Padre 2.

6.1.4 Mutación del AG1

La tasa de mutación es variable. Comienza siendo 1, es decir, hay un 100% de probabilidad de mutación, y en cada mutación que se realiza disminuye en 0.01 hasta alcanzar el valor de 0.5. A partir de ese valor, la tasa de mutación permanece constante. La mutación se realiza con un 100% de probabilidad si el Padre 1 y el Padre 2 son iguales.

La mutación consiste en el intercambio de tres genes (o trabajos, o intervenciones en el caso de este proyecto) seleccionados aleatoriamente. Si no se realiza mutación, el descendiente es el hijo.

6.1.5 Proceso de reemplazo del AG1

El descendiente generado (ya sea mutante o hijo) da un valor de la función objetivo, es decir, su *fitness*. Este valor se compara con el *fitness* de toda la población de la anterior generación, individuo a individuo. Para aquellos individuos que tienen un peor *fitness*, se decide aleatoriamente si será reemplazado por el nuevo descendiente, con una probabilidad del 40% de reemplazo.

Si el descendiente es mejor que alguno de los viejos individuos, pero la probabilidad no ha permitido realizar ningún reemplazo, el descendiente reemplaza al individuo con peor *fitness* de toda la población.

6.1.6 Criterio de terminación del AG1

Hay dos situaciones ante las cuales el Algoritmo Genético deja de iterar. En primer lugar, si se cumplen las 250 iteraciones. En segundo lugar, si se dan 60 iteraciones seguidas en las que no ha habido reemplazo de la población anterior por el descendiente, ya que este último siempre ha sido peor que los individuos de la generación pasada.

6.2 Algoritmo Genético 2 o AG2

6.2.1 Población inicial del AG2

El Algoritmo Genético 2 tiene una población inicial compuesta por 15 individuos. Dicho tamaño se mantendrá constante a lo largo de las iteraciones. De los 15 individuos, uno procede de la Regla de Despacho 5, diez son vecinos por Intercambio General de la Regla de Despacho 5 aleatoriamente generados y cuatro son secuencias generadas aleatoriamente.

6.2.2 Estrategia de selección del AG2

Los dos padres son seleccionados mediante dos torneos probabilísticos, con un umbral de 0.3. En cada torneo hay dos concursantes, de los que resultará ganador, convirtiéndose de este modo en padre, en el 30% de las ocasiones el que tenga peor *fitness*, y en el otro 70% el que tenga mejor *fitness*.

6.2.3 Cruce del AG2

La tasa de cruce es 0.9, por lo que hay un 90% de probabilidad de que tenga lugar. En caso de que así sea, se

genera un solo hijo mediante el operador de cruce para representación de orden. Concretamente, el hijo copia la primera mitad del Padre 1, y ordena el resto de sus componentes según el orden del Padre 2. Si no tiene lugar el cruce, el hijo es, aleatoriamente, una copia del Padre 1 o del Padre 2.

6.2.4 Mutación del AG2

La tasa de mutación es variable. Comienza siendo 1, es decir, hay un 100% de probabilidad de mutación, y en cada mutación que se realiza disminuye en 0.01 hasta alcanzar el valor de 0.5. A partir de ese valor, la tasa de mutación permanece constante. La mutación se realiza con un 100% de probabilidad si el Padre 1 y el Padre 2 son iguales.

La mutación consiste en el intercambio de tres genes (o trabajos, o intervenciones en el caso de este proyecto) seleccionados aleatoriamente. Si no se realiza mutación, el descendiente es el hijo.

6.2.5 Proceso de reemplazo del AG2

El descendiente generado (ya sea mutante o hijo) da un valor de la función objetivo, es decir, su *fitness*. Este valor se compara con el *fitness* de toda la población de la anterior generación, individuo a individuo. Para aquellos individuos que tienen *fitness* peor que el del descendiente y peor que la media de *fitness*, se decide aleatoriamente si será reemplazado por el nuevo descendiente, con una probabilidad del 60% de reemplazo.

Si el descendiente es mejor que alguno de los viejos individuos, pero no se ha realizado ningún cambio, se reemplaza a la mitad de los individuos que tengan peor *fitness* que el descendiente, concretamente a los que peores valores tengan.

6.2.6 Criterio de terminación del AG2

Hay dos situaciones ante las cuales el Algoritmo Genético deja de iterar. En primer lugar, si se cumplen las 250 iteraciones. En segundo lugar, si se dan 60 iteraciones seguidas en las que no ha habido reemplazo de la población anterior por el descendiente, ya que este último siempre ha sido peor que los individuos de la generación pasada.

6.3 Algoritmo Genético 3 o AG3

6.3.1 Población inicial del AG3

El Algoritmo Genético 3 tiene una población inicial compuesta por 15 individuos. Dicho tamaño se mantendrá constante a lo largo de las iteraciones. De los 15 individuos, cinco proceden de las Reglas de Despacho 1, 2, 3, 4 y 5 y diez son vecinos por Intercambio Adyacente de la Regla de Despacho 2 aleatoriamente generados.

6.3.2 Estrategia de selección del AG3

Los dos padres son seleccionados por el método de la ruleta. A cada individuo se le calcula su probabilidad de ser seleccionado y, posteriormente, se compara un número generado aleatoriamente con las puntuaciones acumuladas de los individuos. De este modo se obtienen sucesivamente los dos padres.

6.3.3 Cruce del AG3

La tasa de cruce es 0.9, por lo que hay un 90% de probabilidad de que tenga lugar. En caso de que así sea, se genera un solo hijo mediante el operador de cruce para representación de orden. Concretamente, el hijo copia la primera mitad del Padre 1, y ordena el resto de sus componentes según el orden del Padre 2. Si no tiene lugar el cruce, el hijo es, aleatoriamente, una copia del Padre 1 o del Padre 2.

6.3.4 Mutación del AG3

La tasa de mutación es variable. Comienza siendo 1, es decir, hay un 100% de probabilidad de mutación, y en cada mutación que se realiza disminuye en 0.01 hasta alcanzar el valor de 0.5. A partir de ese valor, la tasa de mutación permanece constante. La mutación se realiza con un 100% de probabilidad si el Padre 1 y el Padre 2 son iguales.

La mutación consiste en el intercambio de cuatro genes (o trabajos, o intervenciones en el caso de este proyecto) seleccionados aleatoriamente. Si no se realiza mutación, el descendiente es el hijo.

6.3.5 Proceso de reemplazo del AG3

El descendiente generado (ya sea mutante o hijo) da un valor de la función objetivo, es decir, su *fitness*. Este valor se compara con el *fitness* de toda la población de la anterior generación, individuo a individuo. Para aquellos individuos que tienen un peor *fitness*, se decide aleatoriamente si será reemplazado por el nuevo descendiente, con una probabilidad del 60% de reemplazo.

Si el descendiente es mejor que alguno de los viejos individuos, pero la probabilidad no ha permitido realizar ningún reemplazo, el descendiente reemplaza al individuo con peor *fitness* de toda la población.

6.3.6 Criterio de terminación del AG3

Hay dos situaciones ante las cuales el Algoritmo Genético deja de iterar. En primer lugar, si se cumplen las 250 iteraciones. En segundo lugar, si se dan 60 iteraciones seguidas en las que no ha habido reemplazo de la población anterior por el descendiente, ya que este último siempre ha sido peor que los individuos de la generación pasada.

6.4 Algoritmo Genético 4 o AG4

6.4.1 Población inicial del AG4

El Algoritmo Genético 5 tiene una población inicial compuesta por 15 individuos. Dicho tamaño se mantendrá constante a lo largo de las iteraciones. De los 15 individuos, uno procede de la Regla de Despacho 5 y catorce son vecinos por Intercambio General de la Regla de Despacho 5 aleatoriamente generados.

6.4.2 Estrategia de selección del AG4

El Padre 1 es el mejor individuo, es decir, aquel con mejor *fitness*, y el Padre 2 es un individuo seleccionado aleatoriamente.

6.4.3 Cruce del AG4

La tasa de cruce es 0.9, por lo que hay un 90% de probabilidad de que tenga lugar. En caso de que así sea, se genera un solo hijo mediante el operador de cruce para representación de orden. Concretamente, el hijo copia la primera mitad del Padre 1, y ordena el resto de sus componentes según el orden del Padre 2. Si no tiene lugar el cruce, el hijo es, aleatoriamente, una copia del Padre 1 o del Padre 2.

6.4.4 Mutación del AG4

La tasa de mutación es variable. Comienza siendo 1, es decir, hay un 100% de probabilidad de mutación, y en cada mutación que se realiza disminuye en 0.01 hasta alcanzar el valor de 0.5. A partir de ese valor, la tasa de mutación permanece constante. La mutación se realiza con un 100% de probabilidad si el Padre 1 y el Padre 2 son iguales.

La mutación consiste en el intercambio de cuatro genes (o trabajos, o intervenciones en el caso de este

proyecto) seleccionados aleatoriamente. Si no se realiza mutación, el descendiente es el hijo.

6.4.5 Proceso de reemplazo del AG4

El descendiente generado (ya sea mutante o hijo) da un valor de la función objetivo, es decir, su *fitness*. Este valor se compara con el *fitness* de toda la población de la anterior generación, individuo a individuo. Para aquellos individuos que tienen un peor *fitness*, se decide aleatoriamente si será reemplazado por el nuevo descendiente, con una probabilidad del 40% de reemplazo.

Si el descendiente es mejor que alguno de los viejos individuos, pero la probabilidad no ha permitido realizar ningún reemplazo, el descendiente reemplaza al individuo con peor *fitness* de toda la población.

6.4.6 Criterio de terminación del AG4

Hay dos situaciones ante las cuales el Algoritmo Genético deja de iterar. En primer lugar, si se cumplen las 250 iteraciones. En segundo lugar, si se dan 60 iteraciones seguidas en las que no ha habido reemplazo de la población anterior por el descendiente, ya que este último siempre ha sido peor que los individuos de la generación pasada.

6.5 Resumen de los Algoritmos Genéticos propuestos

Tabla 6-1. Resumen de los Algoritmos Genéticos propuestos.

	AG1	AG2	AG3	AG4
Población inicial	- 5: Reglas 1, 2, 3, 4, 5 - 10: aleatorias	- 11: Regla 5 y vecinos GS - 4: aleatorias	- 11: Regla 2 y vecinos AS - 4: Reglas 1, 3, 4, 5	- 15: Regla 5 y vecinos GS
Selección de padres	Torneo determinista	Torneo probabilístico (umbral = 0.3)	Ruleta	Mejor individuo y aleatorio
Cruce	- Tasa de cruce = 0.9 - Copia de la mitad del Padre 1, y el resto según el orden del Padre 2	Ídem AG1	Ídem AG1	Ídem AG1
Mutación	- Tasa de mutación variable: de 1 a 0.5 - Intercambio aleatorio de 3 genes	Ídem AG1	- Tasa de mutación variable: de 1 a 0.5 - Intercambio aleatorio de 4 genes	Ídem AG3
Reemplazo	- Individuos peores que el descendiente - Probabilidad de 0.4	- Individuos peores que el descendiente y que la media - Probabilidad de 0.6	- Individuos peores que el descendiente - Probabilidad de 0.6	Ídem AG1
Criterio de terminación	- 250 iteraciones - 60 iteraciones sin cambios en la población	Ídem AG1	Ídem AG1	Ídem AG1

6.6 Algoritmo Extendido

El algoritmo extendido se utiliza para poder hallar una solución inicial adecuada, es decir, una distribución de los quirófanos y las consultas entre las distintas unidades hospitalarias que dé resultados satisfactorios.

En el apartado 7.1.2 *Quirófanos y consultas virtuales*, se explicará que, durante la generación de instancias, a cada unidad se le ha asignado un número aleatorio de lo que se han denominado consultas y quirófanos virtuales. Las intervenciones de cada instancia se han asignado a las unidades hospitalarias según este número, que es desconocido. De este modo, aquella unidad que recibió un número mayor de quirófanos y consultas virtuales, tendrá en su lista de espera un mayor número de intervenciones a realizar.

Sabiendo esto, se puede deducir que la distribución de quirófanos y consultas reales más parecida a la asignación de quirófanos y consultas virtuales durante la generación de la instancia, será la que dará lugar a un valor de la función objetivo mejor.

Puesto que, como ya se ha dicho, los quirófanos y consultas virtuales de cada unidad se desconocen, se ha elaborado este algoritmo extendido, que pretende encontrar una solución inicial satisfactoria.

Este algoritmo consiste en comparar el valor de la función objetivo, obtenido con un Algoritmo Genético, para distintas soluciones iniciales en tres fases. En la primera fase, se ejecuta el Algoritmo Genético para un número de soluciones iniciales aleatorias. En la segunda fase, se ejecuta el Algoritmo Genético para un número de soluciones iniciales similares a la solución inicial de la primera fase que diera el menor valor de la función objetivo. En la tercera fase, se ejecuta el Algoritmo Genético para un número de soluciones iniciales similares a la solución inicial de la segunda fase que diera el menor valor de la función objetivo.

Este algoritmo busca mejorar una solución existente explorando vecindades cercanas en el espacio de soluciones. Esta idea es propia de los algoritmos de búsqueda local, pero, a diferencia de ellos, que continúan hasta que no se pueden encontrar mejoras significativas, en este caso solo se realizan dos iteraciones.

Las soluciones iniciales se han representado como matrices. Estas matrices poseen tantas filas como unidades tenga el problema, una primera columna que representa las consultas, y una segunda columna que representa los quirófanos.

	consultas	quirófanos
Unidad 0	3	2
Unidad 1	2	1
Unidad 2	2	1
Unidad 3	1	1
Unidad 4	4	3
	12	8
	consultas totales	quirófanos totales

Figura 6-2. Ejemplo de solución inicial.

La búsqueda de una solución inicial satisfactoria para cada instancia comienza con la fase 1, que consiste en la generación de un número de soluciones iniciales aleatorias. Para ello, se utiliza la función “asigna_mat”, que se explica a continuación.

La función “asigna_mat” requiere como datos de entrada la instancia para la que se van a generar las soluciones iniciales aleatorias y el número de matrices deseado. Conociendo, a través de la instancia, el número de unidades, de quirófanos y de consultas, la función elabora las matrices deseadas.

En primer lugar, la función verifica qué es mayor, el número de consultas o de quirófanos, y comienza la matriz por la columna correspondiente. Suponiendo que sea mayor el número de consultas, la función comienza asignando a cada unidad un número aleatorio de consultas entre 1 y un máximo *max_cons*:

$$max_cons \approx 2 \times \frac{n_cons}{n_ud}$$

Es decir, el máximo es el doble de las consultas que a cada unidad le correspondería si el total de ellas se repartiera por igual entre todas las unidades.

Una vez se tienen estas consultas aleatorias, se debe verificar que el total de consultas en la matriz coincide con el total real, es decir, n_cons . En caso contrario:

- Mientras que el total de consultas de la matriz sea mayor que n_cons , se selecciona una unidad al azar y se le quita una consulta (siempre y cuando no tenga una sola consulta).
- Mientras que el total de consultas sea menor que n_cons , se selecciona una unidad al azar y se le da otra consulta más.

Una vez finalizado este proceso, se tienen las consultas definitivas asignadas a cada unidad. Para generar los quirófanos de la matriz, se realiza una regla de tres para cada unidad de manera que, si para un total de consultas n_cons , la unidad u tiene tantas consultas, para un total de quirófanos n_quir , la unidad u tendrá X quirófanos.

Si hubiera más quirófanos que consultas, el procedimiento sería el mismo, pero calculando en primer lugar los quirófanos de la matriz, y después las consultas.

De esta manera, para cada instancia se obtiene el número deseado de soluciones iniciales aleatorias. Este número varía para cada tipo de instancia, ya que, cuantas más unidades hospitalarias haya (y, por tanto, también más consultas y quirófanos), más posibles matrices existirán, y más matrices aleatorias se necesitarán para tener una muestra significativa. Se han tomado los siguientes valores:

Tabla 6-2. Número de soluciones iniciales aleatorias para cada conjunto de instancias.

Tipo de instancias según el horizonte de planificación	Instancias	Número de unidades	Número de soluciones iniciales aleatorias
Instancias medianas	1 - 10	3	15
	11 - 20	5	20
	21-30	7	25
Instancias grandes	31 - 35	3	15
	36 - 40	5	20

Después de generar las matrices aleatorias, se ejecuta el AG3 en todas ellas. Se ha decidido utilizar el AG3 porque, como se verá en el apartado 7.2 *Comparación de Algoritmos Genéticos*, es el mejor de todos los Algoritmos Genéticos. A continuación, se elige la solución inicial para la que el Algoritmo Genético encuentra el mejor valor de la función objetivo. Esta matriz se toma como la mejor solución inicial de la fase 1.

La fase 2 consiste en la evaluación de soluciones iniciales similares a la mejor solución inicial de la fase 1. Para ello, la mejor matriz de la fase 1 se ha utilizado como dato de entrada para una nueva función: “matriz_sim”, con el fin de obtener matrices parecidas a esta, que puedan dar mejores valores de la función objetivo. A continuación, se explica el mecanismo de esta función.

La función “matriz_sim”, cuyo nombre hace referencia a su finalidad de generar matrices similares a una dada, recibe como datos de entrada la matriz de partida y el número de matrices parecidas a esta que se desea crear.

Para cada matriz a generar, el proceso comienza copiando la matriz dada. A continuación, se genera un número aleatorio $n_aleatorio$ entre 0 y 1 que determinará qué pequeña modificación se va a realizar:

- Si $0 \leq n_{aleatorio} \leq 0.6$, es decir, con una probabilidad del 60%, la modificación consiste en que la unidad donante cede una consulta a la unidad receptora.
- Si $0.60 < n_{aleatorio} \leq 0.65$, o sea, con una probabilidad del 5%, la modificación consiste en que la unidad donante cede un quirófano a la unidad receptora.
- Si $0.65 < n_{aleatorio} \leq 0.75$, es decir, con una probabilidad del 10%, la modificación consiste en que la unidad donante cede una consulta y un quirófano a la unidad receptora.
- Si $0.75 < n_{aleatorio} \leq 0.9$, es decir, con una probabilidad del 15%, la modificación consiste en que la unidad donante cede una consulta a una unidad receptora y un quirófano otra unidad receptora.
- Si $0.9 < n_{aleatorio} \leq 1$, es decir, con una probabilidad del 10%, la modificación consiste en que la unidad donante cede una consulta y un quirófano a una unidad receptora, y una consulta a otra unidad receptora.

En todos estos casos, las unidades donantes o receptoras se eligen aleatoriamente, pero siempre teniendo en cuenta que el número mínimo de consultas y quirófanos de cada unidad es 1, y que donante y receptor no pueden ser la misma unidad, pues la matriz generada quedaría igual a la inicial. Por supuesto, en ningún caso se generarán dos matrices iguales.

Como se estaba comentando antes de explicar la función “matriz_sim”, una vez elegida la mejor matriz de la fase 1, esta se introduce como entrada de la función, para crear otras matrices parecidas a esta que podrían dar buenos resultados. De nuevo, el número de matrices similares que se generan depende de la instancia, y se ha especificado en la Tabla 6-3.

Una vez generadas las soluciones iniciales de la fase 2, se ejecuta el AG3 en cada una de ellas. Se elige como mejor solución inicial de la fase 2 a aquella para la que se obtiene un menor valor de la función objetivo.

La fase 3 consiste en la evaluación de soluciones iniciales similares a la mejor solución inicial de la fase 2. Para ello, la mejor matriz de la fase 2 se introduce como entrada de la función “matriz_sim”, y se generan las matrices similares. Una vez creadas, se vuelve a elegir la mejor de ellas, de manera que se tiene la mejor matriz de cada una de las tres fases.

Por último, de entre esas tres matrices, se selecciona como solución inicial satisfactoria a aquella que da lugar a un menor valor de la función objetivo, que es la mejor de todas las soluciones iniciales analizadas.

Tabla 6-3. Número de soluciones iniciales para cada conjunto de instancias en cada fase.

Instancias	Fase 1 Número de soluciones iniciales aleatorias	Fase 2 Número de soluciones iniciales similares a la mejor de la fase 1	Fase 3 Número de soluciones iniciales similares a la mejor de la fase 2	Número total de soluciones iniciales analizadas
1 - 10	15	10	10	35
11 - 20	20	15	15	50
21 - 30	25	20	20	65
31 - 35	15	10	10	35
36 - 40	20	15	15	50

En la Tabla 6-4 se muestran los resultados obtenidos para las cuatro primeras instancias. Las dos primeras columnas pertenecen a la fase 1 y recogen las soluciones iniciales aleatorias y los resultados obtenidos con el AG3 para cada una de ellas. De estos valores, se subraya en verde el mejor, es decir, el mínimo. La solución

inicial que da lugar a este mínimo, es decir, la mejor matriz de la primera columna, se utiliza para elaborar las soluciones iniciales de la fase 2, que aparecen en la tercera columna, junto con sus valores de la función objetivo en la cuarta. De nuevo, se subraya en verde el mejor valor de la función objetivo de la cuarta columna, y se utiliza la matriz que ha dado lugar a ese mínimo para generar las matrices similares de la fase 3, que se recogen en la quinta columna. Los valores de la función objetivo de estas matrices se recogen en la sexta columna, y se vuelve a subrayar en verde el mejor resultado.

De los tres resultados en verde, el mínimo de todos ellos es elegido como el mejor, y se subraya también en verde la solución inicial que ha dado lugar a dicho resultado. Esta solución inicial se considera la más satisfactoria de entre las analizadas para la instancia en cuestión.

Tabla 6-4. Resultados del Algoritmo Extendido.

Instancia 1					
Soluciones iniciales aleatorias	Obj. AG3	Soluciones iniciales similares a 4)	Obj. AG3	Mat. Similar a 17)	Obj. AG3
1) [[3, 2], [4, 2], [1, 1]]	1267	16) [[2, 1], [4, 2], [2, 2]]	1211	26) [[2, 1], [5, 3], [1, 1]]	1171
2) [[2, 1], [2, 2], [4, 2]]	1241	17) [[2, 1], [4, 3], [2, 1]]	1144	27) [[3, 1], [3, 3], [2, 1]]	1158
3) [[5, 3], [1, 1], [2, 1]]	1387	18) [[2, 2], [4, 2], [2, 1]]	1233	28) [[2, 1], [3, 2], [3, 2]]	1224
4) [[2, 1], [5, 3], [1, 1]]	1172	19) [[2, 2], [5, 2], [1, 1]]	1280	29) [[1, 1], [5, 3], [2, 1]]	1173
5) [[1, 1], [2, 1], [5, 3]]	1424	20) [[3, 1], [4, 3], [1, 1]]	1170	30) [[3, 1], [4, 3], [1, 1]]	1174
6) [[3, 2], [2, 1], [3, 2]]	1401	21) [[2, 1], [5, 2], [1, 2]]	1206	31) [[1, 1], [4, 3], [3, 1]]	1173
7) [[1, 1], [4, 2], [3, 2]]	1242	22) [[3, 1], [4, 2], [1, 2]]	1215	32) [[2, 2], [3, 2], [3, 1]]	1241
8) [[1, 1], [3, 2], [4, 2]]	1253	23) [[3, 1], [3, 2], [2, 2]]	1224	33) [[3, 1], [2, 2], [3, 2]]	1241
9) [[4, 2], [3, 2], [1, 1]]	1265	24) [[1, 1], [6, 3], [1, 1]]	1185	34) [[2, 1], [3, 3], [3, 1]]	1156
10) [[1, 1], [5, 2], [2, 2]]	1245	25) [[1, 1], [5, 3], [2, 1]]	1173	35) [[3, 2], [2, 2], [3, 1]]	1265
11) [[3, 2], [3, 2], [2, 1]]	1229				
12) [[4, 2], [2, 1], [2, 2]]	1395				
13) [[2, 1], [3, 2], [3, 2]]	1220				
14) [[5, 3], [2, 1], [1, 1]]	1441				
15) [[3, 2], [1, 1], [4, 2]]	1338				
Instancia 2					
Mat. Aleatorias	Obj. AG3	Mat. Similares a 7)	Obj. AG3	Mat. Similares a 17)	Obj. AG3
1) [[5, 3], [2, 1], [1, 1]]	1693	16) [[4, 2], [2, 1], [2, 2]]	1388	26) [[3, 2], [1, 1], [4, 2]]	1394
2) [[1, 1], [5, 2], [2, 2]]	1670	17) [[3, 2], [2, 1], [3, 2]]	1334	27) [[4, 2], [2, 2], [2, 1]]	1426
3) [[1, 1], [4, 2], [3, 2]]	1551	18) [[3, 1], [2, 1], [3, 3]]	1437	28) [[3, 2], [3, 1], [2, 2]]	1388
4) [[5, 3], [1, 1], [2, 1]]	1493	19) [[4, 3], [1, 1], [3, 1]]	1436	29) [[2, 2], [3, 1], [3, 2]]	1370
5) [[2, 1], [3, 2], [3, 2]]	1415	20) [[5, 2], [1, 1], [2, 2]]	1435	30) [[1, 1], [3, 1], [4, 3]]	1684
6) [[2, 1], [4, 2], [2, 2]]	1465	21) [[3, 1], [2, 2], [3, 2]]	1396	31) [[4, 2], [1, 1], [3, 2]]	1386
7) [[4, 2], [1, 1], [3, 2]]	1383	22) [[3, 1], [1, 1], [4, 3]]	1507	32) [[3, 3], [3, 1], [2, 1]]	1438
8) [[4, 2], [3, 2], [1, 1]]	1696	23) [[2, 1], [2, 1], [4, 3]]	1494	33) [[2, 2], [2, 1], [4, 2]]	1379
9) [[3, 2], [4, 2], [1, 1]]	1715	24) [[5, 3], [1, 1], [2, 1]]	1493	34) [[2, 1], [3, 1], [3, 3]]	1476
10) [[2, 1], [1, 1], [5, 3]]	1506	25) [[4, 1], [1, 2], [3, 2]]	1430	35) [[2, 1], [2, 2], [4, 2]]	1437
11) [[3, 2], [1, 1], [4, 2]]	1394				
12) [[2, 1], [2, 1], [4, 3]]	1479				

13) [[3, 2], [3, 2], [2, 1]]	1416				
14) [[2, 1], [1, 1], [5, 3]]	1516				
15) [[2, 1], [5, 3], [1, 1]]	1827				
Instancia 3					
Mat. Aleatorias	Obj. AG3	Mat. Similares a 10)	Obj. AG3	Mat. Similares a 16)	Obj. AG3
1) [[1, 1], [4, 2], [3, 2]]	1377	16) [[5, 3], [1, 1], [2, 1]]	1119	26) [[5, 3], [2, 1], [1, 1]]	1146
2) [[2, 1], [3, 2], [3, 2]]	1438	17) [[4, 3], [3, 1], [1, 1]]	1171	27) [[4, 3], [1, 1], [3, 1]]	1123
3) [[3, 2], [2, 1], [3, 2]]	1242	18) [[4, 2], [3, 2], [1, 1]]	1216	28) [[5, 2], [1, 1], [2, 2]]	1228
4) [[3, 2], [4, 2], [1, 1]]	1232	19) [[4, 2], [3, 1], [1, 2]]	1282	29) [[5, 2], [1, 2], [2, 1]]	1176
5) [[1, 1], [2, 1], [5, 3]]	1430	20) [[3, 2], [3, 1], [2, 2]]	1242	30) [[3, 2], [2, 1], [3, 2]]	1234
6) [[3, 2], [1, 1], [4, 2]]	1256	21) [[3, 2], [3, 2], [2, 1]]	1202	31) [[4, 2], [1, 2], [3, 1]]	1182
7) [[2, 1], [2, 2], [4, 2]]	1431	22) [[4, 2], [2, 2], [2, 1]]	1171	32) [[4, 3], [2, 1], [2, 1]]	1113
8) [[2, 1], [5, 3], [1, 1]]	1478	23) [[4, 2], [2, 1], [2, 2]]	1236	33) [[4, 2], [2, 1], [2, 2]]	1236
9) [[4, 2], [3, 2], [1, 1]]	1229	24) [[4, 3], [2, 1], [2, 1]]	1122	34) [[6, 3], [1, 1], [1, 1]]	1169
10) [[5, 3], [2, 1], [1, 1]]	1159	25) [[5, 2], [2, 1], [1, 2]]	1271	35) [[4, 2], [1, 1], [3, 2]]	1235
11) [[2, 1], [4, 2], [2, 2]]	1447				
12) [[3, 2], [3, 2], [2, 1]]	1187				
13) [[4, 2], [2, 2], [2, 1]]	1174				
14) [[2, 1], [4, 3], [2, 1]]	1442				
15) [[4, 2], [2, 1], [2, 2]]	1236				
Instancia 4					
Mat. Aleatorias	Obj. AG3	Mat. Similares a 1)	Obj. AG3	Mat. Similares a 16)	Obj. AG3
1) [[4, 2], [3, 2], [1, 1]]	692	16) [[3, 1], [4, 2], [1, 2]]	613	26) [[4, 2], [3, 1], [1, 2]]	626
2) [[2, 1], [2, 1], [4, 3]]	705	17) [[3, 2], [4, 2], [1, 1]]	695	27) [[4, 1], [3, 2], [1, 2]]	613
3) [[2, 1], [5, 3], [1, 1]]	704	18) [[5, 2], [2, 2], [1, 1]]	705	28) [[3, 1], [3, 1], [2, 3]]	723
4) [[4, 2], [1, 1], [3, 2]]	737	19) [[3, 1], [3, 2], [2, 2]]	736	29) [[3, 1], [4, 1], [1, 3]]	608
5) [[2, 1], [4, 2], [2, 2]]	749	20) [[5, 2], [2, 1], [1, 2]]	634	30) [[3, 2], [3, 1], [2, 2]]	738
6) [[2, 1], [4, 3], [2, 1]]	823	21) [[4, 2], [2, 2], [2, 1]]	787	31) [[4, 1], [3, 1], [1, 3]]	609
7) [[5, 3], [1, 1], [2, 1]]	825	22) [[4, 3], [2, 1], [2, 1]]	827	32) [[2, 1], [5, 2], [1, 2]]	617
8) [[5, 3], [2, 1], [1, 1]]	723	23) [[5, 2], [1, 1], [2, 2]]	748	33) [[3, 1], [3, 2], [2, 2]]	742
9) [[3, 2], [3, 2], [2, 1]]	787	24) [[3, 1], [3, 3], [2, 1]]	825	34) [[2, 1], [4, 2], [2, 2]]	744
10) [[3, 2], [1, 1], [4, 2]]	741	25) [[2, 1], [4, 2], [2, 2]]	748	35) [[4, 2], [2, 1], [2, 2]]	741
11) [[3, 2], [2, 1], [3, 2]]	738				
12) [[1, 1], [4, 2], [3, 2]]	746				
13) [[2, 1], [1, 1], [5, 3]]	721				
14) [[2, 1], [3, 2], [3, 2]]	746				
15) [[2, 2], [4, 2], [2, 1]]	793				

En algunas instancias puede suceder que la solución inicial más satisfactoria, subrayada en verde, aparezca también en otra columna subrayada en amarillo, con un valor de la función objetivo un poco peor.

Por ejemplo, en la instancia 3, la solución inicial considerada la mejor es $[[4, 3], [2, 1], [2, 1]]$, que consigue un valor de la función objetivo en la fase 3 de 1113. En la segunda fase, sin embargo, aparece también la matriz, pero con un valor de la función objetivo de 1122, es decir, el valor de la función objetivo se incrementa en un 0.81%.

En la instancia 8 sucede lo mismo. La solución inicial para la cual se consigue el menor valor de la función objetivo es $[[3, 2], [3, 2], [2, 1]]$, con un 609 en la fase 3. En la fase 1 aparece la misma matriz, pero da lugar a un valor de la función objetivo de 613, un 0.66% más.

Que existan estas pequeñas variaciones en el resultado de la función objetivo, no significa que el Algoritmo Genético no cumpla debidamente con su función, simplemente evidencia la componente aleatoria que interviene en el algoritmo: con unos mismos datos de entrada, no siempre se llegará a la misma solución, pero sí a resultados del mismo orden de magnitud.

Distinto sería si, ante la misma instancia y solución inicial, el AG3 obtuviera resultados muy dispares. Esto sí sería señal de deficiencias en el funcionamiento del algoritmo, pero no ha sido el caso.

En la Tabla 6-5 se recogen las instancias en las que ha habido repetición de la mejor solución inicial, y el incremento de la función objetivo que se observa.

Tabla 6-5. Incremento del valor de la función objetivo para una misma instancia y solución inicial.

Instancia	Mejor solución inicial	Mejor valor de la función objetivo	Peor valor de la función objetivo	Incremento
3	$[[4, 3], [2, 1], [2, 1]]$	1113	1122	0.81%
8	$[[3, 2], [3, 2], [2, 1]]$	609	613	0.66%
11	$[[2, 1], [4, 3], [3, 2], [2, 1], [1, 1]]$	1725	1736	0.64%
33	$[[4, 2], [3, 1], [1, 2]]$	3192	3200	0.25%
35	$[[2, 1], [4, 2], [2, 2]]$	4097	4097	0.00%

Efectivamente, el incremento del valor de la función objetivo es prácticamente despreciable, con una media de 0.47%.

Los resultados obtenidos al utilizar una solución inicial adecuada hallada a través del Algoritmo Extendido se compararán, en el capítulo 7.2 *Comparación de Algoritmos Genéticos*, con los resultados obtenidos al utilizar una solución inicial aleatoria cualquiera.

7 RESULTADOS COMPUTACIONALES

Los estudios de simulación basados en un conjunto de instancias de muestra son el método más utilizado para la evaluación del funcionamiento de los algoritmos aproximados. No existe un enfoque universalmente aceptado para llevar a cabo dichos estudios de simulación, pero se pueden sintetizar en dos pasos (Framiñán et al., 2014):

1. Generación del banco de pruebas. Un cierto número de instancias del modelo son, o bien generadas específicamente para el modelo, o bien adaptadas a partir de otras empleadas para otros modelos (Framiñán et al., 2014). Algunos aspectos a tener en cuenta son el tamaño de la instancia, que debe estar en línea con las dimensiones del problema real, y el número de instancias generadas, que debe ser suficiente para proporcionar resultados significativos.
2. Análisis de los algoritmos. Los algoritmos que se desea comparar se ejecutan para cada una de las instancias. A continuación, se utiliza algún parámetro para establecer el desempeño relativo de cada algoritmo en el banco de pruebas (Framiñán et al., 2014).

7.1 Generación del banco de pruebas

La función elaborada para generar las instancias se denomina “geninstancia”. Para desarrollar la función, se ha tomado como referencia el procedimiento utilizado por Cañete et al. (2022), adaptándolo a las especificaciones del problema en cuestión.

7.1.1 Parámetros

La función “geninstancia” recibe como parámetros de entrada:

- *inst_inic*: el número identificativo de la primera instancia que se generará.
- *num_inst*: el número de instancias a generar.
- *n_ud*: el número de unidades hospitalarias en las instancias del conjunto. Ya ha sido definido en el apartado 2.5 Notación.
- *n_cons*: el número de consultas en las instancias del conjunto. Ya ha sido definido en el apartado 2.5 Notación.
- *n_quir*: el número de quirófanos en las instancias del conjunto. Ya ha sido definido en el apartado 2.5 Notación.
- *alfa*: factor de control de cobertura de los cirujanos respecto a las consultas y quirófanos disponibles. Se escribe en tanto por uno.
- *beta*: factor de saturación de las consultas y quirófanos con respecto a las intervenciones a realizar. Se escribe en tanto por uno.
- *H*: número de días en el horizonte de planificación. Ya ha sido definido en el apartado 2.5 Notación.

Los parámetros definidos dentro de la función son:

- l : número de semanas en el horizonte de planificación. Su valor depende de H de la siguiente forma:

$$l \approx \frac{H}{7}$$

- $dhor_cons$: disponibilidad diaria de cada consulta en horas. Se le ha dado un valor de 8 horas/día.
- $dhor_quir$: disponibilidad diaria de cada quirófano en horas. Se le ha dado un valor de 8 horas/día.
- a : disponibilidad diaria de cada cirujano en horas. Se le ha dado un valor de 8 horas/día.
- ds : disponibilidad semanal de cada cirujano en días. Se le ha dado un valor de 4 días/semana.

7.1.2 Quirófanos y consultas virtuales

Para generar una instancia, la función comienza creando una serie de quirófanos y consultas virtuales para cada unidad, de manera aleatoria, de modo que los pacientes que se crearán a continuación obedecerán a la proporción establecida en este momento.

En primer lugar, se crean los quirófanos virtuales. Se asigna aleatoriamente un número de quirófanos a cada unidad, comprendido entre 1 y un máximo $max_quirvirt$:

$$max_quirvirt \approx 2 \times \frac{n_quir}{n_unidades}$$

Es decir, el máximo es el doble de lo que a cada unidad le correspondería si el total de los quirófanos se repartiera por igual.

Una vez se tienen estos quirófanos virtuales aleatorios, se debe verificar que el total de los quirófanos virtuales coincide con el total real, es decir, n_quir . En caso contrario:

- Mientras que el total de quirófanos virtuales sea mayor que n_quir , se selecciona una unidad al azar y se le quita un quirófano virtual (siempre y cuando no tenga un solo quirófano virtual, ya que ninguna unidad puede quedarse con cero quirófanos).
- Mientras que el total de quirófanos virtuales sea menor que n_quir , se selecciona una unidad al azar y se le da otro quirófano virtual.

Una vez finalizado este proceso, se tienen los quirófanos virtuales definitivos de cada unidad, $quir_virt_u$. Para generar las consultas virtuales de cada unidad, $cons_virt_u$, se realiza una regla de tres para cada unidad de manera que, si para un total de quirófanos n_quir , la unidad u tiene tantos quirófanos virtuales, para un total de consultas n_cons , la unidad u tendrá X consultas virtuales.

De este modo, se obtienen los quirófanos y consultas virtuales de cada unidad, $quir_virt_u$ y $cons_virt_u$. A continuación, se comienzan a generar las intervenciones.

7.1.3 Generación de las intervenciones

Las intervenciones se generan de una en una, hasta llegar a un número total de intervenciones que se va actualizando progresivamente. Antes de empezar con la primera intervención, se definen los valores que se mantienen constantes para todas ellas:

- dur_pae : definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 0.3 y 0.6 horas.
- dur_post : definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 0.3 y 0.6 horas.
- $tresp_pae$: definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 28 y 42 días.
- $tresp_post$: definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 28 y 42 días.

- t_{p_o} : definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 0 y 5 días.
- t_{o_p} : definido en el apartado 2.5 Notación. Tiene un valor aleatorio entre 10 y 20 días.

A continuación, comienza la generación de intervenciones. Esta parte se realiza unidad por unidad. Comienza a llevarse a cabo con la primera intervención de la primera unidad, cuando el número total de pacientes de esa unidad es 0, y prosigue mientras que el número total de pacientes de la unidad sea menor que P_u :

$$P_u \approx \frac{(cons_virt_u \times dhor_cons + quir_virt_u \times dhor_quir) \times beta \times H}{durmedia_u}$$

donde $durmedia_u$ es la duración media de las etapas iniciales de las intervenciones de la unidad u

y P_u se ha definido en el apartado 2.5 Notación.

Analizando la expresión anterior, se verifica que P_u se va actualizando porque $durmedia_u$ cambia a medida que se van generando nuevas intervenciones, con distintas duraciones.

La expresión no es complicada: el primer paréntesis es la disponibilidad diaria de las consultas y quirófanos de la unidad en horas, y al dividirlo entre la duración media de las intervenciones, se obtiene el número de intervenciones que se pueden realizar en un día. Este número se multiplica por el número de días, H . Finalmente, $beta$ representa el factor de saturación de las consultas y quirófanos, y al hacerlo mayor que 1, se obliga al número de pacientes a ser superior al ideal.

Este procedimiento se repite para todas las unidades, de modo que se genera el número adecuado de pacientes, según lo establecido por la proporción de quirófanos y consultas virtuales de cada unidad.

Una vez establecida la unidad a la que se asigna la intervención, se genera el resto de datos de las intervenciones:

- $etapa_inic_j$: definido en el apartado 2.5 Notación. Es un número aleatorio entre 0 y 2.
- hab_resp_j : definido en el apartado 2.5 Notación. Es un número aleatorio entre 2 y 20.
- hab_asist_j : definido en el apartado 2.5 Notación. Es un número aleatorio entre 1 y 20. No puede ser mayor que hab_resp_j .
- dur_or_j : definido en el apartado 2.5 Notación. Es un número pseudoaleatorio con distribución normal de media mu y desviación estándar $sigma$. mu es un número aleatorio entre 1 y 4, y $sigma$ es, aleatoriamente, $0.1 \times mu$, $0.2 \times mu$, $0.3 \times mu$, $0.4 \times mu$ o $0.5 \times mu$. No puede ser mayor que $dhor_quir$.
- $tresp_or_j$: definido en el apartado 2.5 Notación. Es un número aleatorio entre 40 y 300.
- $tiempo_acum_j$: definido en el apartado 2.5 Notación.
 - Si la etapa inicial es PAE ($etapa_inic_j = 0$):
 - Si $tresp_pae \geq 35$, $tiempo_acum_j$ es un número aleatorio entre 0 y 42.
 - Si $tresp_pae < 35$, $tiempo_acum_j$ es un número aleatorio entre 0 y 35.
 - Si la etapa inicial es OR ($etapa_inic_j = 1$):
 - Si $tresp_or_j \geq 250$, $tiempo_acum_j$ es un número aleatorio entre 0 y 300.
 - Si $208 \leq tresp_or_j < 250$, $tiempo_acum_j$ es un número aleatorio entre 0 y 250.
 - Si $173 \leq tresp_or_j < 208$, $tiempo_acum_j$ es un número aleatorio entre 0 y 208.

- Si $144 \leq tresp_or_j < 173$, $tiempo_acum_j$ es un número aleatorio entre 0 y 173.
 - Si $120 \leq tresp_or_j < 144$, $tiempo_acum_j$ es un número aleatorio entre 0 y 144.
 - Si $100 \leq tresp_or_j < 120$, $tiempo_acum_j$ es un número aleatorio entre 0 y 120.
 - Si $83 \leq tresp_or_j < 100$, $tiempo_acum_j$ es un número aleatorio entre 0 y 100.
 - Si $69 \leq tresp_or_j < 83$, $tiempo_acum_j$ es un número aleatorio entre 0 y 83.
 - Si $58 \leq tresp_or_j < 69$, $tiempo_acum_j$ es un número aleatorio entre 0 y 69.
 - Si $48 \leq tresp_or_j < 58$, $tiempo_acum_j$ es un número aleatorio entre 0 y 58.
 - Si $tresp_or_j < 48$, $tiempo_acum_j$ es un número aleatorio entre 0 y 48.
- Si la etapa inicial es POST ($etapa_inic_j = 2$):
 - Si $tresp_post \geq 35$, $tiempo_acum_j$ es un número aleatorio entre 0 y 42.
 - Si $tresp_post < 35$, $tiempo_acum_j$ es un número aleatorio entre 0 y 35.
 - $tresp_pae_rel_j$: definido en el apartado 2.5 Notación.
 - $tresp_or_rel_j$: definido en el apartado 2.5 Notación.
 - $tresp_post_rel_j$: definido en el apartado 2.5 Notación.
 - $t_p_o_rel_j$: definido en el apartado 2.5 Notación.
 - $t_o_p_rel_j$: definido en el apartado 2.5 Notación.

Estos últimos cinco tiempos relativos se calculan a partir de los tiempos genéricos y los tiempos acumulados en la lista de espera como se explica al final del apartado 2.5 Notación.

Cuando ya están generadas todas las intervenciones de todas las unidades, es momento de generar a los cirujanos que se harán cargo de ellas.

7.1.4 Generación de los cirujanos

En primer lugar, se calcula el número total de cirujanos, S :

$$S \approx \frac{(n_{cons} \times dhor_{cons} + n_{quir} \times dhor_{quir} \times 2) \times alfa \times H}{ds \times l \times a}$$

La expresión tiene el siguiente significado: el primer paréntesis representa las horas al día que un cirujano es necesitado en una consulta o quirófano. Las horas de quirófano se multiplican por dos ya que en un quirófano siempre debe haber dos cirujanos: el responsable y el asistente. Sintetizando, el paréntesis podría denominarse “horas de trabajo necesarias al día”. Al multiplicarlo por H , se obtienen las horas de trabajo necesarias en todo el horizonte de planificación. Por otro lado, el denominador no es más que las horas que un cirujano trabaja en todo el horizonte de planificación, es decir, las horas de trabajo realizadas por cirujano en todo el horizonte de planificación. Numerador entre denominador dan lugar al número de cirujanos necesario para ocupar adecuadamente todas las consultas y quirófanos. Sin embargo, este número se multiplica por $alfa$, que se hace mayor que 1 para garantizar que siempre hay más cirujanos de los que se necesitan.

A continuación, se calcula el número de cirujanos necesarios para cada unidad. Para ello, se calcula la duración total de las etapas iniciales de las intervenciones de una unidad concreta, y se dividen entre las horas de trabajo de un cirujano en todo el horizonte de planificación:

$$n_{ciruj_unidad}_u = \frac{sumatiempos_u}{ds \times l \times a}$$

siendo $sumatiempos_u$ la duración total de las etapas iniciales de la unidad u .

Una vez calculado el número de cirujanos por unidad, hay que asegurarse de que el total coincide con S . De no ser así:

- Mientras que $\sum_u n_{ciruj_unidad_u} > S$, se le quita un cirujano a una unidad aleatoria.
- Mientras que $\sum_u n_{ciruj_unidad_u} < S$, se le suma un cirujano a una unidad aleatoria.

Una vez se conoce el número de cirujanos y sus correspondientes unidades, se genera el resto de los datos de cada cirujano:

- $disp_{ciruj}$: lista con tantos elementos como días tenga el horizonte temporal. Cada elemento puede ser 0 o a , y representa las horas de trabajo en cada día del horizonte de planificación. Los 0 significan descanso, y a las horas de trabajo. Cada siete elementos se repite la secuencia, ya que los días de trabajo son los mismos para todas las semanas del horizonte. Para elaborar esta lista de cada cirujano, se ha utilizado ds , que es el número de días a la semana que los cirujanos trabajan.
- hab_{ciruj} : definido en el apartado 2.5 Notación. Es un número aleatorio del 1 al 20. Se procura que haya al menos un cirujano con habilidad suficiente para ser responsable de la intervención con mayor nivel de responsable requerido, y al menos un cirujano, distinto al anterior, con habilidad suficiente para ser asistente de la intervención con mayor nivel de asistente requerido.

Todos estos datos se recogen en archivos TXT, que serán utilizados por el modelo, y en archivos XLSX, que muestran la información más visualmente.

	A	B	C	D	E	F	G	H	I	J
	ID intervención	Unidad intervención	Etapas intervención	GENERAL Tiempo de respuesta PAE	GENERAL Tiempo de respuesta OR	GENERAL Tiempo de respuesta POST	Tiempo acumulado	Tiempo máximo de respuesta PAE	Tiempo máximo de respuesta OR	Tiempo máximo de respuesta POST
1										
2	0	0	0	39	184	30	4	35	184	30
3	1	0	0	39	99	30	2	37	99	30
4	2	0	2	39	208	30	7	39	208	23
5	3	0	2	39	213	30	31	39	213	-1
6	4	0	2	39	43	30	21	39	43	9
7	5	0	0	39	126	30	18	21	126	30
8	6	0	1	39	253	30	5	39	248	30
9	7	0	2	39	78	30	35	39	78	-5
10	8	0	0	39	157	30	1	38	157	30
11	9	0	0	39	64	30	15	24	64	30
12	10	0	1	39	255	30	217	39	38	30
13	11	0	2	39	137	30	0	39	137	30
14	12	0	1	39	66	30	69	39	-3	30
15	13	0	2	39	121	30	23	39	121	7

Figura 7-1. Primera parte de la Hoja de Pacientes de una instancia en formato .xlsx.

	A	K	L	M	N	O	P	Q	R	S
	ID intervención	Duración PAE	Duración OR	Duración POST	Habilidad cirujano responsable	Habilidad d cirujano asistente	GENERAL Tiempo mínimo PAE-OR	GENERAL Tiempo mínimo OR-POST	Tiempo mínimo PAE-OR	Tiempo mínimo OR-POST
1										
2	0	0,35	1,87	0,5	5	3	5	17	5	17
3	1	0,35	2,99	0,5	2	2	5	17	5	17
4	2	0,35	0,73	0,5	5	2	5	17	5	10
5	3	0,35	1,16	0,5	13	1	5	17	5	0
6	4	0,35	1,3	0,5	19	18	5	17	5	0
7	5	0,35	2,27	0,5	8	4	5	17	5	17
8	6	0,35	0,7	0,5	12	2	5	17	0	17
9	7	0,35	1,9	0,5	17	9	5	17	5	0
10	8	0,35	0,79	0,5	6	6	5	17	5	17
11	9	0,35	2,73	0,5	7	4	5	17	5	17
12	10	0,35	2,77	0,5	14	5	5	17	0	17
13	11	0,35	1,99	0,5	6	2	5	17	5	17
14	12	0,35	3,81	0,5	10	9	5	17	0	17
15	13	0,35	1,76	0,5	17	17	5	17	5	0

Figura 7-2. Segunda parte de la Hoja de Pacientes de una instancia en formato .xlsx.

	A	B	C	D
	ID cirujano	Unidad cirujano	Habilidad cirujano	Disponibilidad cirujano
1				
2	0	0	19	[8, 0, 8, 8, 0, 0, 8, 8, 0, 8, 8, 0, 0, 8]
3	1	0	4	[0, 8, 8, 8, 0, 0, 8, 0, 8, 8, 8, 0, 0, 8]
4	2	0	11	[0, 8, 0, 0, 8, 8, 8, 0, 8, 0, 0, 8, 8, 8]
5	3	0	20	[0, 0, 8, 0, 8, 8, 8, 0, 0, 8, 0, 8, 8, 8]
6	4	0	9	[8, 8, 0, 0, 0, 8, 8, 8, 8, 0, 0, 0, 8, 8]
7	5	0	3	[8, 8, 0, 8, 0, 0, 8, 8, 8, 0, 8, 0, 0, 8]
8	6	0	20	[0, 0, 0, 8, 8, 8, 8, 0, 0, 0, 8, 8, 8, 8]
9	7	1	14	[0, 8, 0, 8, 8, 8, 0, 0, 8, 0, 8, 8, 8, 0]
10	8	1	20	[8, 0, 8, 0, 8, 8, 0, 8, 0, 8, 0, 8, 8, 0]
11	9	1	5	[8, 8, 8, 8, 0, 0, 0, 8, 8, 8, 8, 0, 0, 0]
12	10	1	12	[8, 0, 0, 0, 8, 8, 8, 8, 0, 0, 0, 8, 8, 8]
13	11	1	6	[8, 8, 8, 8, 0, 0, 0, 8, 8, 8, 8, 0, 0, 0]
14	12	1	1	[0, 0, 8, 8, 0, 8, 8, 0, 0, 8, 8, 0, 8, 8]
15	13	1	18	[8, 8, 0, 8, 0, 0, 8, 8, 8, 0, 8, 0, 0, 8]
16	14	1	13	[8, 0, 0, 8, 8, 0, 8, 8, 0, 0, 8, 8, 0, 8]
17	15	1	9	[8, 0, 8, 8, 0, 0, 8, 8, 0, 8, 8, 0, 0, 8]

Figura 7-3. Hoja de Cirujanos de una instancia en formato .xlsx.

	A	B	C	D	E	F
	Número de consultas	Número de quirófanos	Número de unidades	Disponibilidad diaria consultas	Disponibilidad diaria quirófanos	Horizonte de planificación
1						
2	8	5	3	8	8	14
3						
4						

Figura 7-4. Hoja de Datos de una instancia en formato .xlsx.

7.1.5 Instancias generadas

Se han generado 40 instancias con los siguientes parámetros:

Tabla 7-1. Instancias generadas.

Tipo de instancia	Instancias	<i>n_unidades</i>	<i>num_cons</i>	<i>num_quir</i>	<i>alfa</i>	<i>beta</i>	<i>H</i>
Instancias medianas	1 - 10	3	8	5	1.5	1.25	14
	11 - 20	5	12	8	1.5	1.25	14
	21 - 30	7	17	10	1.5	1.25	14
Instancias grandes	31 - 35	3	8	5	1.5	1.25	28
	35-40	5	12	8	1.5	1.25	28

A través de los distintos parámetros, se han generado instancias considerablemente diversas. Las instancias medianas se caracterizan por poseer un horizonte de planificación de 14 días, es decir, dos semanas. Entre ellas hay instancias mayores, de hasta siete unidades, y más reducidas, con solo tres. Por otro lado, las instancias grandes tienen un horizonte temporal de 28 días, es decir, cuatro semanas. Se ha generado un menor número de estas instancias por el obstáculo que supone para la simulación el aumento del tiempo de ejecución. Sin embargo, las instancias grandes representan una cuarta parte de las instancias totales, por lo que su presencia se considera suficiente.

7.2 Comparación de Algoritmos Genéticos

Para comparar los Algoritmos Genéticos elaborados, se han realizado numerosas simulaciones. A cada una de las 40 instancias generadas se le han asignado cuatro soluciones iniciales aleatorias con la función “asigna_mat”. Esto se ha hecho para garantizar que el Algoritmo Genético que resulte elegido habrá sido evaluado ante todo tipo de soluciones iniciales, ya sean buenas o malas. Las 40 instancias multiplicadas por 4 soluciones iniciales cada una, dan lugar a 160 variantes distintas de los problemas, para las que se han ejecutado los Algoritmos Genéticos AG1, AG2, AG3 y AG4. Esto es igual a un total de 640 simulaciones.

7.2.1 Resultados obtenidos

Como resultado de las simulaciones, se ha obtenido la mejor secuencia de intervenciones y el mejor valor de la función objetivo que cada algoritmo ha conseguido encontrar. Las secuencias de intervenciones resultantes no se han registrado, ya que son muy extensas, pues el tamaño de las instancias va desde las 1000 intervenciones hasta incluso 5000.

Para analizar los resultados obtenidos y determinar cuál es el Algoritmo Genético más fiable, se ha hecho uso del RPD (*Relative Percentage Deviation*) o Desviación Porcentual Relativa. Esta medida permite calcular el grado, en porcentaje, en que se desvía el valor obtenido por cada algoritmo respecto al mejor valor obtenido por todos ellos.

$$RPD_{inst,mat_{dist},ag} = \frac{|obj_{inst,sol_inic,ag} - min_{inst,sol_inic}|}{|min_{inst,sol_inic}|} \times 100,$$

$inst = 1 \dots 40$

$sol_inic = 1 \dots 4$

$ag = 1, 2, 3, 4$

siendo $inst$ la instancia;

sol_inic la solución inicial, es decir, la distribución de los quirófanos y consultas entre las unidades hospitalarias;

ag el Algoritmo Genético;

$obj_{inst,sol_inic,ag}$ el valor de la función objetivo al aplicar el Algoritmo Genético ag en la instancia $inst$ con la solución inicial sol_inic ;

y min_{inst,sol_inic} el mínimo valor de la función objetivo obtenido al aplicar todos los Algoritmos Genéticos en la instancia $inst$ con la solución inicial sol_inic .

Una vez calculado el RPD en todos los Algoritmos Genéticos para cada una de las 160 variantes de los problemas (entendidas como conjunto de instancia y solución inicial), se calcula el *ARPD* (*Average Relative Percentage Deviation*) o Desviación Porcentual Relativa Media de cada Algoritmo Genético. Aquel que presente un *ARPD* menor, es seleccionado como el mejor Algoritmo Genético.

$$ARPD_{ag} = \frac{\sum_{inst} \sum_{sol_inic} RPD_{inst,sol_inic,ag}}{40 \times 4}$$

En la Tabla 7-2 se muestran tanto los resultados de las simulaciones como los *RPD* y *ARPD* calculados. Por falta de espacio, no se han escrito en la tabla las soluciones iniciales, y solo se ha escrito su índice 1, 2, 3 o 4. En el Excel que contiene los resultados, sí están registradas de la siguiente manera:

	A	B	C	D	E	F
1	Instancia	Solución inicial	Genético 1	Genético 2	Genético 3	Genético 4
2	1	[[4, 2], [2, 1], [2, 2]]	1402	1474	1395	1486
3	1	[[2, 1], [3, 2], [3, 2]]	1230	1221	1220	1212
4	1	[[5, 3], [2, 1], [1, 1]]	1462	1592	1441	1592
5	1	[[3, 2], [1, 1], [4, 2]]	1348	1351	1338	1341
6	2	[[2, 1], [2, 1], [4, 3]]	1481	1490	1479	1504
7	2	[[3, 2], [3, 2], [2, 1]]	1411	1406	1416	1410
8	2	[[2, 1], [1, 1], [5, 3]]	1506	1500	1516	1505
9	2	[[2, 1], [5, 3], [1, 1]]	1865	1879	1827	1874
10	3	[[3, 2], [3, 2], [2, 1]]	1190	1327	1187	1313
11	3	[[4, 2], [2, 2], [2, 1]]	1173	1249	1174	1253
12	3	[[2, 1], [4, 3], [2, 1]]	1449	1698	1442	1934
13	3	[[4, 2], [2, 1], [2, 2]]	1236	1273	1236	1272
14	4	[[1, 1], [4, 2], [3, 2]]	746	802	746	810
15	4	[[2, 1], [1, 1], [5, 3]]	722	720	721	728
16	4	[[2, 1], [3, 2], [3, 2]]	746	766	746	759
17	4	[[2, 2], [4, 2], [2, 1]]	793	1024	793	1041
18	5	[[2, 1], [3, 2], [3, 2]]	1162	1165	1162	1165

Figura 7-5. Excel con los resultados obtenidos.

Tabla 7-2. Comparación de los Algoritmos Genéticos.

Inst.	Sol. Inic.	Obj. AG1	Obj. AG2	Obj. AG3	Obj. AG4	Mín.	RPD AG1	RPD AG2	RPD AG3	RPD AG4
1	1	1402	1474	1395	1486	1395	0,50%	5,66%	0,00%	6,52%
	2	1230	1221	1220	1212	1212	1,49%	0,74%	0,66%	0,00%
	3	1462	1592	1441	1592	1441	1,46%	10,48%	0,00%	10,48%
	4	1348	1351	1338	1341	1338	0,75%	0,97%	0,00%	0,22%
2	1	1481	1490	1479	1504	1479	0,14%	0,74%	0,00%	1,69%
	2	1411	1406	1416	1410	1406	0,36%	0,00%	0,71%	0,28%
	3	1506	1500	1516	1505	1500	0,40%	0,00%	1,07%	0,33%
	4	1865	1879	1827	1874	1827	2,08%	2,85%	0,00%	2,57%
3	1	1190	1327	1187	1313	1187	0,25%	11,79%	0,00%	10,61%
	2	1173	1249	1174	1253	1173	0,00%	6,48%	0,09%	6,82%
	3	1449	1698	1442	1934	1442	0,49%	17,75%	0,00%	34,12%
	4	1236	1273	1236	1272	1236	0,00%	2,99%	0,00%	2,91%
4	1	746	802	746	810	746	0,00%	7,51%	0,00%	8,58%
	2	722	720	721	728	720	0,28%	0,00%	0,14%	1,11%
	3	746	766	746	759	746	0,00%	2,68%	0,00%	1,74%
	4	793	1024	793	1041	793	0,00%	29,13%	0,00%	31,27%
5	1	1162	1165	1162	1165	1162	0,00%	0,26%	0,00%	0,26%
	2	1298	1472	1293	1475	1293	0,39%	13,84%	0,00%	14,08%
	3	1197	1197	1198	1198	1197	0,00%	0,00%	0,08%	0,08%
	4	1097	1095	1092	1092	1092	0,46%	0,27%	0,00%	0,00%
6	1	977	1509	999	1491	977	0,00%	54,45%	2,25%	52,61%
	2	935	1023	932	1025	932	0,32%	9,76%	0,00%	9,98%
	3	1087	1724	1082	1727	1082	0,46%	59,33%	0,00%	59,61%
	4	1078	1724	1073	1741	1073	0,47%	60,67%	0,00%	62,26%
7	1	946	1028	946	1037	946	0,00%	8,67%	0,00%	9,62%
	2	888	1075	892	1054	888	0,00%	21,06%	0,45%	18,69%
	3	1019	1245	994	1234	994	2,52%	25,25%	0,00%	24,14%
	4	929	939	933	940	929	0,00%	1,08%	0,43%	1,18%
8	1	674	917	674	921	674	0,00%	36,05%	0,00%	36,65%
	2	729	977	739	972	729	0,00%	34,02%	1,37%	33,33%
	3	737	742	731	718	718	2,65%	3,34%	1,81%	0,00%
	4	692	699	697	682	682	1,47%	2,49%	2,20%	0,00%
9	1	1187	1234	1181	1220	1181	0,51%	4,49%	0,00%	3,30%
	2	1332	1452	1333	1455	1332	0,00%	9,01%	0,08%	9,23%
	3	1367	1398	1376	1419	1367	0,00%	2,27%	0,66%	3,80%
	4	1347	1581	1356	1587	1347	0,00%	17,37%	0,67%	17,82%
10	1	1308	1313	1311	1307	1307	0,08%	0,46%	0,31%	0,00%
	2	1172	1173	1171	1171	1171	0,09%	0,17%	0,00%	0,00%
	3	1163	1162	1161	1162	1161	0,17%	0,09%	0,00%	0,09%
	4	1490	1705	1494	1710	1490	0,00%	14,43%	0,27%	14,77%
11	1	2260	2247	2246	2260	2246	0,62%	0,04%	0,00%	0,62%
	2	2052	2868	2035	2925	2035	0,84%	40,93%	0,00%	43,73%

	3	2185	3173	2196	3173	2185	0,00%	45,22%	0,50%	45,22%
	4	1921	2449	1931	2444	1921	0,00%	27,49%	0,52%	27,23%
12	1	2045	2370	2084	2384	2045	0,00%	15,89%	1,91%	16,58%
	2	2100	2356	2060	2322	2060	1,94%	14,37%	0,00%	12,72%
	3	1878	1877	1880	1868	1868	0,54%	0,48%	0,64%	0,00%
	4	2088	2262	2118	2318	2088	0,00%	8,33%	1,44%	11,02%
13	1	1971	1980	1981	1980	1971	0,00%	0,46%	0,51%	0,46%
	2	1950	1958	1955	1957	1950	0,00%	0,41%	0,26%	0,36%
	3	2018	2017	2019	2017	2017	0,05%	0,00%	0,10%	0,00%
	4	2366	2369	2370	2348	2348	0,77%	0,89%	0,94%	0,00%
14	1	1716	1811	1693	1825	1693	1,36%	6,97%	0,00%	7,80%
	2	1771	1766	1771	1774	1766	0,28%	0,00%	0,28%	0,45%
	3	1865	1808	1861	1767	1767	5,55%	2,32%	5,32%	0,00%
	4	1785	1944	1773	1939	1773	0,68%	9,64%	0,00%	9,36%
15	1	1545	1567	1544	1562	1544	0,06%	1,49%	0,00%	1,17%
	2	1503	1476	1484	1470	1470	2,24%	0,41%	0,95%	0,00%
	3	1406	1562	1401	1579	1401	0,36%	11,49%	0,00%	12,71%
	4	1451	1730	1442	1733	1442	0,62%	19,97%	0,00%	20,18%
16	1	2495	2467	2439	2475	2439	2,30%	1,15%	0,00%	1,48%
	2	3031	2953	2934	2985	2934	3,31%	0,65%	0,00%	1,74%
	3	2377	2369	2371	2379	2369	0,34%	0,00%	0,08%	0,42%
	4	2477	3225	2507	3219	2477	0,00%	30,20%	1,21%	29,96%
17	1	1924	2402	1897	2423	1897	1,42%	26,62%	0,00%	27,73%
	2	1845	1849	1874	1920	1845	0,00%	0,22%	1,57%	4,07%
	3	1814	2287	1787	2330	1787	1,51%	27,98%	0,00%	30,39%
	4	1873	2429	1820	2401	1820	2,91%	33,46%	0,00%	31,92%
18	1	2475	2838	2335	2878	2335	6,00%	21,54%	0,00%	23,25%
	2	2326	2642	2316	2585	2316	0,43%	14,08%	0,00%	11,61%
	3	2466	2884	2391	2908	2391	3,14%	20,62%	0,00%	21,62%
	4	2415	2691	2384	2758	2384	1,30%	12,88%	0,00%	15,69%
19	1	2674	2907	2658	2907	2658	0,60%	9,37%	0,00%	9,37%
	2	2745	2749	2744	2727	2727	0,66%	0,81%	0,62%	0,00%
	3	3006	3590	2983	3614	2983	0,77%	20,35%	0,00%	21,15%
	4	2660	2957	2686	2964	2660	0,00%	11,17%	0,98%	11,43%
20	1	3284	3280	3296	3291	3280	0,12%	0,00%	0,49%	0,34%
	2	2728	2728	2727	2728	2727	0,04%	0,04%	0,00%	0,04%
	3	3285	3374	3369	3377	3285	0,00%	2,71%	2,56%	2,80%
	4	2643	2642	2640	2637	2637	0,23%	0,19%	0,11%	0,00%
21	1	3449	3560	3448	3583	3448	0,03%	3,25%	0,00%	3,92%
	2	3395	3467	3399	3461	3395	0,00%	2,12%	0,12%	1,94%
	3	3381	3335	3226	3337	3226	4,80%	3,38%	0,00%	3,44%
	4	2908	2919	2920	2917	2908	0,00%	0,38%	0,41%	0,31%
22	1	1874	1998	1871	2009	1871	0,16%	6,79%	0,00%	7,38%
	2	1710	2030	1709	2041	1709	0,06%	18,78%	0,00%	19,43%
	3	1689	1679	1676	1676	1676	0,78%	0,18%	0,00%	0,00%

	4	1698	1969	1685	1966	1685	0,77%	16,85%	0,00%	16,68%
23	1	2320	2714	2307	2678	2307	0,56%	17,64%	0,00%	16,08%
	2	1956	1939	1973	1943	1939	0,88%	0,00%	1,75%	0,21%
	3	1740	1761	1758	1774	1740	0,00%	1,21%	1,03%	1,95%
	4	1881	1875	1866	1879	1866	0,80%	0,48%	0,00%	0,70%
24	1	2915	3012	2826	2705	2705	7,76%	11,35%	4,47%	0,00%
	2	2742	2739	2743	2742	2739	0,11%	0,00%	0,15%	0,11%
	3	2983	3029	2971	2980	2971	0,40%	1,95%	0,00%	0,30%
	4	3292	3331	3288	3370	3288	0,12%	1,31%	0,00%	2,49%
25	1	1695	1695	1701	1712	1695	0,00%	0,00%	0,35%	1,00%
	2	1654	1658	1672	1641	1641	0,79%	1,04%	1,89%	0,00%
	3	1787	1999	1804	1966	1787	0,00%	11,86%	0,95%	10,02%
	4	1867	2134	1866	2168	1866	0,05%	14,36%	0,00%	16,18%
26	1	1516	2041	1491	2043	1491	1,68%	36,89%	0,00%	37,02%
	2	1392	1685	1388	1652	1388	0,29%	21,40%	0,00%	19,02%
	3	1495	1511	1499	1512	1495	0,00%	1,07%	0,27%	1,14%
	4	1657	1870	1661	1859	1657	0,00%	12,85%	0,24%	12,19%
27	1	1975	1971	1968	1968	1968	0,36%	0,15%	0,00%	0,00%
	2	2337	2546	2329	2535	2329	0,34%	9,32%	0,00%	8,84%
	3	2125	2264	2111	2266	2111	0,66%	7,25%	0,00%	7,34%
	4	2162	2400	2155	2412	2155	0,32%	11,37%	0,00%	11,93%
28	1	2221	2190	2226	2261	2190	1,42%	0,00%	1,64%	3,24%
	2	2079	2043	2071	2019	2019	2,97%	1,19%	2,58%	0,00%
	3	2070	2142	2087	2171	2070	0,00%	3,48%	0,82%	4,88%
	4	2253	2080	2194	2135	2080	8,32%	0,00%	5,48%	2,64%
29	1	3910	4410	3918	4393	3910	0,00%	12,79%	0,20%	12,35%
	2	3715	3815	3707	3823	3707	0,22%	2,91%	0,00%	3,13%
	3	3572	3743	3626	3744	3572	0,00%	4,79%	1,51%	4,82%
	4	3868	4342	3816	4340	3816	1,36%	13,78%	0,00%	13,73%
30	1	1968	2241	1971	2229	1968	0,00%	13,87%	0,15%	13,26%
	2	1867	2008	1867	2002	1867	0,00%	7,55%	0,00%	7,23%
	3	2059	2518	2068	2530	2059	0,00%	22,29%	0,44%	22,88%
	4	1815	2214	1818	2202	1815	0,00%	21,98%	0,17%	21,32%
31	1	3293	3302	3296	3287	3287	0,18%	0,46%	0,27%	0,00%
	2	3613	3609	3591	3584	3584	0,81%	0,70%	0,20%	0,00%
	3	4252	4797	4220	4840	4220	0,76%	13,67%	0,00%	14,69%
	4	4161	4703	4232	4717	4161	0,00%	13,03%	1,71%	13,36%
32	1	4327	4339	4382	4273	4273	1,26%	1,54%	2,55%	0,00%
	2	4057	4054	4000	4068	4000	1,43%	1,35%	0,00%	1,70%
	3	5607	5605	5582	5596	5582	0,45%	0,41%	0,00%	0,25%
	4	5868	6218	5943	6228	5868	0,00%	5,96%	1,28%	6,13%
33	1	4804	5012	4769	5064	4769	0,73%	5,10%	0,00%	6,19%
	2	3896	3905	3937	3931	3896	0,00%	0,23%	1,05%	0,90%
	3	4976	5009	4951	5036	4951	0,50%	1,17%	0,00%	1,72%
	4	4707	5226	4663	5263	4663	0,94%	12,07%	0,00%	12,87%
34	1	6811	7593	6699	7590	6699	1,67%	13,35%	0,00%	13,30%

	2	8098	8084	8094	8033	8033	0,81%	0,63%	0,76%	0,00%
	3	5802	5834	5877	5823	5802	0,00%	0,55%	1,29%	0,36%
	4	7271	7451	7660	7516	7271	0,00%	2,48%	5,35%	3,37%
35	1	6819	10731	6854	10774	6819	0,00%	57,37%	0,51%	58,00%
	2	6016	5838	6426	6452	5838	3,05%	0,00%	10,07%	10,52%
	3	5309	5314	4996	5313	4996	6,27%	6,37%	0,00%	6,35%
	4	5638	6888	5673	6905	5638	0,00%	22,17%	0,62%	22,47%
36	1	7461	7529	7462	7533	7461	0,00%	0,91%	0,01%	0,97%
	2	10318	10304	10197	10345	10197	1,19%	1,05%	0,00%	1,45%
	3	7378	8333	7390	8342	7378	0,00%	12,94%	0,16%	13,07%
	4	7373	7680	7427	7721	7373	0,00%	4,16%	0,73%	4,72%
37	1	10456	10450	10466	10352	10352	1,00%	0,95%	1,10%	0,00%
	2	10548	12076	10287	11994	10287	2,54%	17,39%	0,00%	16,59%
	3	7754	7742	7735	7742	7735	0,25%	0,09%	0,00%	0,09%
	4	10982	10983	10638	10942	10638	3,23%	3,24%	0,00%	2,86%
38	1	7289	7330	7280	7320	7280	0,12%	0,69%	0,00%	0,55%
	2	5870	6406	6042	6456	5870	0,00%	9,13%	2,93%	9,98%
	3	5791	5783	5856	5834	5783	0,14%	0,00%	1,26%	0,88%
	4	7212	8925	7394	9287	7212	0,00%	23,75%	2,52%	28,77%
39	1	7973	10508	8113	10496	7973	0,00%	31,79%	1,76%	31,64%
	2	8732	8487	8717	10117	8487	2,89%	0,00%	2,71%	19,21%
	3	9025	9017	9069	8999	8999	0,29%	0,20%	0,78%	0,00%
	4	8817	11464	9732	11608	8817	0,00%	30,02%	10,38%	31,65%
40	1	6353	6377	6384	6362	6353	0,00%	0,38%	0,49%	0,14%
	2	7190	7254	7157	7180	7157	0,46%	1,36%	0,00%	0,32%
	3	7303	7322	7243	7307	7243	0,83%	1,09%	0,00%	0,88%
	4	8508	8491	8430	8536	8430	0,93%	0,72%	0,00%	1,26%
ARPD:							0,80%	9,60%	0,68%	9,95%

Tabla 7-2. Comparación de los Algoritmos Genéticos.

Para cada solución inicial de cada instancia, se ha subrayado en verde el valor mínimo obtenido. Además, se han subrayado en naranja los valores de *RPD* mayores que 10%, ya que reflejan una desviación considerable respecto al resultado mínimo obtenido.

Estos resultados ponen de manifiesto que los Algoritmos Genéticos 1 y 3 son mejores que los Algoritmos Genéticos 2 y 4. Estos últimos han obtenido un *ARPD* de 9.60% y 9.95% respectivamente, es decir, una desviación media con respecto al mínimo de casi el 10% en ambos Algoritmos, que puede considerarse bastante alta. Los Algoritmos Genéticos 1 y 3, por su lado, han obtenido un *ARPD* de 0.80% y 0.68%, que son muy buenos resultados en ambos casos.

El Algoritmo seleccionado como el mejor, es el AG3. Además de haber obtenido el menor *ARPD*, es el Algoritmo que más veces ha conseguido el valor mínimo: en 79 ocasiones, un 49.38% del total. Por su parte, el AG1, que también ha obtenido buenos resultados, ha alcanzado el mínimo 56 veces, es decir, un 35% del total de ocasiones. Por último, el AG4 ha alcanzado el mínimo 24 veces, y el AG2 solo en 16 ocasiones.

Conocidas las especificaciones de todos los Algoritmos Genéticos, pueden sacarse conclusiones muy interesantes sobre estas metaheurísticas.

7.2.1.1 Influencia del grado de variedad de la población inicial

Los dos mejores Algoritmos Genéticos, AG1 y AG3, han sido los que cuentan con una mayor variedad de

individuos en su población inicial, ya que incluyen las cinco Reglas de Despacho elaboradas para este problema. Tanto el AG2 como el AG4, por el contrario, basan su población inicial en una única Regla de Despacho: la 5. Estos dos últimos han obtenido unos resultados claramente peores a los del AG1 y el AG3. El AG2 ha obtenido un *RPD* de más del 10% el 35.6% de las veces (en 57 ocasiones), de más del 20% el 16.9% de las veces (en 27 ocasiones), y de más del 30% el 8.1% de las veces (en 13 ocasiones). Por su parte, el AG4 ha obtenido un *RPD* de más del 10% en el 36.9% de los casos (59 veces), de más del 20% en el 16.9% de los casos (27 veces), y de más del 30% en el 9.4% de los casos (15 veces).

Sería un error pensar que el peor desempeño del AG2 y el AG4 se debe a que parten de una Regla de Despacho ineficiente, o a que los otros dos Algoritmos cuentan con Reglas de Despacho mejores. Esto no es así, ya que, como reflejan los resultados recogidos en la Tabla 7-3, la Regla de Despacho 5 es la que obtiene un *ARPD* menor, por lo que puede considerarse la mejor. Para elaborar esta tabla, se han evaluado las cinco Reglas de Despacho en las 40 instancias, pero solo con una solución inicial en cada una de ellas.

Tabla 7-3. Comparación de las Reglas de Despacho.

Inst.	Obj. RD1	Obj. RD2	Obj. RD3	Obj. RD4	Obj. RD5	Mín.	<i>RPD</i> RD1	<i>RPD</i> RD2	<i>RPD</i> RD3	<i>RPD</i> RD4	<i>RPD</i> RD5
1	1288	1337	1589	1345	1332	1288	0,00%	3,80%	23,37%	4,43%	3,42%
2	1576	1671	1603	1407	1396	1396	12,89%	19,70%	14,83%	0,79%	0,00%
3	1236	1292	1562	1295	1276	1236	0,00%	4,53%	26,38%	4,77%	3,24%
4	741	863	1098	835	823	741	0,00%	16,46%	48,18%	12,69%	11,07%
5	1279	1420	1473	1200	1198	1198	6,76%	18,53%	22,95%	0,17%	0,00%
6	1059	1799	1602	1762	1756	1059	0,00%	69,88%	51,27%	66,38%	65,82%
7	842	960	1111	993	955	842	0,00%	14,01%	31,95%	17,93%	13,42%
8	615	838	933	844	838	615	0,00%	36,26%	51,71%	37,24%	36,26%
9	1486	1388	1420	1332	1334	1332	11,56%	4,20%	6,61%	0,00%	0,15%
10	2673	1748	1913	1768	1735	1735	54,06%	0,75%	10,26%	1,90%	0,00%
11	3051	2207	2391	2191	2169	2169	40,66%	1,75%	10,24%	1,01%	0,00%
12	1943	2390	2371	2336	2331	1943	0,00%	23,01%	22,03%	20,23%	19,97%
13	3016	2208	2353	2090	2084	2084	44,72%	5,95%	12,91%	0,29%	0,00%
14	1714	1725	1974	1736	1716	1714	0,00%	0,64%	15,17%	1,28%	0,12%
15	1411	1664	1748	1607	1587	1411	0,00%	17,93%	23,88%	13,89%	12,47%
16	2213	2457	2822	2516	2439	2213	0,00%	11,03%	27,52%	13,69%	10,21%
17	2282	2148	2446	2157	2210	2148	6,24%	0,00%	13,87%	0,42%	2,89%
18	2271	2783	2898	2552	2532	2271	0,00%	22,55%	27,61%	12,37%	11,49%
19	2675	3039	3328	3008	3028	2675	0,00%	13,61%	24,41%	12,45%	13,20%
20	3382	3688	3636	3381	3361	3361	0,62%	9,73%	8,18%	0,60%	0,00%
21	3172	3525	3741	3287	3270	3172	0,00%	11,13%	17,94%	3,63%	3,09%
22	1819	2349	2519	2353	2350	1819	0,00%	29,14%	38,48%	29,36%	29,19%
23	2156	2391	2468	2067	2018	2018	6,84%	18,48%	22,30%	2,43%	0,00%
24	2904	3366	3689	3334	3306	2904	0,00%	15,91%	27,03%	14,81%	13,84%
25	1836	1975	2069	1696	1656	1656	10,87%	19,26%	24,94%	2,42%	0,00%
26	1557	1898	2058	1855	1824	1557	0,00%	21,90%	32,18%	19,14%	17,15%
27	2177	2260	2497	2190	2151	2151	1,21%	5,07%	16,09%	1,81%	0,00%
28	2041	2251	2496	2226	2177	2041	0,00%	10,29%	22,29%	9,06%	6,66%
29	3804	4329	4563	4290	4288	3804	0,00%	13,80%	19,95%	12,78%	12,72%
30	2025	2393	2606	2154	2122	2025	0,00%	18,17%	28,69%	6,37%	4,79%

31	7223	4143	4922	3636	3642	3636	98,65%	13,94%	35,37%	0,00%	0,17%	
32	6677	7749	9753	7509	7697	6677	0,00%	16,06%	46,07%	12,46%	15,28%	
33	4822	5872	6696	5187	5152	4822	0,00%	21,78%	38,86%	7,57%	6,84%	
34	11663	9424	10142	8497	8390	8390	39,01%	12,32%	20,88%	1,28%	0,00%	
35	4972	6596	7644	5246	5130	4972	0,00%	32,66%	53,74%	5,51%	3,18%	
36	7123	8226	10351	6156	6025	6025	18,22%	36,53%	71,80%	2,17%	0,00%	
37	16843	12669	11686	11038	10978	10978	53,43%	15,40%	6,45%	0,55%	0,00%	
38	14956	7769	7954	7226	7184	7184	108,18%	8,14%	10,72%	0,58%	0,00%	
39	10709	12354	10530	11019	10973	10530	1,70%	17,32%	0,00%	4,64%	4,21%	
40	8298	10776	10756	10290	10262	8298	0,00%	29,86%	29,62%	24,01%	23,67%	
							ARPD	12,89%	16,54%	25,92%	9,58%	8,61%

Sabiendo que los peores Algoritmos Genéticos son los que parten únicamente de la mejor Regla de Despacho (además de sus vecinos o secuencias aleatorias), se deduce que, en el problema que se está abordando, es más eficiente una población inicial variada, aunque eso signifique una menor calidad al inicio de la búsqueda de soluciones. Esta afirmación, sin embargo, no puede llevarse al extremo, como se verá a continuación.

7.2.1.2 Influencia de las secuencias aleatorias en la población inicial

El AG2 y el AG4 tienen una diferencia importante en sus poblaciones iniciales: mientras que el AG4 solo incluye la Regla de Despacho 5 y 14 vecinos de la secuencia obtenida con esa regla, el AG2 incluye la Regla de Despacho 5, 10 vecinos de la secuencia obtenida con esa regla y 4 secuencias aleatorias.

La introducción de secuencias aleatorias puede resultar muy útil en problemas de dimensiones pequeñas, sin embargo, en un problema de tanta complejidad como este, la probabilidad de que una secuencia aleatoria acerque al algoritmo a un resultado satisfactorio es muy pequeña. De ahí la importancia de las reglas de despacho elaboradas.

Para poner número concretos, solo en cuatro ocasiones el *RPD* del AG2 mejoró en más de un 5% al *RPD* del AG4, es decir, solo en el 2.5% de las simulaciones puede deducirse que las secuencias aleatorias del AG2 tuvieron un papel significativo en el resultado encontrado. De estas cuatro veces, en dos ocasiones halló el AG2 el valor mínimo de la función objetivo.

Una comparación similar puede realizarse entre el AG1 y el AG3. Ambos Algoritmos incluyen en sus poblaciones iniciales las cinco reglas de despacho, pero mientras que el AG3 incluye 10 vecinos de la secuencia obtenida con la Regla 2, el AG1 incluye 10 secuencias aleatorias. Solo en tres ocasiones el *RPD* del AG1 mejoró al del AG3 en más de un 5%, es decir, solo en el 1.88% de las simulaciones tuvieron un efecto relevante las secuencias aleatorias del AG1. De estas tres veces, el AG1 encontró el mejor valor de la función objetivo, venciendo al resto de algoritmos, en dos ocasiones.

Aunque en un problema tan vasto como es la secuenciación de miles de intervenciones, las secuencias aleatorias no son de gran ayuda, hay que hacer hincapié en que su uso puede ser decisivo en problemas de menor tamaño, como en la búsqueda de una solución inicial satisfactoria. Como se vio en el apartado 6.7 *Algoritmo extendido*, las soluciones iniciales aleatorias tienen un papel fundamental en el desarrollo del algoritmo.

7.2.1.3 Influencia de la solución inicial

En la Tabla 7-2, se han comparado los distintos Algoritmos Genéticos, y se ha llegado a la conclusión de que el mejor es el AG3. Sin embargo, esta tabla puede mirarse desde otra perspectiva e inferir más información interesante. En la Tabla 7-4 se ha extraído parte de la Tabla 7-2, solo con una instancia, y esta vez con las soluciones iniciales, en lugar de sus índices.

Tabla 7-4. Resultados de una instancia.

Inst.	Solución inicial	Obj. AG1	Obj. AG2	Obj. AG3	Obj. AG4
9	1) [[1, 1], [4, 2], [3, 2]]	1187	1234	1181	1220
9	2) [[2, 1], [3, 2], [3, 2]]	1332	1452	1333	1455
9	3) [[3, 2], [2, 1], [3, 2]]	1367	1398	1376	1419
9	4) [[2, 1], [2, 2], [4, 2]]	1347	1581	1356	1587

El mejor resultado para toda la instancia lo obtiene el AG3 para la solución inicial 1) [[1, 1], [4, 2], [3, 2]]. Para esta matriz, los resultados que ofrecen el resto de los Algoritmos son peores. Sin embargo, es importante percatarse de que más valdría utilizar el peor Algoritmo Genético para esta solución inicial, que el mejor, el AG3, para cualquier otra solución inicial.

La influencia de la solución inicial, o sea, la distribución de consultas y quirófanos a las distintas unidades hospitalarias, en el resultado, es altamente superior a la influencia de los Algoritmos Genéticos una vez fijada esta.

Para la solución inicial 1, los resultados varían entre 1181 y 1234. Es decir, el peor resultado es un 4.49% mayor que el mejor. Para la solución inicial 2, los resultados varían entre 1332 y 1455, o sea que el peor resultado es un 9.23% mayor que el mejor. Para la solución inicial 3, los resultados van desde 1367 hasta 1419, es decir, el peor resultado es un 3.80% mayor que el mejor. Por último, para la solución inicial 4, los resultados abarcan desde 1347 hasta 1587, o sea que el peor resultado es un 17.82% mayor que el mejor. Realizando la media de estos porcentajes, se obtiene una diferencia entre el mejor y el peor resultado obtenido al variar los Algoritmos de 8.86%.

Si se realiza este análisis fijando cada Algoritmo Genético, en lugar de la solución inicial, el resultado cambia. Para el AG1, los resultados varían entre 1187 y 1367, es decir, el peor resultado es un 15.16% mayor que el mejor. Para el AG2, los resultados van desde 1234 hasta 1581, o sea que el peor resultado es un 28.12% mayor que el mejor. Para el AG3, los resultados abarcan desde 1181 hasta 1376, lo que significa que el peor resultado es un 16.51% mayor que el mejor. Por último, para el AG4, los resultados varían entre 1220 y 1587, por lo que el peor resultado es un 30.08% mayor. Al realizar la media aritmética de estos porcentajes, se obtiene que, al variar la matriz de distribución, los resultados cambian en un 22.47%. Este dato es significativamente mayor al 8.86% obtenido al variar los Algoritmos Genéticos.

Como conclusión, si bien utilizar el mejor Algoritmo Genético elaborado será de gran ayuda para obtener un buen resultado, es necesario hallar una buena solución inicial, ya que, de lo contrario, la efectividad del Algoritmo queda ensombrecida por la incapacidad de usar adecuadamente los recursos del problema.

Para continuar demostrando la relevancia de la solución inicial, se procede ahora a comparar, para cada instancia, el mejor valor de la función objetivo obtenido en el apartado 6.7 *Algoritmo Extendido*, al utilizar una solución inicial adecuada, con la media de los valores obtenidos con el AG3 en la comparación de algoritmos, donde se han utilizado cuatro soluciones iniciales aleatorias.

Tabla 7-5. Comparación de los valores de la función objetivo obtenidos al partir de una solución inicial satisfactoria o de una solución inicial aleatoria.

Instancia	Valor de la función objetivo al utilizar una solución inicial satisfactoria (Algoritmo Extendido)	Media de los valores de la función objetivo al utilizar cuatro soluciones iniciales aleatorias	Incremento del valor de la función objetivo

1	1144	1348.5	17.88%
2	1334	1559.5	16.90%
3	1113	1259	13.12%
4	608	751.5	23.60%
5	861	1186.25	37.76%
6	578	1021.5	76.73%
7	835	941.25	12.72%
8	609	710.25	16.63%
9	1083	1311.5	21.10%
10	1161	1284.25	10.62%
11	1725	2102	21.86%
12	1758	2035.5	15.78%
13	1881	2081.25	10.65%
14	1568	1774.5	13.17%
15	1270	1467.75	15.57%
16	1977	2562.75	29.63%
17	1579	1844.5	16.81%
18	2120	2356.5	11.16%
19	2366	2767.75	16.98%
20	2604	3008	15.51%
21	2792	3248.25	16.34%
22	1483	1735.25	17.01%
23	1617	1976	22.20%
24	2510	2957	17.81%
25	1528	1760.75	15.23%
26	1332	1509.75	13.34%
27	1834	2140.75	16.73%
28	1846	2144.5	16.17%

29	3353	3766.75	12.34%
30	1796	1931	7.52%
31	2915	3834.75	31.55%
32	3357	4976.75	48.25%
33	3192	4580	43.48%
34	3989	7082.5	77.55%
35	4097	5987.25	46.14%
36	4632	8119	75.28%
37	7019	9781.5	39.36%
38	3885	6643	70.99%
39	5192	8907.75	71.57%
40	4831	7303.5	51.18%
		Incremento medio:	28.11%

El incremento medio del 28.11% del valor de la función objetivo, pone de manifiesto que, ciertamente, el uso de una buena solución inicial supone una diferencia importante en el resultado obtenido. Se puede observar que las instancias con un mayor incremento porcentual son las que se definieron como grandes (de la 31 a la 40). Estas instancias presentan un incremento medio del valor de la función objetivo del 55.54%, mientras que las instancias medianas (de la 1 a la 30) tienen un incremento medio del 18.92%.

Esto se debe a que, como ya se explicó, las instancias medianas tienen un horizonte de planificación de dos semanas, mientras que las instancias grandes tienen uno de cuatro semanas. Por tanto, en este último tipo de instancias, la distribución inadecuada de los quirófanos y las consultas entre las distintas unidades, implica la imposibilidad de hacer frente correctamente a las intervenciones programadas durante cuatro semanas, y el consecuente aumento de la función objetivo durante ese periodo. Esto, por otro lado, significa que la mejoría que supone una solución inicial adecuada, puede verse reflejada en los resultados durante cuatro semanas, de ahí que el incremento del valor de la función objetivo sea mayor que en las instancias medianas.

Por tanto, se puede concluir que, si bien la utilización de una solución inicial satisfactoria siempre mejorará visiblemente los resultados, cuanto mayor sea el horizonte de planificación, más evidente será esa mejora, y más necesario será hallar una buena solución inicial.

8 CONCLUSIONES

El presente trabajo aborda la programación y planificación de un hospital de referencia real, con el fin de minimizar los retrasos totales de la atención a cada uno de los pacientes. De este modo, se garantiza el aumento, no solo de la calidad de vida de cada paciente, que es el objetivo principal, sino también de su grado de satisfacción respecto al cuidado recibido y de la confianza que depositará en el sistema en ocasiones futuras.

La programación del hospital en el corto plazo implica establecer las fechas precisas en que cada paciente será atendido en cada etapa, así como el cirujano (o los cirujanos) que llevará a cabo esta consulta o intervención, y la consulta médica o quirófano en que se realizará. Para afrontar la programación de las intervenciones, se han elaborado cuatro metaheurísticas que, a través del característico proceso iterativo, hallan una secuencia de intervenciones que da lugar a un valor de la función objetivo aceptable.

Para afrontar la planificación de los recursos del hospital en el medio plazo, se ha elaborado un Algoritmo Extendido que permite hallar una distribución satisfactoria de los quirófanos y las consultas entre las unidades hospitalarias. Este algoritmo parte de una serie de soluciones iniciales aleatorias y, tras analizar los resultados de la función objetivo y elegir la mejor, busca mejorar dicha solución explorando vecindades cercanas en el espacio de soluciones. Este procedimiento, similar al de un algoritmo de búsqueda local, se repite en dos iteraciones.

Resulta muy interesante poder aplicar los conocimientos sobre sistemas de producción al entorno sanitario, ya que la salud es primordial y cualquier avance o mejora en su tratamiento supone un enorme progreso para toda la sociedad.

9 LÍNEAS DE FUTURA INVESTIGACIÓN

Sería muy interesante continuar la investigación sobre la aplicación de la teoría de sistemas productivos en el entorno sanitario, ya que tiene mucho potencial por la semejanza de ambos entornos.

En primer lugar, se podrían añadir más restricciones al problema que permitieran equiparlo aún más a la realidad. Por ejemplo, podrían añadirse condicionantes para decidir qué cirujanos realizan las consultas PAE o POST.

Por otro lado, podría profundizarse en la programación de la producción teniendo en cuenta, además del proceso médico-quirúrgico PAE-OR-POST, las consultas genéricas. En este proyecto, las consultas genéricas se han tenido en cuenta únicamente a la hora de reducir la disponibilidad de cada cirujano, que en lugar de trabajar cinco días a la semana, se considera que trabajan cuatro.

Además, resultaría interesante ampliar la búsqueda de soluciones iniciales. Cambiando los criterios de terminación del Algoritmo Extendido, que actualmente solo itera dos veces, y algunas especificaciones más, podría elaborarse un algoritmo de búsqueda local que hallara soluciones iniciales mejores, y en consecuencia, permitiera mejorar la función objetivo.

Por otra parte, para programar las intervenciones se ha utilizado un sistema *open-scheduling*, según el cual un cirujano asignado a consultas un día, puede atender a muchos pacientes en consultas distintas. Del mismo modo, un cirujano que está asignado a quirófanos un día, puede atender a muchos pacientes en distintos quirófanos. Esta forma de programación puede resultar desorganizada y quizá despertar recelo en los cirujanos, y habría que comprobar si realmente resulta eficiente. Podría elaborarse, sin embargo, un modelo del problema para el cual el cirujano asignado a consultas un día, está asignado a una consulta en concreto. En cuanto a los quirófanos, es más complicado establecer este concepto de permanencia, ya que cada intervención requiere un nivel de habilidad para responsable y asistente distinto, y, por tanto, una pareja de cirujanos distinta.

Por último, el modelo que se ha elaborado implica la sucesión directa de PAE, OR y POST de cada intervención de la secuencia. Por esta razón, la mayoría de las reglas de despacho que se han elaborado sitúan antes a las intervenciones que entran en POST, ya que, de lo contrario, estas podrían quedarse sin tiempo (habría tenido lugar antes el PAE, OR y POST de las intervenciones que entran a PAE, y el OR y POST de las intervenciones que entran a OR). Sería interesante elaborar un modelo que programara la etapa inicial de cada intervención primero, y luego siguiera con las etapas posteriores.

REFERENCIAS

- Araujo, L., & Cervigón, C. (2009). *Algoritmos evolutivos: un enfoque práctico*. Ra-Ma.
- Cañete, C., Fernández-Viagas, V. & Molina, J. M. (2022). *Métodos iterativos voraces para resolver la planificación integrada de consultas y quirófanos*. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla.
https://idus.us.es/bitstream/handle/11441/140234/TFG4001_Ca%C3%B1ete%20Yaque.pdf?sequence=1
- De Antonio Suárez, O. (2013). Una aproximación a la heurística y metaheurísticas. *INGE@UAN*, 1(2).
<http://revistas.uan.edu.co/index.php/ingean/article/view/217>
- Framinan, J. M., Leisten, R., & García, R. R. (2014). *Manufacturing scheduling systems: An Integrated View on Models, Methods and Tools*. Springer Science & Business Media.
- Gorgemans, S., & Urbina, O. (2017). Aproximación al proceso productivo del sistema sanitario español. *ICADE*, 0(99), 69. <https://doi.org/10.14422/icade.i99.y2016.003>
- Herrera, F. (2017). Introducción a los Algoritmos Metaheurísticos. -.
<https://upcommons.upc.edu/handle/2117/191402>
- Melián, B., Moreno, J. A., & Moreno, J. M. (2003). Metaheurísticas: una visión global. *Inteligencia artificial: Revista Iberoamericana de Inteligencia Artificial*, 7(19), 7-28. ISSN: 1137-3601.
<https://www.redalyc.org/pdf/925/92571901.pdf>
- Morera, C. P., Darós, L. C., Herrera, M., & Maheut, J. (2012). Toma de decisiones en la empresa: proceso y clasificación. <http://hdl.handle.net/10251/16502>
- Pérez, P., Framiñán, J. M., & Fernández-Viagas, V. (2018). *Programación de Operaciones*. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla.

Pérez, P., Framiñán, J. M., & Navarro, B. (2022). *Programación de Operaciones*. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla.

Sipper, D., & Bulfin, R. L. (1998). *Planeación y control de la producción*.

<https://s39fbb900fa1cbc52.jimcontent.com/download/version/1519071965/module/14098469730/name/1.-Planeacion-y-control-de-la-produccion--Sipper-Bulfin-1-384.pdf>

Tejeda, A. S., (2011). Mejoras de Lean Manufacturing en los sistemas productivos. *Ciencia y Sociedad*, 36(2), 276-310. ISSN: 0378-7680. <https://www.redalyc.org/articulo.oa?id=87019757005>

Universidad de Granada (2013). *Bioinformática. Algoritmos Genéticos I. Conceptos Básicos*.

<https://sci2s.ugr.es/sites/default/files/files/Teaching/GraduatesCourses/Bioinformatica/Tema%2006%20-%20AGs%20I.pdf>