

**Solución del problema de optimización integrado de programación de salas de cirugía y
turnos de enfermería aplicando técnicas de optimización matemática**

Andrés David Castellanos Ávila y Carlos Eduardo Rojas Villegas

Trabajo de Grado para optar al Título de Ingeniero Industrial

Director:

Javier Eduardo Arias Osorio

Magister en Administración

Universidad Industrial de Santander

Facultad de Ingenierías Físico-mecánicas

Escuela de Estudios Industriales y Empresariales

Bucaramanga

2020

Dedicatoria

A todos aquellos que nos siguieron apoyando

a pesar de las dificultades.

Agradecimientos

A nuestro director, el profesor Javier Arias, por toda su paciencia y apoyo durante el desarrollo de este proyecto, así como por su dedicación y excelencia como docente.

A todos los profesores que durante nuestra vida universitaria han sabido guiarnos y motivarnos a superar las dificultades de la carrera.

A nuestra familia y amigos que han resultado indispensables durante estos años, especialmente desde iniciada esta investigación, por escucharnos y animarnos en los momentos en que más lo necesitábamos

Finalmente, a la Universidad Industrial de Santander y a la Escuela de Estudios Industriales y Empresariales, por sus programas de apoyo estudiantil

Tabla de contenido

Introducción	15
1. Generalidades del Proyecto.....	18
1.1. Objetivos	18
1.1.1. Objetivo General.....	18
1.1.2. Objetivos Específicos.....	18
1.2. Metodología	19
1.2.1. Revisión de literatura	19
1.2.2. Formulación del modelo	19
1.2.3. Elección de la técnica de solución	20
1.2.4. Validación y evaluación del algoritmo	20
1.2.5. Elaboración del artículo	20
1.3. Planteamiento del Problema.	21
2. Revisión de la Literatura	22
2.1. Análisis Bibliométrico	22
2.2. Análisis de Literatura.....	25
2.3. Marco de Antecedentes.....	29
3. Marco Teórico.....	31
3.1. Problema de Optimización.....	31

3.1.1. Programación Salas de Cirugía (ORP).	32
3.1.2. Programación Turnos Enfermería (NSP).....	32
3.1.2.1. Marco legal.	32
3.1.3. Método de Clasificación Única de Procedimientos en Salud – CUPS	33
3.1.3.1. Capítulos.	34
3.1.3.2. Grupos.....	34
3.1.3.3. Subgrupos.	34
3.1.3.4. Categoría.	35
3.1.3.5. Subcategoría.....	35
3.2. Modelos de Optimización	35
3.2.1. Modelos Estáticos y Dinámicos	36
3.2.2. Modelos Enteros y No Enteros	36
3.2.3. Modelos Estocásticos y Determinísticos	37
3.2.4. Modelos Lineales y No Lineales.....	37
3.3. Técnicas de Optimización.....	38
3.3.1. Métodos Exactos	38
3.3.1.1. Programación Lineal.....	38
3.3.1.1.1. Método Simplex.....	38
3.3.1.1.2, Técnica De La Frontera Superior.....	38
3.3.1.2. Programación Lineal Entera.	39

3.3.1.2.1. Branch And Bound.	39
3.3.2. Métodos Heurísticos	39
3.3.2.1. Métodos De Descomposición.	39
3.3.2.2. Métodos Constructivos.	39
3.3.2.3. Método De Reducción.	40
3.3.2.4. Método De Búsqueda Local.....	40
3.3.2.5. Método Combinado.	40
3.3.3. Metaheurísticas	40
3.3.3.1. Búsqueda Local.....	40
3.3.3.2. Algoritmo Colonia de Hormigas (ACO).....	41
3.3.3.3. Búsqueda Tabú.....	41
3.3.3.4. Algoritmos Genéticos	41
4. Diseño del Modelo.....	42
4.1. Descripción del Problema	42
4.2. Descripción de Fases.....	43
4.2.1. Fase 1	43
4.2.2. Fase 2	45
4.3. Modelo Matemático	46
4.3.1. Sets.....	46
4.3.2. Parámetros.....	47

4.3.3. Variables	47
4.3.4. Fase 1	49
4.3.4.1. Función objetivo	49
4.3.4.2. Restricciones.....	50
4.3.5. Fase 2	51
4.3.5.1. Función objetivo.....	51
4.3.5.2. Restricciones.....	52
5. Algoritmo.....	54
5.1. Definición de clases	54
5.1.1. Clase cirugía.....	54
5.1.2. Clase sala.....	55
5.1.2. Clase doctor	56
5.1.3. Clase enfermera	56
5.2. Aleatorización en la generación de clases	57
5.3. Construcción de objetos	58
5.4. Solución del escenario	59
5.5. Algoritmo de variación de hormigas.....	65
5.5.1. Algoritmo de filtrado inicial	66
5.5.2. Algoritmo de eliminación	66
5.5.3. Algoritmo de combinación	67

5.5.4. Algoritmo de búsqueda local	69
6. Evaluación del algoritmo	71
6.1. Descripción de instancias.....	71
6.2. Experimentación	74
7. Análisis de Resultados	79
7.1. Visualización de resultados.....	79
7.2. Análisis de los escenarios iniciales	81
7.3. Análisis de variación de costos	85
7.3.1. Análisis de escenarios finales	89
8. Conclusiones.....	94
9. Recomendaciones	97
Referencias Bibliográficas	99

Lista de Tablas

Tabla 1. Cumplimiento de objetivos	17
Tabla 2. Número de resultados por ecuación de búsqueda.....	23
Tabla 3. Clasificación de resultados según área	24
Tabla 4. Clasificación de resultados según autor.....	25
Tabla 5. Clasificación única de procedimientos en salud - CUPS.....	33
Tabla 6. Código de aleatorización de clases	57
Tabla 7. Estructura para la creación de clases	58
Tabla 8. Estructura de la creación de días.....	60
Tabla 9. Estructura de la separación de cirugías por día.....	60
Tabla 10. Estructura de la separación de salas por categoría.....	61
Tabla 11. Estructura de la programación de cirugías.....	62
Tabla 12. Estructura de la asignación de personal.....	64
Tabla 13. Estructura de la eliminación de costo	67
Tabla 14. Estructura del código de búsqueda local.....	70
Tabla 15. Parámetros de cirugías	72

Lista de figuras

Figura 1. Clasificación de resultados según el país.	24
Figura 2. Visualización del vector solución.....	66
Figura 3. Proceso de combinación de elementos	68
Figura 4. Punto interno de intercambio.....	69
Figura 5. Requerimiento de enfermeras por categoría.....	72
Figura 6. Costos de enfermeras.....	73
Figura 7. Costos de cirujanos.....	74
Figura 8. Pruebas de variación en la población de hormigas.....	75
Figura 9. Pruebas de variación de iteraciones.....	76
Figura 10. Gráfico de comparación de los porcentajes de variación	76
Figura 11. Prueba adicional con variación de iteraciones.....	77
Figura 12. Tabla de visualización de la programación	80
Figura 13. Ejemplo de la programación semanal.....	81
Figura 14. Resultados del primer escenario	82
Figura 15. Uso promedio de salas por día.....	82
Figura 16. Uso promedio de las salas por categoría	83
Figura 17. Cirugías no realizadas según categoría escenario	83
Figura 18. Resultados segundo escenario	84
Figura 19. Cirugías no realizadas por categoría. Escenario 2.....	84
Figura 20. Combinaciones de programación escenario 3	85
Figura 21. Comparativa de optimización de costos	87

Figura 22. Comparativa de optimización de cirugías no realizadas	88
Figura 23. Resultados tercer escenario	89
Figura 24. Resumen cuarto escenario	90
Figura 25. Datos de No realización por categoría de sala.....	90
Figura 26. Uso de tiempo extra, escenario 3.....	91
Figura 27. Comparación uso de tiempo estándar escenarios finales	92
Figura 28. Comparativa del uso de tiempo extra en escenarios finales	93
Figura 29. Cantidad promedio de cirujanos por capítulo de cirugía.....	93

Lista de Apéndices

Ver apéndices adjuntos y pueden ser consultados en la base de datos de la UIS

Apéndice A. Clasificación única de Procedimientos en Salud

Apéndice B. Ejemplo de Programación Semanal – Experimento 1

Apéndice C. Ejemplo de Programación Semanal– Experimento2

Apéndice D. Ejemplo de Programación Semanal – Experimento 3

Apéndice E. Ejemplo de Programación Semanal – Experimento 4

Apéndice F. Ejemplo de Programación Semanal – Experimento 5

Apéndice G. Artículo Publicable

Apéndice H. Artículo Publicable en Inglés

Resumen

Título: Solución del problema de optimización integrado de programación de salas de cirugía y turnos de enfermería aplicando técnicas de optimización matemática *

Autores: Andrés David Castellanos – Carlos Eduardo Rojas **

Palabras clave: Programación multiobjetivo, Programación de salas de cirugía, Programación de enfermeras, Metaheurística de colonia de hormigas, ORP/NSP

Descripción:

Una de las dificultades presentadas a la hora de elaborar la planeación de cirugías en un hospital es la gran variación de éstas, tanto en su duración como con la demanda de cantidad y tipos de recursos. Este proyecto busca dar solución a los dos problemas principales abordados por la literatura en este ámbito, los cuales son: La programación de recursos físicos (salas de cirugía) y programación del personal (doctores y enfermeras). Si bien lo más frecuente es que ambos casos se consideren de manera individual, el objetivo de esta investigación es la de resolver un escenario considerando la integración de ambos temas. Para esto se plantea un modelo de optimización multi-objetivo de dos fases que maximiza el tiempo de ocupación de los quirófanos y minimiza los costos de penalización relacionados con las opciones de programación usuales (aplazamiento, uso de tiempo extra y subutilización de recursos) al tiempo que relaciona los procedimientos programados con un equipo de cirugía que cumpla con unos requisitos mínimos para llevarlos a cabo (cantidad de integrantes y habilidades específicas de cada uno de ellos) . Para la solución del modelo, se opta por aplicar una metaheurística basada en el algoritmo de colonia de hormigas (ACO), para la programación de 120 cirugías con disponibilidad de recursos limitada en una ventana de tiempo de una semana, basándose en múltiples casos de estudio expuestos en la literatura.

* Trabajo de grado

** Facultad de Ingenierías Físico-Mecánicas. Escuela de Estudios Industriales y Empresariales. Director: Magister en Administración Javier Eduardo Arias Osorio

Abstract

Title: solution to the operation room and nurse scheduling integrated programming problem by using mathematic optimization techniques *

Authors: Andrés David Castellanos – Carlos Eduardo Rojas **

Keywords: Multi objective programming. Operation room scheduling, Nurse scheduling, Ant colony optimization, ORP/NSP

Description:

One of the complications when it comes to elaborating surgery planning in a hospital is the great variety amongst them, not only on their durations but also on the demand that varies from procedure to procedure as well as the resources necessary for them to take place. This project's aim is to solve the two main programming problems that have been researched in the past related to this issue: Physical resource programming (operating rooms) and personnel programming (doctors and nurses). Although the more frequent is to find these cases tackled separately, the goal for this study is to solve a scenario that takes into account the integration of both topics. For this means, a two phase, multi-objective optimization model is proposed that maximizes surgery room occupancy while minimizing penalization costs related to the usual programming options (Postponement, overtime and underuse of resources) by programming the procedures in a weekly planning horizon and assigning them the corresponding team to carry them out, following the minimum human resource requirements (skillset and team size). To solve this model, ant colony optimization was the tool that was chosen (ACO) for the programming of 120 procedures with restricted resource availability within the aforementioned time window, based on previous literary study cases.

* Bachelor's degree

** Faculty of Physical Mechanical Engineering. School of Industrial and Business Studies.
Director: Magister of Business Administration Javier Eduardo Arias Osorio

Introducción

La optimización de recursos tanto físicos como humanos, se convierte en una tarea esencial en la búsqueda de la completa productividad y eficiencia de los procesos. La asignación del personal adecuado en las situaciones adecuadas, favorecen la correcta toma de decisiones, desempeño y resultados de la actividad como tal. Sin embargo, es común encontrar, que, en muchos sectores, como, por ejemplo, el sector salud, esta programación se desarrolle empíricamente, basándose en la experiencia y criterio de un Jefe de área, lo que puede llegar a desembocar en costos por sub-utilización o cargos extra por sobre-utilización de los recursos. Este problema ha contribuido al cada vez más grande interés en la optimización de la toma de decisiones administrativas de programación hospitalaria (turnos de enfermería-NSP y salas de cirugía-ORP).

Estos problemas de optimización se suelen considerar como decisiones de nivel táctico (con un horizonte de planificación de 1-2 semanas), de alta importancia en la búsqueda de minimización de costos por uso ineficiente de recursos y maximización en lo relacionado con la satisfacción del cliente (tiempos de espera). Se plantean diferentes restricciones, tales como la clasificación y disponibilidad de sus elementos (cirujanos, enfermeras, salas, etc.), según el enfoque que proponga el autor de la investigación. Sin embargo, es común que se aborden estos problemas, y, por lo tanto, sus objetivos, de manera independiente.

En este proyecto se pretende contribuir a la generación de investigaciones sobre los problemas de administración de recursos de cirugía tratados como un conjunto, que como se puede ver en el análisis bibliográfico, resultan escasos. Para esto, se decide desarrollar un modelo matemático que facilite la integración de los problemas NSP-ORP, y proceder a resolverlo mediante la

implementación de un algoritmo de optimización de colonia de hormigas (ACO) seguido por la evaluación del mismo y el análisis de los resultados obtenidos, tomando como base instancias presentadas en la literatura

Cumplimiento de Objetivos

Tabla 1.

Cumplimiento de objetivos

OBJETIVO	CUMPLIMIENTO
1. Realizar una revisión de literatura sobre los modelos de optimización y técnicas aplicadas en la solución del problema de optimización integrado de programación de salas de cirugía y turnos de enfermería.	Capítulo 2
2. Formulación de un modelo de optimización matemática para el problema integrado de programación de salas de cirugía y turnos de enfermería	Capítulo 4
3. Seleccionar y aplicar dos técnicas de optimización para la solución del modelo definido, a través de software especializado	Capítulo 5
4. Validar y evaluar el desempeño de los algoritmos desarrollados a partir de las técnicas seleccionadas de la revisión de literatura	Capítulo 7
5. Elaborar un artículo de carácter publicable sobre la investigación realizada.	Apéndice G

1. Generalidades del Proyecto

1.1. Objetivos

1.1.1. Objetivo General

Solucionar el problema de optimización integrado de programación de salas de cirugía y turnos de enfermería aplicando técnicas de optimización matemática.

1.1.2. Objetivos Específicos

Realizar una revisión de literatura sobre los modelos de optimización y técnicas aplicadas en la solución del problema de optimización integrado de programación de salas de cirugía y turnos de enfermería.

Formulación de un modelo de optimización matemática para el problema integrado de programación de salas de cirugía y turnos de enfermería.

Seleccionar y aplicar dos técnicas de optimización para la solución del modelo definido, a través de software especializado.

Validar y evaluar el desempeño de los algoritmos desarrollados a partir de las técnicas seleccionadas de la revisión de literatura.

Elaborar un artículo de carácter publicable sobre la investigación realizada.

1.2. Metodología

Para el desarrollo del presente trabajo, se siguió con la siguiente metodología.

1.2.1. Revisión de literatura

Abarca la lectura general de los artículos y trabajos de grado relacionados con el fin de comprender mejor las características del problema y los enfoques abordados por diversos autores, así como para definir las restricciones principales del algoritmo y una posible técnica de solución. Para esto, se construye una ecuación de búsqueda con una ventana de tiempo de 5 años (2014-2019)

1.2.2. Formulación del modelo

Se realiza la construcción de un modelo matemático para la optimización de los problemas de programación de turnos de enfermería (NSP) y salas de cirugía (ORP).

Este modelo busca optimizar la asignación de los recursos, relacionando cada cirugía a programar durante la semana con una sala y un equipo de trabajo adecuado, teniendo en consideración lo analizado en la fase anterior y restricciones adicionales derivadas del marco legal del país. Finalmente se decide dividir el modelo en dos fases con el fin de facilitar la solución del mismo, funcionando de la siguiente manera:

Fase 1: se encarga de realizar una programación preliminar de las cirugías vinculándolas con una sala y en un tiempo estándar diario disponible. Busca generar una configuración que permita la minimización de los costos resultantes de la subutilización de las salas. Para esto, se consideran

como infinitos los recursos de personal (cirujanos y enfermeras), con el fin de maximizar la cantidad total de cirugías programadas en el horizonte de programación.

Fase 2: Se encarga de la asignación del recurso humano adecuado para la programación preliminar. Para esto, evalúa la factibilidad de cumplimiento de la programación resultante de la primera fase tomando en cuenta las restricciones por disponibilidad del personal y requerimientos de cada cirugía según su dificultad. En caso de que esta solución no sea posible, se realiza una reprogramación de las cirugías buscando alejarse lo menos posible de la configuración inicial.

1.2.3. Elección de la técnica de solución

Se realiza la selección de las técnicas de solución más utilizadas por los diferentes autores en la literatura, analizando las características y ventajas de cada una. Posteriormente se escogen las más adecuadas para el proyecto, resultando en el uso una metaheurística basada en el algoritmo de colonia de hormigas para la solución final del modelo. El lenguaje de programación elegido para esta tarea es Python (versión 3.8.3) debido a su facilidad, flexibilidad y eficiencia en el uso de memoria para el procesamiento.

1.2.4. Validación y evaluación del algoritmo

Se construyen los indicadores de desempeño a utilizar en el análisis de resultados, así como las instancias a usar y los diferentes escenarios con los que se busca comprobar la efectividad el algoritmo. Posteriormente se pone a prueba el modelo y se evalúan los resultados obtenidos, verificando de tal forma la utilidad del mismo.

1.2.5. Elaboración del artículo

Se desarrolla un artículo de carácter publicable que sintetiza de manera breve y sencilla, la metodología, resultados y conclusiones encontradas en el desarrollo del proyecto.

1.3. Planteamiento del Problema.

La prestación de servicios a nivel global está siendo definida por el gran crecimiento poblacional que se presenta actualmente, no solo esto, sino que la expectativa de vida es cada vez mayor.

En Colombia se ha venido evidenciando una brecha cada vez más grande entre las necesidades hospitalarias de los ciudadanos y los servicios ofrecidos de manera eficiente por parte de las entidades que los prestan.

Situaciones tales como la deuda actual cercana a los 9.2 billones de pesos de las EPS, el creciente déficit de endeudamiento de las entidades (entre el 4% y 5%), la migración de usuarios entre EPS debido al cierre de las mismas (Saturación del sistema), y que los esfuerzos realizados por brindar un servicio más accesible, han logrado aumentar la cobertura de los sistemas de salud, representan una mayor carga sobre los recursos disponibles, lo que implica que las entidades necesitan encontrar maneras de utilizarlos más eficientemente (Portafolio, 2018).

Uno de los acercamientos a este uso eficiente de recursos es la disminución de los costos hospitalarios con una adecuada programación de turnos de enfermería y salas de cirugía, problemas de optimización que se han abordado principalmente de manera independiente y sobre los cuales se realizaron los trabajos de grado con el grupo de investigación Ópalo titulados: Estudio del problema de programación de turnos de enfermería (Meneses, 2017) y Desarrollo de un estudio para el problema de programación de salas de cirugía (Muñoz, 2018). La integración de los problemas de NSP y ORP, como se evidencia en el artículo “*A branch-and-price approach for integrating nurse and surgery scheduling*” de la *European Journal of operation research*, permitiría la administración de estos recursos de manera sincronizada y congruente lo que

significaría un aumento en la efectividad de la prestación de los servicios hospitalarios a partir de una reducción significativa de costos fuente del desperdicio debido al exceso de mano de obra y la poca flexibilidad de las nóminas planeadas.

2. Revisión de la Literatura.

2.1. Análisis Bibliométrico

Con el fin de conseguir una visión más amplia de la problemática abordada, así como de las vías utilizadas en investigaciones anteriores para la solución del problema, se realiza un análisis bibliométrico en las siguientes bases de datos: *Science Direct*; *Scopus* y *Web of Science*, utilizando una ventana de tiempo de 5 años (2014-2019).

Basado en lecturas preliminares, se construyen las siguientes ecuaciones de búsqueda:

1. TITLE-ABS-KEY ((schedul* OR plan* OR program* OR model) nurse* AND operat* AND room*).
2. TITLE-ABS-KEY ((schedul* OR plan* OR program* OR model) nurse* AND operat* AND room*) AND PUBYEAR > 2014.
3. TITLE-ABS-KEY ("optimiz*" ("schedul*" OR "plan*" OR "program*" OR model) "nurse*" AND "operat* room*").

Inicialmente se selecciona la ecuación (3) obteniendo un total de 66 resultados posterior al proceso de filtración. Debido al gran número de artículos con poca relación con el tema, se decide

refinar la ecuación de búsqueda añadiendo un filtro de exclusión de área después de un proceso de lectura y análisis de los *abstract*. Con la ecuación (4) se obtienen finalmente 29 resultados.

4. TITLE-ABS-KEY ("optimiz*" ("schedul*" OR "plan*" OR "program*") "nurse*" AND "operating room*") AND PUBYEAR > 2014 AND (EXCLUDE (SUBJAREA , "NURS")).

Tabla 2.

Número de resultados por ecuación de búsqueda.

Ecuación	No Resultados
1	1673
2	412
3	66
4	29

El criterio de selección de esta ecuación (4) radica en el hecho de que se obtienen la mayor cantidad de artículos que podrían aportar de alguna manera al desarrollo de la investigación, y el número de artículos irrelevantes o con poca relación al tema se reduce considerablemente. Se procede a realizar un análisis de los datos de los artículos hallados, destacando los siguientes resultados:

- Resultados según el país

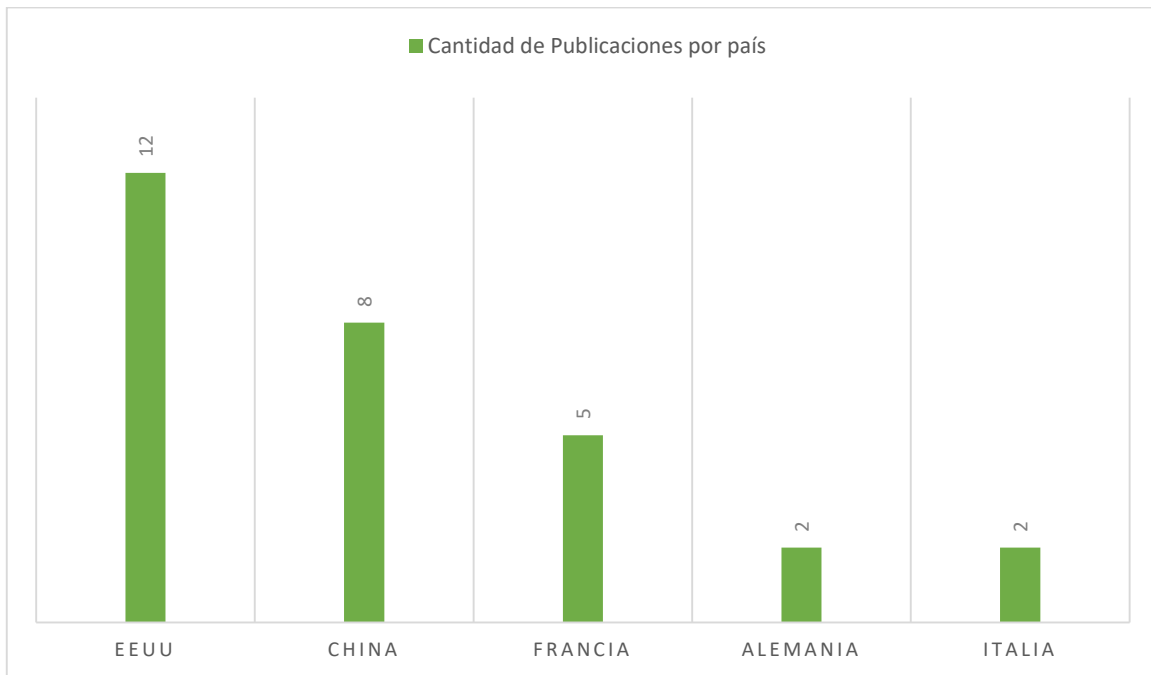


Figura 1. Clasificación de resultados según el país.

- Resultados según el área.

Tabla 3.

Clasificación de resultados según Área

	Número de veces	Área
1	16	Medicina
2	8	Computación
3	8	Ingeniería
4	4	Matemáticas
5	3	Negocios y administración

- Resultados según autor

Tabla 4.

Clasificación de resultados según autor.

	Número de veces	Autores
1	5	Xiang W.
2	2	Demirtas, M.
3	2	Di Martinelly, C.
4	2	Lim, G.
5	1	Abbas, P.I.

2.2. Análisis de Literatura

El problema de optimización de la programación de recursos médicos ha sido ampliamente tratado en la literatura debido a la alta relación con los costos de operación de los hospitales.

Principalmente se han abordado el problema de programación de salas de cirugía (OR) usando un acercamiento multi-objetivo con un foco en la reducción de los costos totales de funcionamiento y la reducción de la subutilización de los OR. Xiang, Yin y Lim (2015) proponen un problema de programación de múltiples salas de cirugía tomando en cuenta restricciones de recursos (enfermeras) de la vida real, desarrollando un modelo de optimización de programación no lineal de enteros mixtos y se implementa un método de optimización de colonia de hormigas (ACO) de 2 capas para resolverlo, la primera capa establece la secuencia de programación de los OR y la segunda la asignación de las enfermeras para cada caso, sin embargo, no se toman en cuenta especialidades entre las salas de cirugía (Se consideran homogéneas).

Guido y Conforti (2017) proponen un modelo multi-objetivo de programación lineal entera que busca determinar de manera integrada los tiempos asignados del OR y la programación semanal de las cirugías con un enfoque de solución de algoritmo genético híbrido (Algoritmo NSGA-II). Una variación de la resolución del problema utilizando un enfoque meta-heurístico de algoritmo genético también se puede observar en Bellkhamza, Jarboui y Masmoudi (2018), los cuales tratan el problema de la programación diaria de cirugías con múltiples fases (pre-operatoria, intra-operatoria y post-operatoria) con múltiples restricciones de recursos (considerando enfermeras homogéneas) basado en un problema “job shop” flexible, proponiendo 2 meta-heurísticas, búsqueda local iterativa y un algoritmo genético híbrido, para realizar la comparación con soluciones anteriormente propuestas con el método ACO, mostrando una mejora en términos de tiempos totales y balance de recursos.

Duenas, Martinelly, Yazgi y Aguado (2017) trabajan el problema de asignación y programación de intervenciones quirúrgicas bajo restricciones de recursos reales (enfermeras heterogéneas) con un modelo de programación de enteros mixtos y resolviéndolo con una técnica de programación de compromiso, el cual se basa en encontrar una solución óptima para cada objetivo sin necesidad de identificar todo el conjunto de soluciones de Pareto.

Más recientemente, Akbarzadeh, Moslehi, Reisi-Nasfchi y Maenhout (2019) abordan el problema de la reprogramación de las intervenciones quirúrgicas debido a la diferencia entre la demanda real con la planificada, con el fin de equilibrar la capacidad del nivel operativo y reducir los costos por sub utilización de los recursos, esto lo logran proponiendo un modelo de programación de enteros mixtos y una metodología de solución heurística de 3 fases: primero, generación de columnas para resolver el problema relajado; segundo, técnicas heurísticas para

transformar la solución fraccionaria en una solución entera factible; finalmente, búsqueda local para mejorar la solución.

La programación de turnos de enfermería y equipos de trabajo enfocados en el desarrollo de intervenciones quirúrgicas ha tratado principalmente los objetivos de minimización de tiempos de inactividad y afinidad entre el personal asignado buscando aumentar la eficiencia del procedimiento. Burke, Li y Qu (2010) presentan un modelo híbrido multi-objetivo que combina programación entera (IP) y búsqueda de vecindad variable (VNS) para resolver un problema altamente restringido de programación de turnos de enfermería. La IP se utiliza para resolver el sub-problema que incluye el conjunto completo de restricciones suaves y duras relacionadas a la carga laboral y restricciones legales, luego el VNS sigue como un procedimiento de post-procesamiento para mejorar aún más las soluciones resultantes de IP.

Lim, Mobasher, Bard y Najjarbashi (2016) presentan una solución al problema de asignación de turnos de enfermería teniendo en cuenta los tiempos de descanso, abordando un modelo de optimización multi-objetivo y desarrollando un algoritmo de generación de columnas y una heurística de intercambio de 2 fases para encontrar asignaciones factibles rápidamente.

Martinelly y Meskens (2017) proponen modelo de programación entera bi-objetivo para la construcción de equipos quirúrgicos teniendo en cuenta la programación semanal de los OR, maximizando la afinidad cirujano-enfermera, mientras minimiza el tiempo de inactividad de las enfermeras. La solución de este problema se lleva a cabo utilizando el método de restricción ϵ , el cual busca la optimización de cada función objetivo para hallar los puntos ideales y límites del modelo, sin generar bucles en las iteraciones de la solución y resolviendo la secuencia de sub-problemas donde cada objetivo se convierte posteriormente en una restricción.

La planificación de las salas de cirugía y los turnos de enfermería se pueden considerar como 2 problemas principales en la programación de recursos hospitalarios, y aunque se han tratado ampliamente de manera independiente, los artículos que trabajan el problema integrado son escasos. Belien y Demeulemeester (2008) abordan el problema de programación semanal de OR bajo restricciones de asignación de enfermeras homogéneas, la planificación de la cirugía utiliza la programación de enteros para desarrollar un programa maestro de cirugía con el objetivo de cumplir las preferencias de las enfermeras al tiempo que garantiza el cumplimiento de los requisitos de demanda. El problema es resuelto utilizando la técnica de generación de columnas, el cual involucra la solución de 2 tipos de problemas de precios: el primero se resuelve con un enfoque de programación dinámica estándar; el segundo por medio de un optimizador de programación de enteros mixtos.

Xiang, Ma y Wang (2015) proponen un problema de optimización multi-objetivo para la programación conjunta de salas de cirugía y equipos de trabajo bajo un enfoque ACO de 2 capas similar al trabajado por Xiang et al. (2015). Este acercamiento toma en cuenta la programación semanal de los OR y los requerimientos de especialización de las enfermeras (heterogéneas) y tiene como objetivo, la minimización de la variación de las cirugías programadas y carga de trabajo asignado, así como la maximización de la satisfacción de preferencias del equipo.

Después de una revisión completa de los artículos, se concluye que el problema integrado de programación de OR y NS se ha trabajado en escasas ocasiones, evidenciando principalmente revisiones de los temas por separado y trabajando especialmente con modelos de programación lineal entera mixta; en cuanto a las técnicas de solución se evidencia una preferencia por las técnicas de optimización de colonia de hormigas, de algoritmo genético y una combinación de heurísticas apropiadas para el modelo en específico.

2.3. Marco de Antecedentes

En cuanto a lo que la logística hospitalaria se refiere, si se analiza el hospital como una organización, ha de tenerse en cuenta el sinnúmero de procesos que se realizan y un conjunto de recursos necesarios para su ejecución apropiada. Dada la naturaleza de estos recursos y la importancia que estos procesos representan para la sociedad, es necesario que sean optimizados mediante el uso de diferentes técnicas que permitan cuantificar y finalmente mejorar la calidad y efectividad de los servicios resultado de las actividades hospitalarias.

Bautista (2015) desarrolla una meticulosa revisión de algunos de los artículos relacionados a los procesos hospitalarios que han sido analizados y formulados como problemas de optimización en su “Estado del arte de los modelos de optimización en la logística hospitalaria”, clasificándolos según las actividades logísticas en las que se desarrollan dichos procesos, y ahondando específicamente en la revisión de los siguientes ámbitos: El abastecimiento de la cadena de suministro, incluyendo la gestión de inventarios farmacéuticos y el ruteo interno y externo de inventarios; los servicios del departamento de emergencia en cuanto a la localización de los centros de atención y puntos de distribución, los problemas relacionados al sistema de vehículos de emergencia (localización y ruteo de ambulancias, y la programación del personal en urgencias y unidades médicas); la localización de centros de atención hospitalarios, asignación de recursos médicos, programación de admisiones y atención de salud domiciliaria; y en última instancia la programación de salas de cirugía (ORP) y la programación de turnos de enfermería (NSP).

Tras la exhaustiva revisión, se hace notar el número de artículos relacionados con la programación de turnos de enfermería y se enfatiza en la importancia de la existencia de herramientas y métodos que nos permitan atender las necesidades cubiertas por la atención en salud de manera tan ágil como el sistema lo permita. En este orden de ideas se decide por su

relevancia, ahondar en los problemas de programación de salas de cirugía y turnos de enfermería mediante la realización de dos estudios individuales de cada problema.

El primer problema es revisado por Muñoz (2018) en su trabajo “Desarrollo de un Estudio para el Problema de Programación de Salas de Cirugía” en el que se plantea un modelo de programación de bloques de cirugía, dividido en dos fases, una fase preliminar en la que se establecen los pacientes a atender por día en el horizonte de planeación (1-2 semanas) teniendo en cuenta factores como la urgencia del procedimiento y una segunda fase en la que se secuencian las intervenciones quirúrgicas asignadas a cada día orientando la programación en la minimización de los costos operativos de las operaciones. Tras modelar el problema se utiliza el solver de Excel y GAMS para comprobar su correcto funcionamiento y posteriormente se utilizan los datos de una entidad prestadora de servicios de salud (IPS Serviclínicos Dromédica S.A.) para someter el modelo a una situación con datos reales, situación en la que se evidencia una mayor eficiencia en la utilización de los recursos hospitalarios tras aplicar el modelo de optimización planteado por el autor a comparación del método usado por Serviclínicos.

Para el segundo problema, Meneses (2017) elabora un “Estudio del problema de programación de turnos de enfermería”, en él, se modela el problema mediante el uso de una matriz de turnos y funciones de ajuste, teniendo como objetivo la minimización de la sub-programación y sobre-programación de las enfermeras, sin dejar de lado la sinergia del equipo de trabajo en veras de maximizar la calidad del servicio ofrecido. Después de programar y probar el modelo en MATLAB, se utilizan métodos Metaheurísticos de solución como búsqueda adaptiva y un algoritmo híbrido de colonia de abejas y ascensión en colina. El trabajo logra demostrar tras la solución del modelo, que mediante el uso de técnicas de programación y optimización es posible

cubrir satisfactoria y efectivamente la demanda del recurso humano sin sacrificar la sinergia de los equipos de trabajo.

Como se mencionó anteriormente, estos dos problemas han sido estudiados por separado en diversas ocasiones, y como se pudo evidenciar en los trabajos mencionados, modelarlos con herramientas de programación ha generado resultados satisfactorios, tanto desde el punto de vista del mejoramiento de la calidad del servicio prestado como de la optimización del uso de los recursos disponibles, sin embargo, dada la estrecha relación que guarda la programación de los turnos de enfermería con la programación de salas de cirugía, integrar la optimización de ambos problemas en uno solo permitiría la distribución de dichos recursos de una manera más sinérgica lo que potencialmente podría mejorar los procedimientos actuales para la programación de ambos recursos.

3. Marco Teórico

3.1. Problema de Optimización

Se puede entender como una representación escrita de un caso o problema de la vida real y que tiene como fin, buscar la mejora (ya sea maximización o minimización) de una o varias variables, las cuales son representadas por medio de una ecuación matemática (Díaz, Huecas, Allende, & Manso, 2007). Para abordar los problemas se requiere generar un modelo, el cual se encargará de describir matemáticamente los objetivos y las restricciones que regirán el problema y elegir la/as técnicas que permitan hallar una solución óptima o factible. Para esta investigación en particular, se trabajarán los siguientes problemas de optimización:

3.1.1. Programación Salas de Cirugía (ORP).

Aborda la programación de las salas de cirugía de un hospital, las cuales se pueden considerar como homogéneas (sin especialización por tipo de cirugía) o por bloques (las salas solo pueden atender cierto tipo de cirugías por restricciones de especialización). La programación de las cirugías generalmente se trata como una decisión táctica (1-2 semanas) y tomando en cuenta restricciones de secuenciación, uso de recursos, priorización de cirugías, entre otras.

Este problema ha sido ampliamente tratado en la literatura y tanto su función objetivo como los parámetros y restricciones, varían según el autor, siendo los más comunes, la búsqueda de la minimización tanto de los costos totales, como de la sub-utilización de las salas (OR).

3.1.2. Programación Turnos Enfermería (NSP)

Aborda la programación de los turnos de enfermería siguiendo un conjunto de restricciones blandas y duras, las cuales generalmente están dadas por el contrato de trabajo y el marco legal del país en cuestión. En lo relacionado a las prácticas quirúrgicas, Es común que los turnos de enfermería se programen de manera empírica por el/la Jefe de enfermería, basando la decisión en variables como la demanda actual, la habilidad o experiencia, la especialización requerida (según el tipo de operación), el rango (scrub nurse – circulating nurse) y disponibilidad del personal.

Al igual que el problema ORP, el NSP también ha sido extensamente trabajado en la literatura, siendo su foco central la minimización de costos producidos por el trabajo de horas extra y la maximización de la afinidad entre el grupo de trabajo para la realización de las cirugías.

3.1.2.1. Marco legal. La normatividad que rige la profesión de enfermería cobra importancia en este contexto, al exponer condiciones y limitaciones inviolables a las cuales el personal de enfermería y por lo tanto el programador de turnos está sujeto. estas normas se representan en

restricciones duras (debido a su importancia e inflexibilidad) y se corresponden a los temas relacionados con la secuenciación de turnos, máximo de horas extra y tiempos de descanso obligatorios (días libres y descansos post cirugía) (Xiang, Ma, & Wang, 2015).

3.1.3. Método de Clasificación Única de Procedimientos en Salud – CUPS

Corresponde a un ordenamiento lógico y detallado de los procedimientos y servicios en salud que se realizan en el país, identificados por un código y una descripción validada por los expertos” (Resolución No 0005851 de 2018). Esta herramienta se divide en 5 secciones y 28 capítulos como se muestra en la Tabla 5.

Tabla 5.

Clasificación Única de Procedimientos en Salud - CUPS.

CUPS	
	Procedimientos Quirúrgicos E
Sección 00	Intervencionistas Capítulos 01 – 14
	Procedimientos No Quirúrgicos
Sección 01	Capítulos 15 – 24
	Salud Pública
Sección 02	Capítulo 25
	Riesgos Laborales
Sección 03	Capítulos 26 Y 27
	Servicios Salud A La Población Indígena
Sección 04	Capítulo 28

3.1.3.1. Capítulos. Organizan y facilitan la ubicación de un procedimiento, pero no hacen parte de la estructura del código del procedimiento seleccionado.

3.1.3.2. Grupos. Señalado por los 2 primeros caracteres del código, según el capítulo en el cual se encuentra ubicado.

- Zona anatómica específica para los capítulos del 01 al 14
- La unidad de producción específica para los capítulos 15 al 24
- El proceso en la colectividad para el capítulo 25
- Las condiciones, factores y tipos de proceso en el trabajo para los capítulos 26 y 27
- Los servicios de salud para el capítulo 28

3.1.3.3. Subgrupos. Definido por el tercer carácter, según el grupo en el que se encuentra ubicado. Indica:

- Tipo de procedimiento para los grupos 01 al 86
- Tipo de imagen para los grupos 87 y 88
- Tipo de área técnica para los grupos 90 y 91
- Tipo de acción para los grupos 89, 90 y 91
- Tipo de estrategia para los grupos A1 al A5
- Tipo de fase en la atención para los grupos T1 y T2
- Tipo de nivel institucional o territorial para el grupo T9
- Tipo de servicios para los grupos S0, S1, S2, S3, S4 y S5

3.1.3.4. Categoría. Identificado por el cuarto carácter indica en forma genérica la nomenclatura del procedimiento.

3.1.3.5. Subcategoría. Identificado por los dos últimos caracteres, define con mayor precisión el procedimiento de acuerdo a variables como técnica, tecnología, método, tipo de muestra, entre otras.

3.2. Modelos de Optimización

En cuanto a lo que la investigación de operaciones se refiere, existe un sinnúmero de modelos matemáticos que permiten, de manera cuantitativa y a través de herramientas matemáticas, formular problemas cotidianos como la planificación de inventarios para un almacén, el ruteo de vehículos por la ruta más corta y la planificación del número de servidores en un sistema de colas, estos son conocidos como modelos de optimización. (Hillier & Lieberman, 2015).

Para formular un modelo de optimización es necesario conocer el problema a fondo, con el fin de establecer objetivos claros para el modelo y conocer qué elementos han de tenerse en cuenta que puedan restringir el modelo, e incluso analizar las relaciones o similitudes que pudieran llegar a haber con otras áreas, como herramienta para tener alternativas de acción.

Después de definir el problema, se debe formular de manera que sea conveniente para analizar. Aquí es donde entra en juego la investigación de operaciones, cuyo acercamiento se basa, como se mencionó anteriormente, en la construcción de un modelo matemático que representa la esencia del problema.

Esta representación, llamada función objetivo, relaciona de manera apropiada a la situación, las decisiones cuantificables que nos permitirá tomar el modelo, y las representa como variables de decisión; las constantes relacionadas al problema se conocen como parámetros; y cualquier otra

consideración que pueda afectar la función objetivo se conoce como restricción, estas generalmente se representan mediante sumatorias y desigualdades.

3.2.1. Modelos Estáticos y Dinámicos

Basándose en Hillier y Lieberman (2015). Los modelos estáticos son técnicas matemáticas que buscan determinar una solución para una serie de condiciones fijas que no presentarán cambios significativos a corto plazo (no se toma en cuenta el factor del tiempo). En esencia, este modelo resuelve problemas luego de un solo intento cuyas soluciones dictan valores óptimos.

Por otra parte, los modelos dinámicos, son usados para la toma de una secuencia de decisiones interrelacionadas, las cuales condicionan la evaluación futura del sistema (toma en consideración el factor tiempo). Estos modelos no cuentan con una formulación matemática estándar, por lo cual las ecuaciones utilizadas deben desarrollarse para cada situación específica. Se caracteriza por:

- El problema puede dividirse en fases, donde cada una posee una política de decisión.
- Cada fase tiene un número de estados asociados
- El proceso de solución implica el análisis de cada fase en orden inverso (de la última a la primera), finalizando con la obtención del óptimo del problema posterior al análisis de la primera fase.
- Las políticas de decisión se consideran independientes con respecto a las de las etapas anteriores (principio de optimalidad)

3.2.2. Modelos Enteros y No Enteros

Cuando nos referimos al modelo matemático de programación entera hacemos referencia al modelo de programación lineal con una restricción adicional que indica que las variables tienen que tener un valor entero, adicionalmente, cuando solo se requiere que algunas de las variables

tengan valores enteros en el modelo, se hace referencia a programación entera mixta (MIP) (Hillier & Lieberman, 2015).

3.2.3. Modelos Estocásticos y Determinísticos

Uno de los factores a considerar a la hora de formular un modelo es la variabilidad que presentan los parámetros del modelo, y dado que muchos de estos se utilizan para decidir un curso de acción futuro, los valores de los parámetros deben estar basados en una predicción de las condiciones futuras. Esto en la mayoría de los casos resulta en una cantidad considerable de incertidumbre respecto a los valores reales que el parámetro puede tomar cuando la solución óptima del modelo se implemente.

En el caso de aquellos modelos en los que todos los datos se suponen como conocidos y constantes se estaría tratando de un modelo determinístico, sin embargo, cuando alguno de los datos se considera incierto y se ve representado por distribuciones de probabilidad el modelo se trata de un modelo estocástico. (Taha, 2012)

3.2.4. Modelos Lineales y No Lineales

Como se mencionó anteriormente, la función objetivo y restricciones son representaciones matemáticas del problema y las circunstancias que lo afectan, pues bien, se considera un modelo lineal a aquel en el que tanto sus restricciones, como la propia función objetivo están modelados como una función lineal.

Los modelos no lineales son aquellos donde al menos una de las variables de decisión y/o las restricciones del modelo se representan mediante una función no lineal, esto nos permite trabajar problemas que normalmente estarían restringidos por el factor de linealidad, e incluso trabajar

problemas que normalmente se modelan linealmente bajo suposiciones, de manera más aproximada al comportamiento real. (Hillier & Lieberman, 2015).

3.3. Técnicas de Optimización

Después de la formulación del modelo matemático del problema, el siguiente paso a seguir es el de obtener soluciones para este. Para ello existen métodos que arrojan soluciones factibles u óptimas, conocidos como métodos de optimización, los cuales generalmente se programan a manera de algoritmo con el fin de utilizar la capacidad de procesamiento de un computador para generar soluciones suficientes para obtener la información necesaria para la toma de decisiones.

3.3.1. Métodos Exactos

3.3.1.1. Programación Lineal.

3.3.1.1.1. Método Simplex. El método simplex es un procedimiento algebraico cuyos conceptos se basan en la geometría, y cuya utilidad es la de resolver de manera eficiente y rutinaria problemas de programación lineal de gran volumen, esto se debe en gran manera a la programación del método en los computadores hoy en día. (Hillier & Lieberman, 2015).

3.3.1.1.2, Técnica De La Frontera Superior. Dado que es usual que muchos de los problemas de programación lineal tengan alguna o todas sus variables relacionadas a restricciones de frontera superior, y que en cuanto a tiempo computacional las restricciones funcionales son las que principalmente aumentan este tiempo, se desarrolló el método de la frontera superior, el cual elimina las restricciones de frontera superior de las restricciones funcionales y las trata por separado, aplicando el método simplex al nuevo problema simplificado sin dejar de respetar las restricciones de frontera superior. (Hillier & Lieberman, 2015).

3.3.1.2. Programación Lineal Entera.

3.3.1.2.1. Branch And Bound. A pesar de que para muchos de los problemas de programación lineal entera existan métodos que permitan encontrar una solución óptima dentro del conjunto finito de soluciones para el problema, sin embargo, muchos de estos métodos logran explorar este conjunto de manera exitosa a cambio de tomar tiempo considerable en conseguirlo. El método de Branch and Bound y sus variaciones, analiza porciones del conjunto de solución acotando y limitando la función objetivo progresivamente convirtiéndola en funciones más pequeñas que sean más fáciles de resolver, es por esto que también se le conoce como divide y vencerás. (Hillier & Lieberman, 2015).

3.3.2. Métodos Heurísticos

Algunos problemas de optimización se consideran demasiado complejos para encontrar la solución óptima del modelo. Para estos casos, comúnmente se utilizan soluciones heurísticas, las cuales son procedimientos con los que se busca encontrar soluciones factibles a un problema específico que se acerquen lo más posible al óptimo. Este procedimiento generalmente se lleva a cabo mediante un algoritmo iterativo, donde cada iteración es una búsqueda de una mejor solución. Los métodos heurísticos son usualmente diseñados para ajustarse a problemas específicos sin posibilidad de generalización o aplicación a problemas similares (Hillier & Lieberman, 2015). Entre las técnicas más conocidas se encuentran:

3.3.2.1. Métodos De Descomposición. En los cuales se descompone el problema original en sub-problemas de menor complejidad que son tratados de manera independiente (Suarez, 2011).

3.3.2.2. Métodos Constructivos. Son métodos deterministas que desarrollan paso a paso la solución del problema y se basan en la mejor elección de cada iteración (Cunquero, s.f)

3.3.2.3. Método De Reducción. Identifica propiedades que son cumplidas por las buenas soluciones y se introducen como nuevas restricciones con el fin de simplificar el problema (Suarez, 2011).

3.3.2.4. Método De Búsqueda Local.

Consiste en la mejora progresiva de una solución inicial, deteniéndose cuando la solución no puede ser mejorada (se encuentra un óptimo local). (Cunquero, s.f)

3.3.2.5. Método Combinado. Es posible realizar la combinación de un método constructivo y uno de búsqueda local, considerando el primero como la solución inicial del segundo. (Cunquero, s.f)

3.3.3. Metaheurísticas

Según Hillier y Lieberman (2015), se puede entender la naturaleza de las metaheurísticas como “un tipo general de método de solución que proporciona tanto una estructura general como las pautas estratégicas para desarrollar un método heurístico específico que se adapte a un tipo particular de problema”. El rol principal de las metaheurísticas es lidiar con problemas que son demasiado extensos y complejos para ser resueltos por algoritmos exactos y se han convertido en una herramienta indispensable para la resolución de problemas como los de ORP y NSP.

3.3.3.1. Búsqueda Local. Es una metaheurística basada en la trayectoria, la cual, partiendo de un punto inicial, va mejorando la solución a partir de la exploración del entorno cercano dentro del espacio de búsqueda con el fin de encontrar un óptimo local. Tras cada iteración, la nueva solución se evalúa según condiciones establecidas inicialmente y si es aceptada, se convertirá en un el nuevo óptimo local. El proceso se repetirá hasta que no se encuentre una mejor solución dentro el espacio de búsqueda (vecindario) (Rodríguez, Peña, & Rincón, 2014).

3.3.3.2. Algoritmo Colonia de Hormigas (ACO). Es una metaheurística basada en población que utiliza una estrategia de búsqueda en la que un conjunto de individuos (población) transmite e intercambia información con el resto, abarcando una gran cantidad de soluciones y permitiendo la elección de las mejores siguiendo los parámetros de decisión designados inicialmente. Esta técnica se inspira en la forma de búsqueda de comida de las hormigas, en la cual, los individuos (hormigas) eligen arbitrariamente zonas de búsqueda cercanas al hormiguero, en caso de encontrar comida, la transportan al hormiguero, dejando un rastro (feromona) que sirve como mecanismo de comunicación para los demás individuos de la colonia. Las hormigas de generaciones siguientes son atraídas a realizar la búsqueda en el espacio de solución cercano que ha conseguido mejores resultados previamente, evitando así los caminos (soluciones) menos óptimos (Rodríguez et al., 2014).

3.3.3.3. Búsqueda Tabú. Es una metaheurística basada en la trayectoria, su estrategia se basa en la búsqueda de soluciones en diferentes vecindarios permitiendo el movimiento por soluciones menos óptimas y guardando las mejores soluciones en una lista. La técnica prohíbe temporalmente movimientos que puedan volver a soluciones recientes para evitar bucles, estas prohibiciones se realizan mediante el uso de una memoria de corto plazo que guarda los movimientos prohibidos la cual se le denomina “lista tabú”. La técnica también incorpora conceptos como la intensificación, el cual permite la búsqueda exhaustiva en zonas de la vecindad que se identifiquen como posibles contenedores de buenas soluciones, y la Diversificación, el cual fuerza la búsqueda en nuevas zonas modificando las reglas de elección (Hillier & Lieberman, 2015).

3.3.3.4. Algoritmos Genéticos. Es un tipo de algoritmo evolutivo, el cual a la vez pertenece a las metaheurísticas basadas en la población. Este método se basa en la teoría evolutiva de la solución natural formulada por Charles Darwin donde los individuos heredan rasgos de los padres

y aquellos con las mejores características tienen más posibilidades de sobrevivir, ser padres y por lo tanto transmitir sus genes a nuevas generaciones, mejorando así, poco a poco la población. Esta técnica no trabaja con una sola solución posterior a cada iteración, sino con una población de soluciones (entre ellas las poseedoras de las mejores características, las cuales se miden con el valor de la función objetivo).

Ocasionalmente se presentan mutaciones en los individuos de nuevas generaciones que les permiten poseer características que no se encontraban en los padres (ya sean beneficiosas o no) y que permiten la búsqueda en nuevas y quizá mejores zonas de la región factible. Eventualmente, la supervivencia de los mejores individuos conducirá al algoritmo a encontrar una solución cercana a la óptima. (Hillier & Lieberman, 2015)

4. Diseño del Modelo

4.1. Descripción del Problema

El problema de programación de los recursos médicos orientados a la realización de cirugías comprende un gran número de variables y parámetros íntimamente ligados entre ellos que pueden dificultar la toma de decisiones al momento de realizar su organización sin que se vea afectado negativamente la efectividad de la planificación, tanto a nivel financiero como en lo relacionado satisfacción del cliente.

Por un lado, se debe realizar la correcta asignación de procedimientos/cirugías en una sala de operaciones que cumpla con los requisitos mínimos para que ésta se lleve a cabo, esto es, considerar su disponibilidad física en una fecha prevista establecida por el cirujano, una hora

específica a lo largo del día y la duración estimada de la cirugía, de tal forma que no genere conflictos con otras operaciones programados (ya sea porque las ventanas de tiempo no son suficiente para su realización o porque se presentan cirugías de mayor prioridad). Adicional a esto se debe verificar su clasificación, la cual determina la competencia de la sala en cuanto a la realización de ciertos procedimientos.

Por otro lado, se deben tomar en cuenta las restricciones derivadas del recurso humano (cirujanos y enfermeras) entre las que destacan: la cantidad mínima de personal necesario para llevar a cabo la cirugía (número que depende de su complejidad), así como las habilidades de los integrantes del equipo, su experiencia en el área y la duración máxima de los turnos laborales (incluyendo horas extra).

A partir de lo anteriormente mencionado, una programación óptima será aquella que logre minimizar los costos relacionados con el uso ineficiente de recursos, las cuales son: subutilización de salas, uso de horas extras, aplazamiento y no realización de las cirugías. Para esto, se decide dividir la programación en dos fases con el fin de facilitar el análisis y hallazgo de soluciones factibles.

4.2. Descripción de Fases

4.2.1. Fase 1

Esta fase se encarga de realizar la programación de las cirugías en sus respectivas salas y la minimización de la subutilización de las mismas. Inicialmente se supone que se cuenta con disposición de personal suficiente para realizar cualquier procedimiento, es decir, se considera como recurso infinito la disponibilidad de enfermeras y cirujanos con el fin de maximizar en la fase inicial el número de procedimientos realizados durante el periodo de programación (horizonte de programación de una semana). Para ello también se considera que tanto enfermeras como

cirujanos trabajan turnos sincronizados de 12 horas diarias, los cuales estarán divididos en 48 ventanas de tiempo uniformes con una duración estándar de 15 minutos. A su vez, se considera la posibilidad de la utilización de las salas por un tiempo extra máximo de 3 horas posteriores a las 12 horas de turno estándar, estas horas extra estarán igualmente divididas en un total de 12 ventanas de tiempo de 15 minutos programables que se usarán según como la demanda semanal lo requiera.

La programación de las cirugías se lleva a cabo al inicio de la semana (por lo que no se contempla la inclusión de procedimientos de emergencia) y se ordena de acuerdo con una prioridad asignada por el cirujano (dada en una escala ascendente de 1 a 5) que está relacionada con la urgencia, la complejidad y riesgo que conlleve la misma, y que primará por sobre los días de retraso acumulados de la operación. La duración de los procedimientos se toma como fija e incluirá: tiempo de realización de la cirugía, preparación de la sala e imprevistos.

Las salas de cirugía están catalogadas de acuerdo a los insumos con los que cuentan para efectuar una determinada intervención quirúrgica, y el conjunto de los procedimientos que pueden realizar se definen basándose en los 14 capítulos expuestos por la Clasificación Única de Procedimientos en Salud (CUPS) (Ver apéndice A), los cuales a su vez se pueden desglosar en 86 grupos que especifican la parte o zona del cuerpo en la que se llevará a cabo la cirugía. Tomando esto en consideración, se contempla como posible adecuar una sala para atender un procedimiento para el que normalmente no estaría preparada, a esta adecuación se le denominará como subutilización y se le asignará un costo de penalización.

4.2.2. Fase 2

En esta fase, el modelo busca la mejor asignación del recurso humano de acuerdo a los requerimientos de la programación de la fase 1, minimizando los costos relacionados al uso ineficiente de los integrantes del equipo de trabajo

Para la correcta asignación de recursos, se realiza una clasificación del personal, según las necesidades específicas que puedan satisfacer para la adecuada realización de las cirugías. De manera que los cirujanos se dividirán en 3 grupos según su tiempo de experiencia (Grupo C: menor a 3 años de experiencia; Grupo B: de 3 a 10 años de experiencia; Grupo A: mayor a 10 años de experiencia) y las enfermeras en 3 grupos según su especialización (Grupo A: Primera ayudante; Grupo B: enfermera instrumentadora; Grupo C: enfermera general/circulante). Y se considerará subutilización, cuando un cirujano de mayor experiencia realice cirugías destinadas inicialmente a cirujanos de menor experiencia.

Durante la asignación de recursos, se tiene en cuenta, que tanto cirujanos como enfermeras comparten un turno sincronizado de 12 horas (de 6 am a 6 pm) dividido en 48 ventanas de tiempo como se especifica en la fase 1. Así mismo, se busca programar los tiempos de descanso opcionales para el personal, los cuales contarán con una duración estándar igual tanto para enfermeras como para cirujanos, que iniciarán una vez se haya alcanzado un número específico de procedimientos y que variarán según la prioridad de la cirugía realizada (2 horas para una cirugía de prioridad 5, 1 hora para una cirugía de prioridad 3-4 y 0,5 hora para una cirugía de prioridad 1-2).

Posterior a la asignación del equipo adecuado, el modelo evalúa la factibilidad de cumplir con la programación ideal, establecida en la fase 1 y la reprograma en caso de ser necesario, comparando configuraciones con las que se cumplan las restricciones de la programación de

personal, incluyendo los tiempos de descanso y contemplando la posibilidad de que una cirugía pueda ser realizada por un cirujano diferente con disponibilidad de tiempo, siempre y cuando este tenga un nivel de experiencia igual o mayor al del doctor que agendó la cirugía durante la fase 1.

4.3. Modelo Matemático

Dado que el problema se divide en dos fases, primero se enumerarán los conjuntos/sets, parámetros y variables necesarios para la formulación y solución del modelo, y posteriormente se procederá a la descripción de las funciones objetivo y restricciones de cada fase por separado.

4.3.1. Sets

- **D** = conjunto de días laborales disponibles; (d =lunes (1), martes (2), miércoles (3), jueves (4), viernes (5), sábado (6))
- **L**= Categoría de las salas de cirugía; ($l=1,2,3,4$)
- **P**= conjunto de posibles prioridades de la cirugía i dada por el cirujano c ; ($p=1,2,3,4,5$)
- **H**= tipo/clasificación de las cirugías a realizar; ($h= 1,2,3... 86$)
- **V** = ventanas de tiempo estándar para programar en el día ($v= 1, 2, \dots 48$)
- **VE**= ventanas de tiempo extra disponibles para programar en el día ($ve= 49,50...60$)
- **B** = conjunto de ventanas de tiempo totales disponibles ($b =1,2,3... 60$)
- **S** = conjunto de salas de cirugía; ($s =1,2,3... S$)
- **C** =conjunto de cirujanos; ($c= 1,2,3... C$)
- **N** = conjunto de enfermeras; ($n= 1,2,3...N$)
- **K** = tipo/clasificación de las enfermeras; ($k = A, B, C$)
- **G**= tipo/clasificación de los cirujanos; ($g= A, B, C$)
- **I**= conjunto de cirugías a realizar; ($i= 1,2,3...P$)

4.3.2. *Parámetros*

- **Trd** = número de ventanas de tiempo estándar diarias disponibles para la realización de cirugías en la sala s en el día d .
- **Q**= ventanas de tiempo extra diarias disponibles para la realización de cirugías
- **CEC(g)** = Costo por cada hora extra trabajada por un cirujano de categoría g .
- **CEN(k)** = Costo por cada hora extra trabajada por una enfermera de categoría k .
- **CSC(g)** = Costo por subutilización de un cirujano de categoría g .
- **U(i)** = Duración estándar de la cirugía i .
- **CES(s)** = Costo por el uso de una ventana de tiempo extra de una sala durante la realización de la cirugía.
- **CSS(s)** = Costo por el uso de una ventana de tiempo en subutilización de una sala de cirugía.
- **R(p)**= Costo por cada día retrasado para iniciar una cirugía de prioridad p .
- **CA(p)** = Costo por la no realización de una cirugía de prioridad p durante la semana
- **E(k,h)**= Número de enfermeras de tipo k requeridas para realizar una cirugía de categoría h .
- **A(h)**= Número total de enfermeras requeridas para una cirugía de categoría h .
- **RE(i,h)** = Número de ventanas de tiempo de descanso establecidas al finalizar una cirugía i de categoría h .

4.3.3. *Variables*

- **TO(i,s,b,d)**= ventana de tiempo b de inicio de la cirugía i , en la sala s del día d .
- **TF(i,s,b,d)** = ventana de tiempo b de finalización de la cirugía i , en la sala s del día d .

- $ES(s,b,d) = 1$ si la sala de cirugía s , se encuentra en uso durante una ventana de tiempo $b > 48$ del día d (tiempo extra).
- $BS(s,b,d) = 1$ si la sala de cirugía s se subutiliza para un procedimiento quirúrgico durante la ventana de tiempo b del día d .
- $J(i,h) =$ Día programado inicial para la realización de la cirugía i de la categoría h ; ($J \in d$).
- $O(i,p,d) = 1$ si la cirugía i de prioridad p no fue realizada el día d en que fue programada.
- $ZW(i,s,b,d) = 1$ si la cirugía i fue asignada a la sala de cirugía s durante la ventana de tiempo $b < 49$ del día d .
- $ZWE(i,s,b,d) = 1$ si la cirugía i fue asignada a la sala de cirugía s durante la ventana de tiempo $b > 48$ del día d (tiempo extra).
- $W(i,p) = 1$ si la cirugía i de prioridad p no es programadas durante la semana.
- $\beta C(c,b,g) = 1$ si el cirujano c de categoría g se encuentra trabajando en subutilización en la ventana de tiempo b .
- $TDCO(b,c,i,h) =$ ventana de tiempo b de inicio del descanso del cirujano c , enfermera n al finalizar la cirugía i de la categoría h .
- $TDCF(b,c,i,h) =$ ventana de tiempo b de finalización del descanso del cirujano c , enfermera n al finalizar la cirugía i de la categoría h .
- $TDNO(b,n,i,h) =$ ventana de tiempo b de inicio del descanso de la enfermera n al finalizar la cirugía i de la categoría h .
- $TDNF(b,n,i,h) =$ ventana de tiempo b de finalización del descanso de la enfermera n al finalizar la cirugía i de la categoría h .
- $WNE(i,n,k,b) = 1$ si la enfermera n de categoría k se encuentra asignada a la cirugía i durante la ventana de tiempo $b > 48$.

- $WCE(i,c,g,b) = 1$ si el cirujano c de la categoría g se encuentra asignado a la cirugía i durante la ventana de tiempo $b > 48$.
- $XW(i,n,k,b,d) = 1$ si la enfermera n de la categoría k se encuentra asignada a la cirugía i durante la ventana de tiempo b del día d .
- $YW(i,c,b,d) = 1$ si el cirujano c de la categoría g se encuentra asignado a la cirugía i durante la ventana de tiempo b del día d .
- $AC(i,h,g) = 1$ si el cirujano de categoría g es apto para realizar la cirugía i de categoría h .
- $AN(i,h,k) = 1$ si la enfermera de categoría g es apto para realizar la cirugía i de categoría h .

4.3.4. Fase 1

4.3.4.1. Función objetivo

Maximizar (1)

$$\sum \sum \sum \sum ZW(i, s, b, d)$$

Minimizar (2)

$$\begin{aligned} & \sum \sum ((\sum ES(s, b, d)) * CES(s)) + \sum \sum (\sum BS(s, b, d) * CSS(s)) \\ & + \sum \sum (\sum O(i, p, d) * R(p)) + \sum (\sum W(i, p) * CA(p)) \end{aligned}$$

La función objetivo (1) se encarga de maximizar la cantidad total de cirugías programadas durante cada día, mientras la función objetivo (2) busca minimizar los costos innecesarios de operación, como lo son el uso de horas extra, subutilización de salas, el aplazamiento y no realización de los procedimientos durante el horizonte de programación.

4.3.4.2. Restricciones.

$$\sum \sum ZW(i, b, s, d) \leq Trd \quad \forall s \in S, \forall d \in D \quad (3)$$

$$\sum \sum ZWE(i, b, s, d) \leq Q \quad \forall s \in S, \forall d \in D \quad (4)$$

Las ecuaciones (3) y (4) se encargan de limitar el número total de ventanas de tiempo posibles a usar dentro de un día normal de programación, así, (3) asegura que no se excederá el número de ventanas de tiempo estándar, mientras que (4) restringe el uso de ventanas de tiempo extra.

$$\sum U(i) \leq Trd + Q \quad \forall s \in S, \forall d \in D \quad (5)$$

La restricción (5) establece que ninguna de las cirugías a programar puede tener una duración superior a la cantidad total de ventanas de tiempo disponibles por día.

$$\sum ZW(i, s, b, d) \leq 1 \quad \forall s \in S, \forall d \in D, \forall b \in B \quad (6)$$

$$\sum ZWE(i, s, b, d) \leq 1 \quad \forall s \in S, \forall d \in D, \forall b \in B \quad (7)$$

Las ecuaciones (6) y (7) plantean la imposibilidad de realizar más de una cirugía en una sola sala al mismo tiempo. (6) limita específicamente las ventanas de tiempo estándar, y (7), las ventanas de tiempo extra.

$$TF(i, s, b, d) - TO(i, s, b, d) \leq U(i) \quad \forall i \in I \quad (8)$$

La restricción (8) determina que la duración de una cirugía no puede ser mayor al estándar establecido inicialmente, y que ésta, debe realizarse en su totalidad en ventanas de tiempo consecutivas (no se permite pausar un procedimiento que ya ha sido iniciado).

$$ES(s, d, b) + \beta S(s, d, b) = 1 \quad \forall s \in S, \forall d \in D, \forall b \in B \quad (9)$$

La restricción (9) establece que una sala de cirugía no puede encontrarse en uso normal y en subutilización al mismo tiempo.

$$TO(i, s, b, d); TF(i, s, b, d); J(i, h) \geq 0 \quad \forall i, s, b, d, h \quad (10)$$

$$ES(s, l, b, d); \beta S(s, b, d); Z(s, b, d); \Omega(i, p, d); \quad \forall s, l, b, d, l, p, n, k, g \quad (11)$$

$$ZW(i, s, b, d); ZWE(i, s, b, d); \mu(i, p); X(n, k, b);$$

$$Y(c, g, b) \in \{0,1\}$$

Finalmente, el conjunto de ecuaciones (10) y (11) definen el dominio al que pertenecen las variables, ya sea como enteros positivos, o como variables binarias.

4.3.5. Fase 2

4.3.5.1. Función objetivo.

Minimizar

$$\sum (WCE(c, g, b) * CEC(g) + \beta C(c, g, b) * CSC(g)) + \sum (WNE(n, k, b) * CEN(k)) \quad (12)$$

La función objetivo (12) se encarga de minimizar los costos innecesarios asociados al personal (uso de tiempo extra y subutilización de cirujanos)

4.3.5.2. Restricciones.

$$TDCF(b, c, i, h) - TDCO(b, c, i, h) = RE(i, h) \quad \forall b \in B, \forall i \in I, \forall c \in C, \forall h \in H \quad (13)$$

$$DNF(b, n, i, h) - TDNO(b, n, i, h) = RE(i, h) \quad \forall b \in B, \forall i \in I, \forall n \in N, \forall h \in H \quad (14)$$

Las restricciones (13) y (14) se aseguran que cumpla con la duración del tiempo de descanso obligatorio de los cirujanos (13) y enfermeras (14).

$$TDCO(b, c, i, h) = TF(i, s, b, d) \quad \forall b \in B, \forall i \in I, \forall c \in C, \forall h \in H, \quad (15)$$

$$\forall s \in S, \forall d \in D$$

$$TDNO(b, n, i, h) = TF(i, s, b, d) \quad \forall b \in B, \forall i \in I, \forall n \in N, \forall h \in H, \quad (16)$$

$$\forall s \in S, \forall d \in D$$

Las ecuaciones (15) y (16) establecen que el inicio del tiempo de descanso obligatorio de los cirujanos (15) y enfermeras (16) ocurra justo después de finalizada la cirugía a la cual había sido programado.

$$\sum XW(n, i, b, d) \leq 1 \quad \forall n \in N, \forall b \in B, \forall d \in D \quad (17)$$

$$\sum YW(c, i, b, d) \leq 1 \quad \forall c \in C, \forall b \in B, \forall d \in D \quad (18)$$

Las restricciones (17) y (18) limitan el uso de uso del personal, de tal forma que las enfermeras y los cirujanos solo puedan estar asignados a una única cirugía al mismo tiempo.

$$\sum YW(c, i, b, d) = U(i) \quad \forall i \in I, \forall c \in C, \forall d \in D \quad (19)$$

$$\sum XW(n, i, b, d) = U(i) \quad \forall i \in I, \forall n \in N, \forall d \in D \quad (20)$$

Las ecuaciones (19) y (20) determinan que tanto el cirujano como las enfermeras programadas deben permanecer desde inicio hasta el final del procedimiento, por lo cual no se permite el cambio de personal en el trascurso de la cirugía.

$$AC(i, h, c, g) * YW(c, i, b, d) = 1 \quad \forall i \in I, \forall c \in C, \forall g \in G, \forall b \in B, \forall d \in D, \forall h \in H \quad (21)$$

$$AN(i, h, n, k) * XW(n, i, b, d) = 1 \quad \forall i \in I, \forall n \in N, \forall h \in H, \forall b \in B, \forall d \in D, \forall k \in K \quad (22)$$

Las restricciones (21) y (22) aseguran que el equipo de trabajo asignado para la cirugía cumpla con las habilidades mínimas necesarias para realizar el procedimiento

$$\beta C + YW = 1 \quad \forall c \in C, \forall b \in B, \forall i \in I, \forall d \in D \quad (23)$$

La ecuación (23) se encarga de asegurar que el cirujano no pueda encontrarse laborando en condiciones normales y de subutilización al mismo tiempo.

$$TO(v, i, s, c, n, p); TF(v, i, s, c, n, p); RE(i, h); TDNF(b, n, i, h); \quad (24)$$

$$TDNO(b, n, i, h); TDCF(b, c, i, h); TDCO(b, c, i, h) \geq 0$$

$$AN(i, h, n, k); AC(i, h, c, g); YW(c, i, b, d); XW(n, i, b, d); WCE(c, g, b); \quad (25)$$

$$WNE(n, k, b); Y(c, g, v); X(n, k, v); \beta C(c, g) \in \{0,1\}$$

Finalmente, el conjunto de restricciones (24) y (25) definen el dominio al que pertenecen las variables, ya sea como enteros positivos, o como variables binarias.

5. Algoritmo

El modelo matemático se solucionará aplicando una heurística tipo ACO “*Ant Colony Optimization*”, tomando como referencias investigaciones tales como la de Xiang, Yin y Lim (2015), en donde se plantea la solución del problema mediante el uso de capas (fases), pero añadiendo además algoritmos de selección aleatoria para asegurar la diversificación de las soluciones evaluadas.

Se agregará en las siguientes secciones un pseudocódigo, con el objetivo de facilitar la comprensión de la metodología de solución.

5.1. Definición de clases

Una clase en el lenguaje de programación, hace referencia a un tipo de dato definido por el usuario, y es el principal bloque de desarrollo del algoritmo, así, se definieron 4 clases diferentes.

5.1.1. Clase cirugía

Es el objeto principal para la programación del algoritmo, que cuenta con los siguientes atributos.

- **ID:** es un número único y ordenado que sirve para identificar una cirugía en específico.
- **Grupo:** un número del 1-86 asociado a la cirugía según la clasificación CUPS.
- **Capítulo:** un número de 1 a 14 asociado a la cirugía según la clasificación CUPS.
- **Tiempo:** duración de la cirugía en número de ventanas de tiempo.
- **Prioridad:** nivel de importancia/urgencia dado por el médico para la realización de la cirugía.
- **Fecha:** día propuesto por el médico para la realización de la cirugía.
- **Costo de aplazamiento:** costo de penalización por el aplazamiento de la cirugía.
- **Costo de no realización:** costo de penalización por la no realización de la cirugía en el horizonte de planeación (1 semana).
- **Categoría:** número de 1 a 4 que expresa la relación entre la cirugía y la sala. (Se ahondará más en la explicación de la clase sala).
- **Doctor:** Es un vector al que se agrega el cirujano programado.
- **Enfermeras:** Es un vector al que se añaden las enfermeras programadas.
- **NurNum:** Es un vector que especifica la cantidad de enfermeras de cada tipo necesarias para la realización de la cirugía.

5.1.2. Clase sala.

Se refiere al lugar donde se realizará la cirugía y cuenta con los siguientes atributos:

- **ID:** es un número único y ordenado que sirve para identificar una sala en específico.
- **Categoría:** número de 1 a 4 que expresa la posibilidad de la sala para realizar un tipo de cirugía. Cada categoría se relaciona con un conjunto de capítulos de la siguiente manera:
 - Cat 1= ([7, 8])

- Cat 2= ([1, 6, 13])
- Cat 3= ([2, 9, 10])
- Cat 4= ([3, 4, 5, 11, 12, 14])
- **Costo de subutilización:** Costo de penalización por subutilizar la sala.
- **Costo extra:** Costo de penalización por abrir ventanas de tiempo extra en la sala.
- **BusyVec:** Vector de utilización al que se agrega las cirugías programadas.

5.1.2. Clase doctor

Hace referencia a cada cirujano y cuenta con los siguientes atributos:

- **ID:** es un número único y ordenado que sirve para identificar un cirujano en específico.
- **Capítulo:** un numero de 1 a 14 que relaciona al cirujano con la cirugía y se asocia según la clasificación CUPS.
- **Costo de subutilización:** Costo de penalización por la subutilización del cirujano.
- **Costo extra:** Costo de penalización por el uso del cirujano en tiempo extra.
- **NumSurg:** un número variable que especifica el número de cirugías seguidas que ha realizado.

5.1.3. Clase enfermera

Hace referencia a cada enfermera a programar y cuenta con los siguientes atributos:

- **D:** es un número único y ordenado que sirve para identificar una enfermera en específico.
- **Categoría:** un número de 1 a 3 que especifica la especialización de la enfermera.
- **Costo extra:** costo de penalización por el uso de la enfermera en tiempo extra.

- **NumSurg:** un número variable que especifica el número de cirugías seguidas que ha realizado

5.2. Aleatorización en la generación de clases

El código contiene una sección específica para los datos de entrada, que nos permite modificar los parámetros con los que se correrá el algoritmo, entre los cuales encontramos los vectores de costos de subutilización y uso de tiempo extra, los vectores de definición de cantidad y categorías de salas, doctores y enfermeras, el número máximo de cirugías seguidas que puede realizar el personal antes de un descanso, los tiempos promedio de duración de las cirugías según su capítulo, la relación entre categorías de salas y capítulos de cirugía, y demás parámetros como cantidad de iteraciones y población de hormigas para la solución del modelo. Adicional a esto, durante la creación de los objetos (elementos de cada clase) que componen el escenario, se establecen los factores para la diversificación de los casos a analizar.

Tabla 6.

Código de aleatorización de clases

Aleatorización de Clases	
1	Por cada elemento “i” en el rango de “SurgNum”
2	Definir ID= i
3	Definir cap= cap_def(capProbVec)
4	Definir time= time_def (cap, promTimeVec)
5	Definir prior= random(1,5)
6	Definir date= random(1,6)

En la Tabla 6 se puede ver un ejemplo del funcionamiento de la aleatorización de las cirugías. Se parte de un parámetro definido “SurgNum” que especifica la cantidad total de cirugías a programar, seguido de esto, a cada cirugía se le agrega un ID siguiendo el orden de creación, se genera un capítulo “cap” según un vector de probabilidad, el “time” o duración, definido por el capítulo y un vector de relación tiempo-capítulo, por último, la generación de la prioridad “prior” y el día previsto de realización “date” según un número aleatorio que se mantiene entre los rangos de movimiento de cada variable.

Este proceso de aleatorización no es exclusivo de la clase cirugía, todos los elementos de las clases “doctores” y “enfermeras” cuentan también con al menos un elemento aleatorizado que le permite al modelo a enfrentarse a una cantidad considerable de casos cercanos a los que se podrían encontrar en la vida real.

5.3. Construcción de objetos

El proceso de construcción de las 4 clases de objetos (cirugías, salas, enfermeras y doctores) siguen la estructura de la Tabla 7.

Tabla 7.

Estructura para la creación de clases

Crear Clases

- 1 Crear Vector_clase = []
 - 2 Crear Contador k= 1
 - 3 Por cada elemento en el vector de origen:
Asignar atributos (ID, Costos, categorías, etc.)
-

4 Añadir objeto al Vector_Clase

5 $K += 1$

Se inicia con la creación de un contador “k” y un vector vacío que varía según la clase, “roomVec” para las salas, “docVec” para los cirujanos, “nurVec” para las enfermeras y “SurgVec” para las cirugías. Se continúa con la lectura de cada elemento de un vector origen que define la cantidad de objetos a crear mediante un comando **for**, seguido a esto, se realiza la asignación de atributos, llevando a cabo el proceso de aleatorización en caso de ser necesario o mediante lectura directa de los parámetros definidos. Finalmente se agregan el objeto al vector de clase y se aumenta en uno el contador, el cual funciona como base para la generación del ID.

5.4. Solución del escenario

Para iniciar con este proceso, se escriben pequeños bloques de código que se encargan de realizar tareas específicas necesarias para la programación de las cirugías. Así, se inicia con la creación de un vector vacío que desarrollará el papel de los días de programación. Mediante un comando **for**, el algoritmo lee la cantidad X de días máximos definidos en los parámetros, y realiza X copias del vector vacío, permitiendo así, variar fácilmente la cantidad total de días a programar, que serán nombrados con un índice numérico iniciado en cero. Debido a que el horizonte de planeación de este proyecto está definido en una semana, se establece el parámetro “limdays = 6”. Así, el lunes estará definido con el índice cero [0] y el sábado con el índice cinco [5]. Cabe aclarar, que el algoritmo no es capaz de discriminar días, por lo cual, en un horizonte de programación más largo, se seguirían generando índices más grandes, caso en el cual, se debería realizar una reinterpretación de los índices mostrados (Tabla 8).

Tabla 8.

Estructura de la creación de días

Creación de días	
1	Crear Vector_Dia = []
2	Por cada elemento en el rango de “limdays”
3	Agregar un vector vacío ([]) al Vector_Dia

El siguiente bloque de código se encarga de separar las cirugías generadas según el día previsto, para esto, se recoge el valor del atributo “Date” y se agrega la cirugía al vector día correspondiente tal como se muestra en la Tabla 9.

Tabla 9.

Estructura de la separación de cirugías por día

Separación de las cirugías por día	
1	Por cada elemento en el rango de “SurgNum”:
2	Definir dt = valor “Date” de la cirugía i ubicada en SurgVec
3	Agregar cirugía i al vector día [dt-1]

Se continúa con el ordenamiento diario de las cirugías según la prioridad de forma descendente, así, la lista generada en el vector día iniciará con las cirugías de prioridad 5, facilitando su programación posterior. Seguido a esto, se separan las salas según su categoría, en un proceso similar a los anteriores, iniciando con la creación de un vector vacío “roomCats” que servirá como

contenedor de los vectores referentes a cada una de las cuatro categorías. Al igual que con el vector de días, los índices iniciarán en cero, por lo que una sala de categoría uno se mostrará por el momento como una sala de índice cero [0]. Luego el algoritmo lee el valor “cat” de cada sala guardada en el roomVec y procede a agregarlos al vector de categoría respondiente tal como se muestra en la Tabla 10.

Tabla 10.

Estructura de la separación de salas por categoría.

Separación de salas por categoría	
1	Crear vector “roomCats” = []
2	Por cada elemento almacenado en el parámetro de categorías:
3	Agregar un vector vacío ([]) al Vector “roomCats”
4	Por cada elemento almacenado en el vector roomVec:
5	Definir rc = valor “cat” de cada elemento leído
6	Agregar elemento al vector “roomCats[rc-1]”

Posteriormente, el algoritmo realiza la programación de las cirugías en una sala capacitada para llevar a cabo este procedimiento, como se puede observar en la Tabla 11. Inicia iterando cada sala según el día, se agrega una variable de tiempo “sumt” que evitará que se programen más cirugías de las que una sala es capaz de atender diariamente y un par de vectores vacíos para separar la programación en los tiempos estándar y extra de cada sala. Se compara la categoría de cada cirugía con la de la sala y en caso de que estas coincidan y de contar con tiempo suficiente para la programación, se agrega la cirugía al vector contenedor (estándar o extra según corresponda). Cabe

aclarar que en durante este paso, el vector de tiempo extra, se utilizará como contenedor para todas las cirugías que no quepan en el tiempo estándar y que puedan ser reprogramadas de cualquier otra manera (subutilización, aplazamiento, no realización o manteniéndose en hora extra), por lo que no se acota un tiempo máximo.

Tabla 11.

Estructura de la programación de cirugías

Programación de cirugías	
1	Para cada elemento en el vector contenedor de cirugías “roomVec”:
2	Definir rcat como una copia del dato “cat” de cada sala leída
3	Para cada elemento en el vector contenedor de días:
4	Definir contador “sumt”
5	Crear vectores vacíos para tiempo estándar y tiempo extra
6	Para cada cirugía almacenada en el vector dia:
7	Si la categoría de la cirugía es igual al valor de “rcat”:
8	Si “sumt” + duración de la cirugía \leq tiempo máximo estándar:
9	Aumentar sumt en el tiempo de duración de la cirugía
10	Agregar la cirugía al vector vacío de tiempo estándar
11	Si “sumt” + duración de la cirugía $>$ tiempo máximo estándar
12	Agregar la cirugía al vector vacío de tiempo extra
13	Agregar vectores al BusyVec de la sala

Para el cálculo de los costos finales, el algoritmo toma todos los elementos almacenados en el vector de tiempo extra y ejecuta las funciones de subutilización y aplazamiento de las cirugías generando un costo para cada escenario, guardándolos junto con los costos de tiempo extra y no realización dentro de otro vector así:

$$\text{Vector} = [\text{índice 0, índice 1, índice 2, índice 3}]$$

Donde el índice 0 representa el valor de no llevar a cabo el procedimiento, el índice 1 representa el valor de realizarlo durante ventanas de tiempo extra, el índice 2 representa el valor de aplazar el procedimiento 1 día y el índice 3 representa el valor de subutilizar una sala para realizar la cirugía. Seguido de esto, compara cada dato eligiendo el menor y realizando una acción determinada según el índice, aumentando el contador del costo total en el valor elegido y eliminando la cirugía de la lista para evitar que sea tenida en cuenta varias veces en la programación.

El proceso de asignación del personal (ilustrado en la Tabla 12) se realiza de manera similar al de las salas. Mediante el uso de varios comandos “for” se leen los vectores contenedores de días, salas y tiempos (estándar y extra). Para cada cirugía se llama a las funciones de selección “DocSubCostFunc” y “NuSubCostFun”, las cuales se encargan de retornar los objetos “doctor” y “enfermeras” con menor costo asociado (teniendo en cuenta los requisitos mínimos de suficiencia y los costos de subutilización asociados). Finalizada la selección del personal, se añaden a la cirugía y se aumenta el valor de la variable de costo total.

Tabla 12.

Estructura de la asignación del personal

Asignación del personal	
1	Para cada día:
2	Para cada sala:
3	Para cada cirugía en tiempo estándar:
4	Llamar función de selección de doctor
5	Si encuentra un doctor:
6	Agregar el doctor a la cirugía
7	Sumar el costo del doctor al costo total
8	Llamar la función de selección de enfermeras
9	Si encuentra enfermeras:
10	Agregar las enfermeras a la cirugía
11	Sumar el costo de las enfermeras al costo total
12	Para cada cirugía en tiempo extra
13	Llamar función de selección de doctor
14	Si encuentra un doctor:
15	Agregar el doctor a la cirugía
16	Sumar el costo del doctor al costo total
17	Sumar al costo extra del doctor al costo total
18	Llamar la función de selección de enfermeras
19	Si encuentra enfermeras:

- | | |
|----|---|
| 20 | Agregar las enfermeras a la cirugía |
| 21 | Sumar el costo de las enfermeras al costo total |
| 22 | Sumar el costo extra de las enfermeras al costo total |
-

Debido a que el equipo de trabajo se añade directamente a la cirugía, y estas por su parte ya se encuentran organizadas dentro del “BusyVec” de cada sala, se da por terminado el proceso de programación del escenario planteado y se continúa con la ejecución de la heurística de colonia de hormigas para la solución final del modelo

5.5. Algoritmo de variación de hormigas

El proceso anteriormente descrito (a excepción de la generación de las cirugías) representa una función única del código denominado “eval_method_int”. Esta función retorna el equivalente a una solución inicial con una combinación de recursos específica. Sin embargo, esta resulta ser tan solo una de las soluciones factibles del problema, por lo que se hace necesario crear variaciones dentro de esta misma solución de tal forma que se evalúen nuevas combinaciones, aumentando así la probabilidad de encontrar una mejor.

Para llevar esto a cabo, se crea una segunda función denominada “eval_method_out” que se encargará repetir el proceso tantas veces como sea de grande la población definida de hormigas, reiniciando cada vez la generación aleatoria de salas, doctores y enfermeras mediante la asignación de un rango (valor máximo y mínimo) por el cual se pueda mover el algoritmo, asegurándose así que se tomen en cuenta la mayor cantidad de combinaciones posibles. Este segmento de código guarda además cada uno de los resultados en el vector de evaluación, que será utilizado posteriormente para la filtración y elección de la mejor solución.

5.5.1. Algoritmo de filtrado inicial

Esta función recibe el vector de evaluación resultante de la ejecución de la función “eval_method_out”. Estos resultados se redefinen como una lista teniendo en cuenta únicamente el costo final, se reorganizan de forma aleatoria y se divide en dos mitades “eval1” y “eval2”, cada uno con la misma cantidad de elementos y sin un ordenamiento específico.

Teniendo en cuenta que se define el porcentaje de filtración en un 50%, se procederá con la selección de las mejores soluciones mediante comparación de los costos entre los índices (elementos) de las dos listas una $X/2$ cantidad de veces, donde X es el número de soluciones retornadas por el “eval_method_out” guardando los elementos con menor costo en un vector transitorio denominado “filtrados”.

5.5.2. Algoritmo de eliminación

Una forma simplificada de visualizar cada solución generada por el algoritmo se presenta en la Figura 2.

Vector de salas	Vector de doctores	Vector de enfermeras	Costo final
-----------------	--------------------	----------------------	-------------

Figura 2. Visualización del vector solución

El filtrado inicial se encarga de generar una lista con una cantidad fija de soluciones basado en la comparación de los valores de la última posición del vector, es decir, el costo final. Para la siguiente fase de la solución resulta interesante realizar cambios en los demás elementos sin tener en cuenta el costo, por lo cual se decide eliminar este factor y mantener tan solo los vectores de salas, doctores y enfermeras.

El proceso de eliminación resulta sencillo debido a la distinción con respecto a los demás elementos de la solución, mientras que las tres primeras casillas muestran un conjunto de listas que representan objetos complejos, la última muestra tan solo un valor numérico, un factor esencial para el proceso de filtrado pero que facilita al mismo tiempo su discriminación y por lo tanto su eliminación del vector.

Tabla 13.

Estructura de la eliminación del costo

Eliminación del costo	
1	Para cada elemento de la población:
2	Si el elemento [-1] (es decir, el último índice del vector) no es una lista:
3	Remove el elemento del vector

Se debe tener en cuenta que la población mencionada en la Tabla 13 hace referencia a cada hormiga generada por el algoritmo y que esta a su vez, representa una solución posible del problema.

5.5.3. Algoritmo de combinación

En este punto se procede a realizar modificaciones dentro de la solución. Para esto el algoritmo recoge el vector de filtrados resultante del primer proceso, llama a la función de eliminación para remover los costos finales del vector, repite el procedimiento de reorganización de forma aleatoria y lo divide de nuevo en dos subgrupos. Posteriormente reúne a dos hormigas distintas (una de cada grupo), escoge un punto aleatorio de la combinación de elementos de la solución y lo intercambia

con la segunda hormiga, de esta manera se logra ampliar el espacio de búsqueda y evaluar alternativas de solución diferentes.

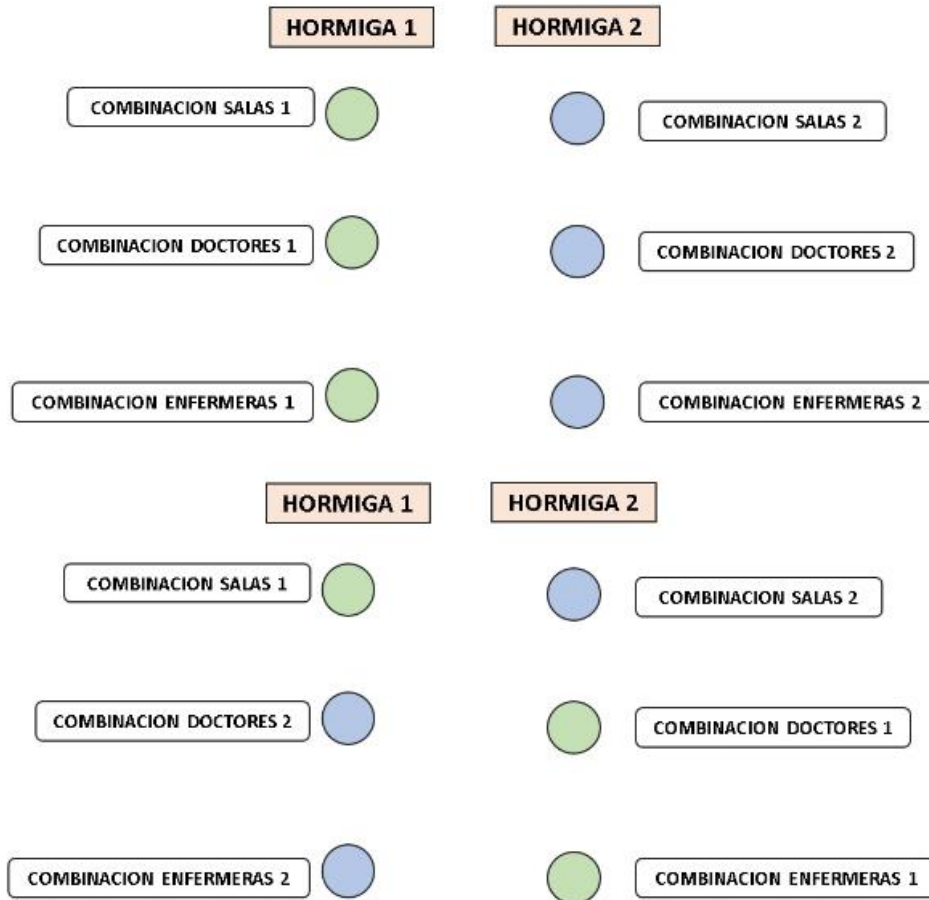


Figura 3. Proceso de combinación de elementos

La Figura 3 muestra de manera simplificada el intercambio de información/rutas entre las hormigas. Como se puede apreciar, el intercambio no es tan solo de un elemento específico de la combinación, sino de todo el vector siguiente al punto elegido, y no se limita únicamente a conjuntos completos, sino a elementos específicos dentro de estos conjuntos, así, es posible intercambiar toda la combinación de doctores y enfermeras o desde un punto interno tal como se muestra en la Figura 4.

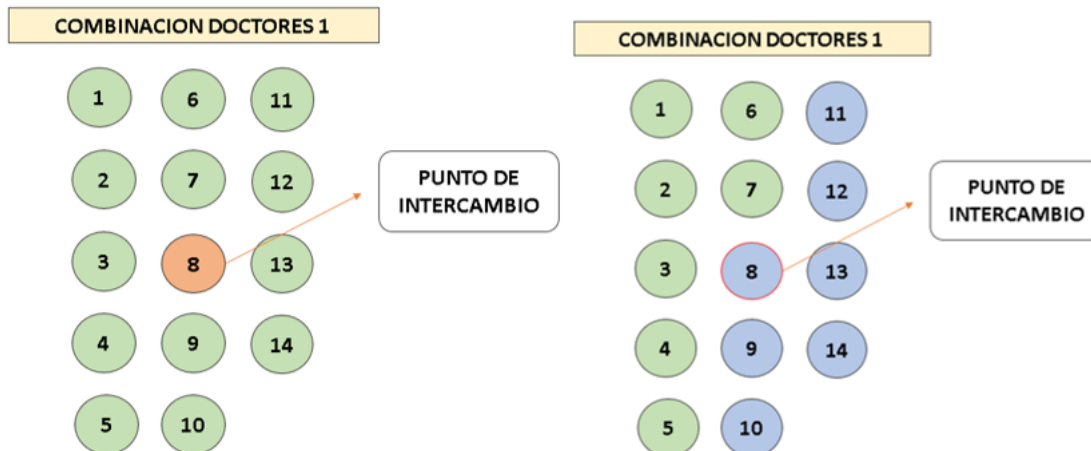


Figura 4. Punto interno de intercambio

Debido a que el algoritmo trabaja con 14 categorías diferentes de doctores, cada una con una cantidad específica de cirujanos, es posible que el elemento de intercambio se encuentre dentro del conjunto general, así, la primera hormiga mantendrá la solución del vector original hasta este punto y reemplazará las combinaciones siguientes con la información obtenida de la segunda hormiga. Como el punto de intercambio se elige de manera aleatoria, es posible que este proceso se realice desde cualquiera de los conjuntos generales (salas, doctores o enfermeras).

Este mecanismo de combinación se repetirá tantas veces como hormigas existan en cada uno de los subgrupos, generando una nueva lista de posibles soluciones que serán guardadas en un vector transitorio denominado “combinados”

5.5.4. Algoritmo de búsqueda local

Finalmente, el modelo le permite al programador definir una probabilidad de que cada elemento almacenado en el vector de “combinados” pase por una última variación donde las hormigas realizan cambios en partes específicas de su solución. Estos cambios, se efectúan de manera

aleatoria al igual que en los procesos anteriores, y tiene como finalidad realizar una búsqueda de alternativas de solución en un espacio más reducido.

Tabla 14.

Estructura del código de Búsqueda Local

Código de Búsqueda Local

- 1 Variable = copia del vector “combinados” en forma de lista
 - 2 Para cada elemento en el rango de la variable:
 - 3 Si un valor random generado entre cero y uno > parámetro de selección
 - 4 Llamado a la función de búsqueda local.
-

Este procedimiento, descrito en la Tabla 14, inicia con el llamado del vector “combinados” y su transformación en una lista que será almacenada en una variable. Posteriormente se genera un número decimal aleatorio entre 0 y 1 para cada elemento y se compara con el valor definido por el programador, de tal forma que si es mayor esa hormiga continuará con la búsqueda local, y en caso contrario, la solución a evaluar será la resultante del proceso de combinación. El algoritmo elegirá un punto aleatorio del vector solución que será donde se realice la variación y procederá a intercambiar la información de sus elementos internos, devolviendo al final a la hormiga con un vector solución diferente.

Terminados todos los procesos de variación, se repite la programación y evaluación de las combinaciones de cada hormiga, eligiendo la mejor (en función de su costo) y guardándola en el vector de soluciones finales.

6. Evaluación del algoritmo

6.1. Descripción de instancias

Las instancias generadas para la evaluación del algoritmo, se basaron en los datos mostrados por Bejarano (2011) en el artículo “evaluación cuantitativa de la eficiencia en las salas de cirugía” donde especifica las condiciones de funcionamiento de las salas quirúrgicas de la Clínica Rafael Uribe Uribe de Cali durante el periodo de julio de 2010 a junio de 2011. Si bien este estudio no pretende formular un modelo matemático para la optimización de recursos, si presenta varias similitudes con los escenarios planteados por diversos autores revisados en el análisis bibliográfico en lo relacionado a las cantidades totales de los recursos a programar (salas, cirujanos, enfermeras) y las especialidades de cada uno de ellos. Permittiéndonos de esta manera, trabajar con datos de un escenario nacional, y que no presenta mayores diferencias con respecto a los estudios que abordan temas más relacionados con el propuesto en este proyecto.

Se planea entonces analizar un escenario de programación semanal con un total de 120 cirugías distribuidas de manera aleatoria a lo largo de la semana. Las duraciones de las cirugías (en ventanas de tiempo) están definidas en la Tabla 15 y se obtienen mediante los tiempos promedio de los 5 procedimientos más comunes de cada tipo más un adicional de 4 ventanas de tiempo fijas que representan el alistamiento de las salas. Las probabilidades de aparición definidas para cada tipo se basan en lo mostrado por Costa A, Jr (2017) en su artículo “Assessment of operative times of multiple surgical specialties in a public university hospital”, donde se analizan los tiempos operativos de más de 8.000 procedimientos durante enero del 2011 y enero del 2012.

Tabla 15.

Parámetros de cirugías

Parámetros														
Categoría	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Duración	28	12	6	8	8	18	20	16	16	20	6	8	12	12
Probabilidad de aparición	0.0	0.05	0.0	0.08	0.08	0.02	0.0	0.02	0.15	0.05	0.05	0.0	0.16	0.0
	2	4	8	6	6	8	4	5	9	8	8	7	6	7

Casanovas Biosca, N et al (2013) nos muestra en su artículo “La gestión quirúrgica según niveles de complejidad” una distribución de personal basada en la clasificación de las intervenciones según los 4 niveles de dificultad presentados por la guía preoperatoria del National Institute for clinical excellence. Si bien esta guía, toma en consideración la posibilidad de utilizar varios cirujanos y demás personal de apoyo como anestesiólogos y auxiliares para cada cirugía (factores que no se aplicaron en este proyecto), si sirve como una base para la definición de requerimientos del personal, dando como resultado la distribución de la Figura 5.

CATEGORIA		ENFERMERAS		
		1	2	3
CIRUGIAS	1	2	1	2
	2	2	1	1
	3	1	1	1
	4	0	1	1

Figura 5. Requerimiento de enfermeras por categoría

Teniendo la enfermera de categoría 1 el rol de primer ayudante, la de categoría 2 el rol de instrumentalista y la de categoría 3 el rol de enfermera circulante. En todos los casos, se utiliza tan solo un cirujano con la misma especialidad/ tipo de la cirugía a realizar.

Para la definición de los costos, se utilizaron los datos obtenidos de la página “salary.com”, la cual se especializa en la recolección análisis y exposición de datos estadísticos relacionados con el mercado laboral, incluyendo la asignación de salarios. Aplicando los respectivos recargos derivados del uso de horas extra definidos por el ministerio de trabajo para el año 2019, se establecieron los valores de la Figura 6 para las enfermeras.

Categoría	Descripción	Costo por ventana de tiempo extra
1	Primer asistente	\$ 8,4157
2	Instrumentalista	\$ 6,7422
3	Circulante	\$ 5,7838

Figura 6. Costos de enfermeras

En el caso de los cirujanos, su costo no está tan solo definido por su especialización, sino también por su categoría (nivel de experiencia), por lo que se fijaron los valores de la Figura 7 para el uso de ventanas de tiempo extra.

ESPECIALIZACIÓN	CUPS	< 3 AÑOS	3-10 AÑOS	> 10 AÑOS
SISTEMA NERVIOSO	1	\$ 50,3786	\$ 52,8976	\$ 55,4165
SISTEMA ENDOCRINO	2	\$ 21,3542	\$ 22,4219	\$ 23,4896
SISTEMA VISUAL	3	\$ 32,2049	\$ 33,8151	\$ 35,4253
SISTEMA AUDITIVO	4	\$ 34,5486	\$ 36,2760	\$ 38,0035
NARIZ, BOCA Y FARINGE	5	\$ 34,5486	\$ 36,2760	\$ 38,0035
SISTEMA RESPIRATORIO	6	\$ 29,8611	\$ 31,3542	\$ 32,8472
SISTEMA CIRCULATORIO	7	\$ 42,0781	\$ 44,1820	\$ 46,2859
SISTEMA HEMATICO Y LINFATICO	8	\$ 34,3750	\$ 36,0938	\$ 37,8125
SISTEMA DIGESTIVO	9	\$ 37,5000	\$ 39,3750	\$ 41,2500
SISTEMA URINARIO	10	\$ 37,0660	\$ 38,9193	\$ 40,7726
SISTEMA REPRODUCTOR MASCULINO	11	\$ 42,4045	\$ 44,5247	\$ 46,6450
SISTEMA REPRODUCTOR FEMENINO	12	\$ 29,0799	\$ 30,5339	\$ 31,9878
SISTEMA OSTEOMULCULAR	13	\$ 45,6931	\$ 47,9778	\$ 50,2625
SISTEMA TEGUMENTARIO	14	\$ 39,4965	\$ 41,4714	\$ 43,4462

Figura 7. Costos de cirujanos

6.2. Experimentación

Tanto la cantidad de iteraciones como la población de hormigas a utilizar resultan un aspecto demasiado variable en las investigaciones anteriores. Xiang, Yin, & Lim. (2015) definen un total de 150 iteraciones basado en la convergencia de resultando mediante el testeado de su propio algoritmo y Bellkhamisa, Jarboui y Masmoudi (2018) definen una población de mil individuos y 500 iteraciones. Tomando esto en consideración, se decide realizar un ensayo inicial para analizar el comportamiento del algoritmo con la variación de estos parámetros y establecer la mejor combinación a utilizar.

En el primer conjunto de pruebas se busca analizar la variación de los resultados obtenidos mediante el cambio de la población de hormigas. Para esto se mantiene una cantidad fija de 100 iteraciones y 120 cirugías para los seis días de programación. Las combinaciones de salas, doctores y enfermeras se definen con límites altos (12, 36 y 72 respectivamente) con el fin de darle mayor libertad al algoritmo en la búsqueda de soluciones. Debido al factor de aleatoriedad presente en el

algoritmo se realizan 3 repeticiones de cada una de las pruebas, arrojando los resultados de la Figura 8.

POBLACION	TIEMPO TOTAL (MIN)	COSTO FINAL	DESV ESTANDAR	DESVIACION %
50	6:15	561.511,47	144.262,62	26.20%
	5:58	721.343,86		
	5:40	368.493,84		
100	11:27	376.883,16	37.248,85	10.60%
	11:12	298.483,50		
	11:34	378.103,20		
200	23:45	383.993,96	32.741,00	9.69%
	23:34	314.934,50		
	23:46	314.151,75		
300	36:55:00	355.018,50	20.267,76	6.08%
	35:01:00	337.632,48		
	35:40:00	306.053,73		
400	48:19:00	361.155,34	15.899,09	4.57%
	47:28:00	325.579,87		
	47:12:00	357.090,00		
500	58:16:00	342.445,30	6.172,06	1.80%
	59:45:00	333.464,43		
	58:14:00	348.487,31		

Figura 8. Pruebas de variación en la población de hormigas

El segundo conjunto de pruebas presentado en la Figura 9 mantiene los mismos parámetros que el anterior, salvo que en este se busca analizar los resultados obtenidos mediante la variación del número de iteraciones. Inicialmente se había planteado mantener el mejor valor de población resultante del primer experimento (500 hormigas), pero debido a que estas combinaciones abarcaban tiempos de ejecución muy altos, se optó por fijar el valor de la población en el mínimo testeado, es decir, 50 hormigas, realizando 3 repeticiones para cada prueba.

ITERACIONES	TIEMPO TOTAL (MIN)	COSTO TOTAL	DESVIACION ESTANDAR	DESVIACION %
100	5:43	383.078,25	246.010,92	38.60%
	5:55	558.481,82		
	6:22	970.050,87		
200	11:04	396.790,42	146.695,55	31.23%
	11:31	337.929,97		
	11:18	674.345,08		
300	17:39	307.389,33	59.070,97	15.83%
	18:37	450.644,05		
	17:43	361.388,04		
400	21:35	420.192,58	42.023,35	11.60%
	23:06	344.971,56		
	24:31:00	321.728,33		
500	30:33:00	353.531,80	23.145,24	6.16%
	28:56:00	407.440,92		
	30:09:00	365.287,18		
600	38:04:00	339.555,57	33.899,67	10.55%
	37:38:00	350.478,32		
	35:52:00	273.729,74		

Figura 9. Pruebas de variación de iteraciones

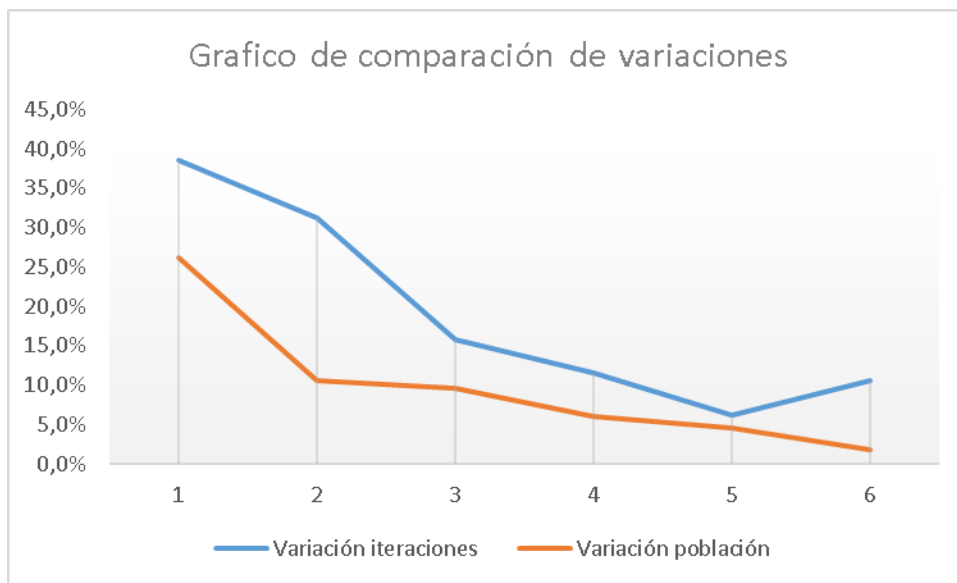


Figura 10. Gráfico de comparación de los porcentajes de variación

Como podemos observar en la Figura 10, existe una clara tendencia a la disminución de la desviación estándar conforme se aumenta la población de las hormigas. Un caso similar ocurre con la gráfica de variación de iteraciones, aunque con una pendiente más marcada. También podemos observar en esta última, que hay un aumento en la desviación con la combinación de 600 iteraciones, por lo que se decide realizar una prueba adicional con una cantidad superior y verificar de esta manera si se mantiene la tendencia (Figura 11).

ITERACIONES	TIEMPO TOTAL (MIN)	COSTO TOTAL	DESVIACION ESTANDAR	DESVIACION %
700	39:04:00	265.107,05	11.099,29	3.97%
	42:10:00	292.174,10		
	41:40:00	280.855,37		

Figura 11. Prueba adicional con variación de iteraciones

El resultado muestra de nuevo la disminución del valor de desviación estándar, y debido que el aumento mostrado con las 600 iteraciones es consecuencia de un dato atípico, se le atribuye la presencia del mismo a la alta aleatoriedad del modelo y se decide no tenerlo en cuenta para la elección de la combinación definitiva.

En la selección de la combinación final se tuvo en cuenta la duración de la prueba y el porcentaje de desviación. Para esto hay que tener presente que estos tiempos de ejecución varían de forma exponencial conforme al aumento de ambos factores (con un tiempo cercano a las 6 horas utilizando una combinación de 500 hormigas y 700 iteraciones) así que se decide usar un arreglo de 500 y 500, con el que se consigue un porcentaje de desviación menor al 5%. También se debe considerar que los tiempos de ejecución del algoritmo variarán en los siguientes experimentos debido a las diferentes mezclas de parámetros a utilizar en cada uno (cantidad de cirugías, salas,

doctores y enfermeras) siendo lo esperado que estos se acorten debido a la disminución de la flexibilidad.

Para la siguiente fase de experimentación, se definieron los siguientes escenarios:

1. En el primer escenario se busca analizar el funcionamiento del modelo sin ninguna alternativa de programación, esto es, eliminando la posibilidad de subutilización, aplazamiento y uso de tiempos extra, manteniendo tan solo la programación en el tiempo estándar y la no realización de las cirugías. Esto se logra aumentando en gran medida sus costos hasta el punto de que el algoritmo no los tome como una opción viable. Esta prueba se repite 5 veces y se usan tan solo 4 salas (1 para cada categoría), una lista de 50 cirugías a programar, 14 cirujanos y 20 enfermeras (Apéndice B).
2. En la segunda prueba se mantiene la restricción de uso de alternativas de programación, pero aumentando la cantidad total de cirugías (100) y salas (8 salas, 2 para cada categoría), esto para analizar si existen cambios significativos en el funcionamiento del modelo para escenarios más robustos. Al igual que el primer experimento, se realizan 5 repeticiones (Apéndice C).
3. En el tercer escenario se busca analizar las mejoras al costo final producidas por la inclusión de alternativas de programación (subutilización, aplazamiento y uso de tiempo extra), por lo que inicialmente se realiza una programación con parámetros similares a los del primer experimento, y posteriormente se habilitan las opciones de programación una a una para verificar su impacto, así, al finalizar se obtiene el resultado para todas las combinaciones posibles (sin tener en cuenta ninguna alternativa, habilitando solo tiempo extra, solo subutilización, solo aplazamientos y las combinaciones entre estas). En esta prueba también se realizan 5 repeticiones (Apéndice D)

4. En el cuarto escenario se busca analizar el comportamiento del modelo forzando la saturación de las salas y encontrar los posibles cuellos de botella, por lo que se plantea la programación de 120 cirugías en 4 salas (1 para cada categoría), manteniendo habilitadas todas las alternativas de programación y aumentando la capacidad del personal a 28 cirujanos y 30 enfermeras. Esta prueba se repite 3 veces (Apéndice E).
5. En el último escenario, se toma la misma lista de cirugías planteadas en el experimento anterior y se aumentan el número de salas a 8 (2 para cada categoría). También se adicionan 14 cirujanos, para un total de 42. Estos cambios permiten analizar el uso de los recursos por parte del algoritmo en un escenario similar a los definidos en los trabajos de investigación consultados durante el análisis bibliográfico. Esta prueba se repite 3 veces (Apéndice F).

7. Análisis de Resultados

7.1. Visualización de resultados

En la Figura 12 se puede apreciar un ejemplo de la tabla en Excel que se decidió utilizar para la visualización de las programaciones resultantes, en ella podemos ver la división vertical de las ventanas de tiempo y la distribución horizontal de las salas pertenecientes a cada día de la semana.

	LUNES				MARTES				MIERCOLES			
	1	2	3	4	1	2	3	4	1	2	3	4
1												
2												
3												
4												
5												
6												
7												
8												
9												
10												

Figura 12. Tabla de visualización de la programación

Las cirugías se identifican con un código de colores según su prioridad (rojo, amarillo, azul, verde y gris) para las prioridades 5,4,3,2 y 1 respectivamente, y se agrupan en la cantidad de ventanas de tiempo correspondientes para cada procedimiento. Dentro de cada bloque de cirugía se señala la información básica (ID, Categoría, cirujano asignado y la experiencia del mismo). En caso de que el tiempo establecido (estándar y extra) no sea suficiente para la realización de una cirugía al 100%, esta no se muestra en la tabla, sino que se agrega en una lista aparte que señala los procedimientos no realizados por día y categoría de sala. En la Figura 13 se puede ver un ejemplo de una de las programaciones realizadas por el modelo, nótese que se sigue la prioridad establecida por el código de colores mencionado anteriormente, ubicando las cirugías más urgentes siempre al inicio del día y finalizando con aquellas de prioridad más baja y, por lo tanto, con menores costos de penalización. Cabe aclarar que este ordenamiento es individual para cada sala, por lo que es posible que al inicio del día se programen cirugías de prioridad menor a las de otras salas, siempre y cuando en la lista de cirugías relacionadas a ese día, no existan otras más urgentes.

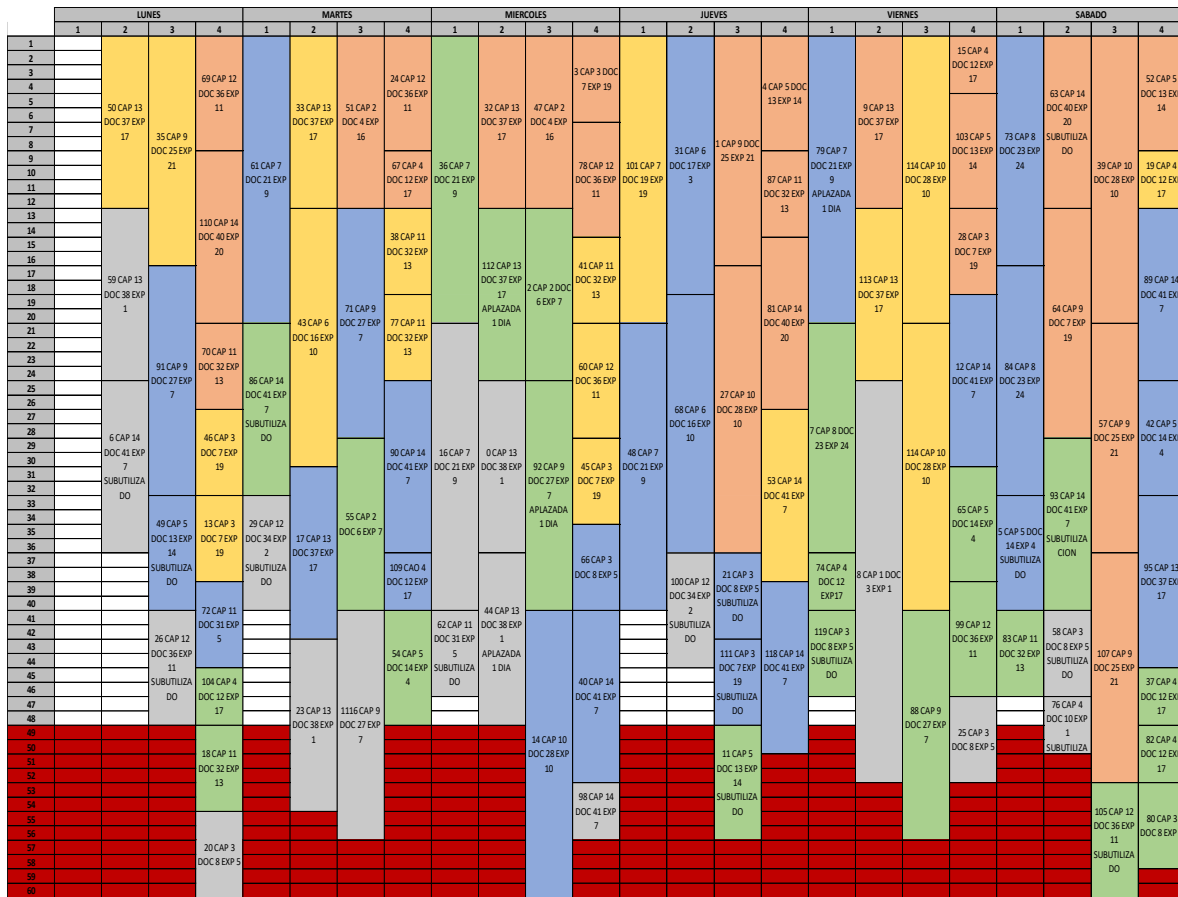


Figura 13. Ejemplo de la programación semanal

7.2. Análisis de los escenarios iniciales

En los primeros escenarios nos encontramos con una programación restringida (sin posibilidad de aplazar, usar tiempo extra o subutilizar), esto para enfocarnos en el uso del tiempo estándar de cada sala y los porcentajes de no realización. Con estos datos, podemos construir un escenario inicial con el cual sea posible realizar las comparaciones referentes a cada mejora posible del modelo. En la Figura 14 podemos ver los resultados de las primeras cinco repeticiones, las cuales representan la programación de 50 cirugías en 4 salas (1 por categoría).

	Cirugías Realizada	Cirugías no realizadas	% uso semanal de tiempo estándar	% NR
1	40	10	44,27%	20,00%
2	47	3	44,79%	6,00%
3	46	4	46,53%	8,00%
4	42	8	42,01%	16,00%
5	46	4	47,05%	8,00%
PROM	44,2	5,8	44,93%	11,60%

Figura 14. Resultados del primer escenario

Como se puede observar, el porcentaje total de uso de ventanas de tiempo no supera el 50% en ninguna de las 5 repeticiones, esto puede considerarse como normal tomando en consideración el tamaño del escenario, sin embargo, también es notorio que incluso con 50 cirugías (menos de la mitad planteada para la prueba final), se presenta un gran porcentaje de procedimientos no realizados, por lo que se hace evidente la necesidad de utilizar alguna o varias de las opciones de programación propuestas en este modelo.

	% utilización tiempo estándar/ día					
	lunes	martes	miercoles	jueves	viernes	sabado
PROM	47,08%	42,50%	54,16%	39,37%	46,24%	40,20%

Figura 15. Uso promedio de salas por día

Si se analiza más detalladamente el uso de las salas, como lo mostrado en la Figura 15, podemos observar que en promedio se cuenta con una ocupación diaria bastante similar, con una desviación cercana al 5%, por lo que se descarta que la asignación aleatoria de las fechas de realización resulte en la acumulación de cirugías en días específicos que favorezca la no realización de estos procedimientos. Sin embargo, si es posible ver un desbalance de la programación si se filtra según la categoría de la sala, sobresaliendo notablemente la categoría 3 por encima de las demás (Figura 16).

	% utilización estándar/categoría			
	1	2	3	4
PROM	21,67%	48,75%	63,33%	45,97%

Figura 16. Uso promedio de las salas por categoría

Tomando en consideración lo anterior, se efectúa el análisis de la no realización de las cirugías basándose en la categoría a la que pertenecen. En la Figura 17 se muestran los resultados obtenidos, evidenciando que a pesar del mayor porcentaje de utilización de la sala de categoría 3, no existe una diferencia significativa en comparación con las salas de categoría 2 y 4, al menos para la configuración actual de variables.

	NR POR CATEGORIA			
	1	2	3	4
1		5	2	3
2		1		2
3		1	3	
4		3	3	2
5			2	2
TOTAL	0	10	10	9
PROMEDIO	-	2,5	2,5	2,25

Figura 17. Cirugías no realizadas según categoría escenario

En el segundo escenario se plantea agregar una sala extra para cada categoría sin tener en cuenta ninguna opción adicional para la programación (aplazamiento, tiempo extra y subutilización), sumado a esto, se aumenta en igual medida la cantidad total de cirugías a programar (100 cirugías) y se analizan los cambios que se pudieran presentar con respecto a las pruebas anteriores. En la Figura 18 se muestran los resultados de las 5 repeticiones realizadas con los nuevos parámetros, donde se destaca una leve mejora en la utilización de las salas (9%) y una disminución considerable en el porcentaje final de no realización, mostrando una mejora de casi el 70% con respecto al primer escenario.

	Cirugías Realizada	Cirugías no realizadas	% uso semanal de tiempo estándar	% NR
1	94	6	51,65%	6,00%
2	91	9	47,40%	9,00%
3	99	1	51,56%	1,00%
4	100	0	48,44%	0,00%
5	98	2	47,83%	2,00%
PROM	96,4	3,6	49,38%	3,60%

Figura 18. Resultados segundo escenario

En lo referente al análisis de utilización diaria y por categoría no se evidencian cambios significativos. Los porcentajes de utilización filtrados por día continúan con valores cercanos al 50% y con una desviación de 4,6%, por otro lado, se mantiene la tendencia con respecto a las categorías, siendo las salas de categoría 3 las de mayor uso (sobrepasando el 60%) y las de categoría 1 con la menor utilización (por debajo del 20%).

Finalmente, en la Figura 19 se muestra la nueva distribución de cirugías no realizadas, siguiendo presente las categorías 2 y 3. Resulta interesante observar que, si bien es mayor el promedio general mostrado por la categoría 2, este se debe en gran medida a una sola de las programaciones realizadas, siendo mucho menor (e incluso inexistente) en las demás repeticiones, mientras que la no realización se evidencia más como una constante en las salas de categoría 3

	NR POR CATEGORIA			
	1	2	3	4
1		4	2	
2		8	1	
3			1	
4				
5			2	
TOTAL	0	12	6	0
PROMEDIO	-	6	1,5	-

Figura 19. Cirugías no realizadas por categoría. Escenario 2

7.3. Análisis de variación de costos

Para la realización del análisis de costos, se partió de un escenario igual al de las primeras pruebas (4 salas, 50 cirugías y uso exclusivo de ventanas de tiempo estándar), esto para conseguir el costo base con el cual comparar las mejoras alcanzadas con la utilización de las demás opciones de programación. A partir de ahí, se realiza la reprogramación las cirugías habilitando una o varias de estas opciones y contrastando los resultados entre ellas. Las combinaciones realizadas fueron:

- Escenario inicial – uso exclusivo de ventanas de tiempo estándar.
- Habilitación de las ventanas de tiempo extra.
- Habilitación de los aplazamientos.
- Habilitación de la subutilización.
- Habilitación de las ventanas de tiempo extra y aplazamiento.
- Habilitación de las ventanas de tiempo extra y subutilización.
- Habilitación de los aplazamientos y subutilización.
- Habilitación de todos los recursos de programación.

	TOTAL REALIZADAS	% USO ESTANDAR	% NO REALIZADAS	COSTO
INICIAL	46	49,13%	8,00%	\$ 2.068.650
EXTRA	50	50,87%	0,00%	\$ 8.627
APL	50	52,95%	0,00%	\$ 15.255
SUB	48	50,52%	4,00%	\$ 1.057.500
EXTRA+ APL	50	52,26%	0,00%	\$ 4.890
EXTRA+ SUB	50	50,87%	0,00%	\$ 8.627
APL+ SUB	50	52,95%	0,00%	\$ 15.255
TODAS	50	52,26%	0,00%	\$ 4.890

Figura 20. Combinaciones de programación escenario 3

En la Figura 20 se muestra el resumen comparativo de una de las 5 programaciones realizadas, resaltando las combinaciones con un menor costo. Resulta interesante destacar 2 puntos, el primero es que la elección del menor costo no asegura la obtención del mejor uso de ventanas de tiempo, es este caso particular, aunque por un porcentaje muy pequeño, el mejor aprovechamiento del tiempo estándar lo tiene la tercera combinación (uso exclusivo del aplazamiento).

El segundo punto a destacar es el papel de la subutilización en las combinaciones. Si bien por si solo logra una mejora considerable del costo (cerca del 50%), resulta insuficiente en comparación con las demás alternativas, omitiéndose por completo en las combinaciones en las que hace parte (nótese que en “extra + subutilización” y “aplazamiento + subutilización” se mantienen los mismos valores que en las de “extra” y “aplazamiento” por separado). Si bien esto puede llevar a pensar en un primer momento que las combinaciones son innecesarias por reflejar únicamente la opción que brinda el menor costo de las 3, la verdad es que con el uso de este recurso es posible encontrar mejores soluciones. Como prueba de esto podemos observar los resultados del escenario “extra + aplazamiento” con el cual no solo se halla un punto intermedio de uso del tiempo estándar, sino que además se logra una reducción de costos de más del 40% con respecto al uso exclusivo de tiempo extra y de un valor cercano al 68% con respecto al aplazamiento.

Efectuando un análisis similar para las cinco repeticiones, se encuentran los valores promedio de optimización de costos mostrados en la Figura 21.

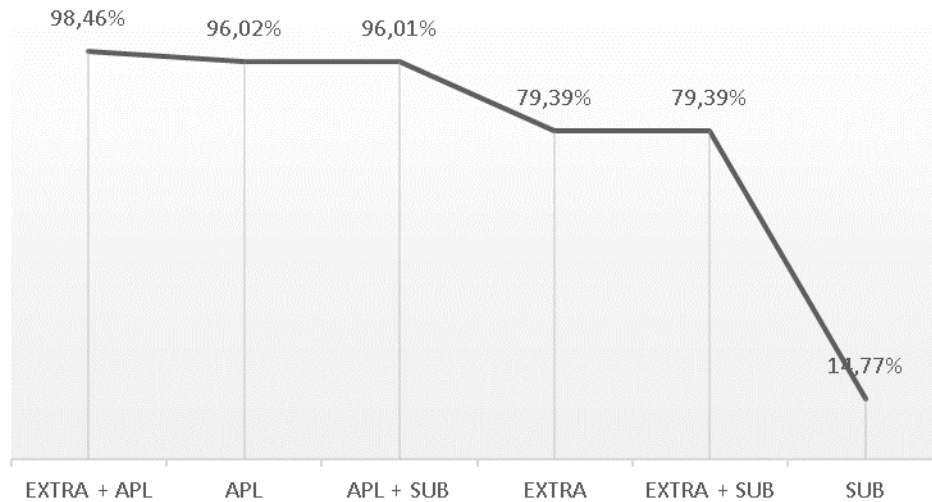


Figura 21. Comparativa de optimización de costos

Resulta evidente que la mayoría de las combinaciones suponen una mejora sobresaliente en comparación con el escenario inicial, siendo el aplazamiento y el uso de tiempo extra las opciones que más favorables. Sin embargo, es necesario tomar en consideración que estos resultados se ven estrechamente ligados con la capacidad de las salas, por lo que es posible que en algunas soluciones sobresalgan en mayor medida alternativas de programación diferentes según la disponibilidad de recursos.

Debido a que gran parte de los costos reducidos por el modelo se deben a la inclusión de las cirugías antes no realizadas, es posible concluir que la gráfica de comparativa en función de esta variable se comportará de manera muy similar que la de los costos, lo cual se puede corroborar fácilmente en la Figura 22, donde se ratifica la combinación de “extra + aplazamiento” como la mejor opción, con un porcentaje de optimización superior al 90%.



Figura 22. Comparativa de optimización de cirugías no realizadas

Finalmente, en lo relacionado al uso del tiempo total de las salas, se evidencia un aumento de más del 10% para las categorías 2 y 3 y ninguna mejora considerable para las salas de categoría 1 y 2, al igual que los porcentajes semanales de utilización que se mantuvieron en valores cercanos al 50% en todas las combinaciones de programación. El uso semanal del tiempo extra evidentemente presenta un aumento en las combinaciones que se encuentra habilitada esta opción, manteniendo un valor promedio de 3% para las soluciones de “extra + aplazamiento” las cuales demostraron tener el mejor desempeño en los experimentos realizados anteriormente. En la Figura 23 se muestran los promedios de utilización obtenidos, junto con los porcentajes de no realización de todas las combinaciones analizadas.

COMBINACIONES	Promedio Cirugías Realizadas	% Tiempo estandar semanal	% Tiempo extra semanal	Promedio de NR
INICIAL	46	47,33%	0,00%	8,00%
EXTRA	49	49,72%	6,94%	2,00%
APL	49,4	51,39%	0,00%	1,20%
SUB	46,8	47,92%	0,00%	6,40%
EXTRA + APL	49,8	52,02%	3,06%	0,40%
EXTRA + SUB	49	49,72%	6,94%	2,00%
SUB + APL	49,4	51,39%	0,00%	1,20%
TODOS	49,8	52,02%	3,06%	0,40%

Figura 23. Resultados tercer escenario

7.3.1. Análisis de escenarios finales

Para las últimas pruebas se busca enfrentar al algoritmo contra escenarios más similares a los que se podrían encontrar en la vida real, para esto, se plantea inicialmente la programación de 120 cirugías en 4 salas con el fin de evaluar el comportamiento del modelo en casos de saturación, así como para contar con un nuevo punto de partida con el cual comparar los resultados obtenidos con el aumento de recursos (salas), similar a lo planteado en los primeros escenarios pero empleando todos los medios de programación disponibles.

En los casos de saturación se alcanzó un promedio de programación superior al 85%. Resulta destacable comparar este valor con el 88% obtenido en los primeros escenarios analizados (4 salas, 50 cirugías) significando un aumento de más del 55% en el número total de procedimientos realizados con una disponibilidad de recursos similares. En la Figura 24 se muestran los promedios de utilización y no realización de las soluciones presentadas por el algoritmo.

	PROMEDIO
C. PROGRAMADAS	103
% USO ESTANDAR	91,84%
% NO REALIZACION	13,89%
% SUBUTILIZACION	11,22%
% APLAZAMIENTO	4,22%
% USO EXTRA	38,65%

Figura 24. Resumen cuarto escenario

Como se puede observar, el porcentaje de aplazamiento resulta bastante inferior a lo esperado con respecto a la subutilización y el uso de tiempo extra, esto claramente provocado por la poca disponibilidad de salas, en las cuales se deben programar primero las cirugías listadas para cada día, reemplazándolas por las aplazadas tan solo cuando estas últimas presentan una mayor prioridad. Debido a que la prioridad es el factor principal de organización de cirugías, resulta poco común que se encuentren aplazamientos con prioridades muy altas, lo que termina disminuyendo la utilización de este recurso en escenarios de saturación.

Si bien el porcentaje promedio de no realización también resulta similar al encontrado en las pruebas del primer escenario, la distribución de estas cirugías según su categoría si presenta una diferencia significativa, sobresaliendo la sala de categoría 3 como la mayor responsable de estos costos y ubicándola como principal cuello de botella en la programación final (Figura 25).

	1	2	3	4
% NR	2,00%	24,00%	64,00%	10,00%
PROMEDIO	0,3	4,0	10,7	1,7

Figura 25. Datos de No realización por categoría de sala

Con lo relacionado al uso de ventanas de tiempo extra, se puede observar en la Figura 26 la imposición de la sala de categoría 4 como mayor beneficiaria de este recurso, con una utilización

de más del 70% del total semanal disponible, seguida de la sala de categoría 3 con un uso de tan solo el 38.9%, lo cual resulta interesante tomando en consideración el gran porcentaje de cirugías no realizadas de esta sala, situación que podría explicarse por la duración de los procedimientos y los porcentajes de aparición definidos en los parámetros iniciales del modelo. Si bien en la sala de categoría 4 se asignaron 6 diferentes procedimientos (el doble que en las demás salas), su promedio de duración es el más bajo (la mitad en comparación con las otras) lo que le permite la asignación diaria de una mayor cantidad de cirugías. Por otro lado, las demás salas cuentan con procedimientos de una larga duración (en promedio mayor que el total de ventanas extra diarias disponibles) por lo que resulta imposible la asignación de más de 1 cirugía en este espacio, y por lo tanto un alto desaprovechamiento del recurso.

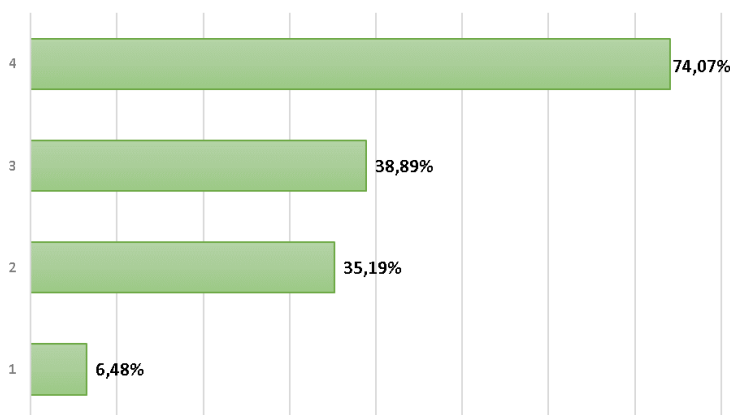


Figura 26. Uso de tiempo extra, escenario 4

Finalmente, con la habilitación de las 8 salas se obtuvo una reducción considerable en la cantidad de cirugías no realizadas (95% para las pertenecientes a la sala de categoría 3 y 67% para las de categoría 2) manteniendo un promedio final de no realización de poco más de 1% (2 cirugías por ensayo) lo cual se vio reflejado en una disminución similar en los costos totales con un porcentaje superior al 80%.

Con el aumento de disponibilidad de las salas, también se vio reducido su porcentaje total de utilización, siendo la más notoria la sala de categoría 1, contemplando de tal manera su aumento en la capacidad como una posible mejora innecesaria teniendo en cuenta el bajo índice de no realización y de uso de tiempo extra en los casos de saturación. En la Figura 27 se puede apreciar una comparativa de los promedios de cada escenario.

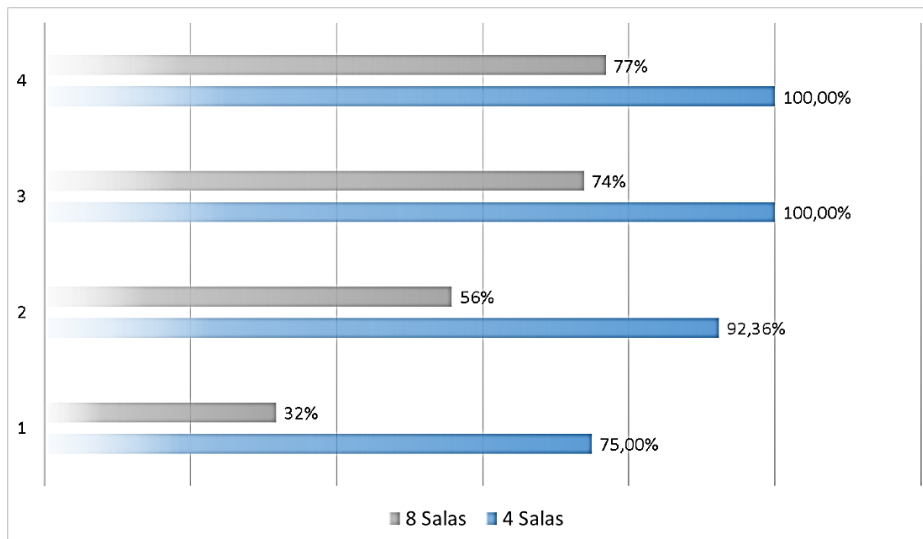


Figura 27. Comparación uso de tiempo estándar escenarios finales

También resulta interesante que el uso promedio de las salas de categoría 3 y 4 no presenten una disminución tan marcada como las demás, lo que se debe a la programación de cirugías que antes se encontraban en subutilización (la cual presenta una reducción del 98% con respecto al escenario de saturación) y en ventanas de tiempo extra. Cuestión que se puede apreciar de forma más clara en la Figura 28.

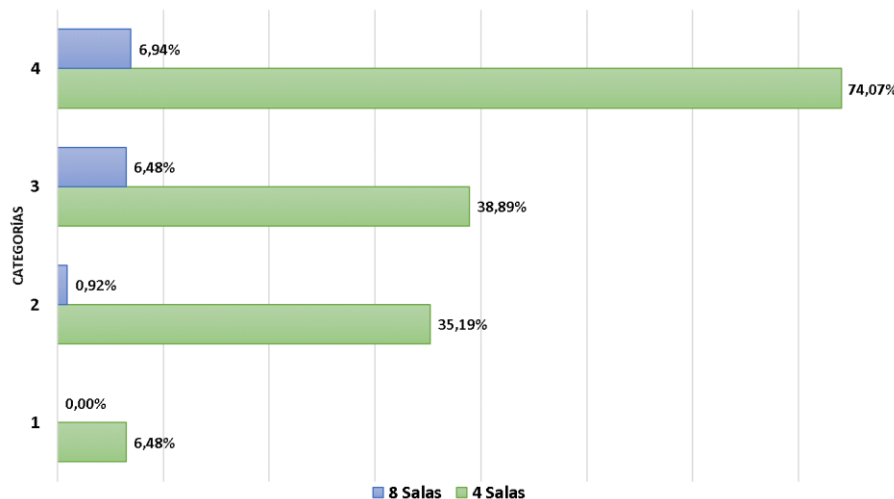


Figura 28. Comparativa del uso de tiempo extra en escenarios finales

Tomando en cuenta que para estos experimentos se dispuso de un total de 42 doctores, se encuentra que se tuvo una utilización promedio de tan solo el 62%, lo que equivalen a un total de 26 cirujanos, los cuales se distribuyen de la manera mostrada en la Figura 29 según el capítulo de cirugía que representan.

CATEGORIA	PROMEDIO	CATEGORIA	PROMEDIO
1	1	8	1
2	2	9	3
3	2	10	2
4	2	11	2
5	2	12	2
6	2	13	2
7	2	14	1

Figura 29. Cantidad promedio de cirujanos por capítulo de cirugía

Hay que recordar igualmente, que para estos experimentos se determinó que los cirujanos realizarían jornadas completas en esta labor, sin tener en consideración tareas adicionales como la consultoría de pacientes, por lo que estos promedios se deben contemplar tan solo como el número

mínimo de cirujanos necesarios para la realización de las cirugías semanales y no como la cantidad adecuada de doctores para el correcto funcionamiento del hospital.

8. Conclusiones

El algoritmo implementado en la solución del problema de programación se ve altamente beneficiado por el uso de instancias de gran tamaño, no porque sea incapaz de encontrar soluciones factibles en escenarios donde estos atributos tienen un tamaño menor, sino porque la confiabilidad en los porcentajes de optimización resultantes mejora conforme se le permite al modelo evaluar una mayor y más variada combinación de escenarios. Sin embargo, el esfuerzo computacional aumenta considerablemente conforme al incremento de las variables, incluyendo la población de hormigas, iteraciones a realizar, cantidad de salas, doctores y enfermeras, por lo que los tiempos totales de ejecución pueden variar desde algunos minutos hasta varias horas según la configuración elegida. Para los experimentos finales realizados en este proyecto el tiempo de ejecución fue cercano a una hora y media.

Debido a la aleatorización de algunas variables, la heurística de construcción permitió que el algoritmo se enfrentara a una gran cantidad de escenarios diversos, acercándose a la imprevisibilidad de los casos de la vida real y facilitando de esta manera analizar el funcionamiento del modelo en situaciones más cercanas a las que se encontrarían en la práctica.

En lo concerniente al uso de las salas, se tiene que si bien en la literatura se encuentran un número variable (de entre 3 hasta 12), en este proyecto se consideró apropiado analizar dos configuraciones específicas, una con 4 salas (1 por categoría) y otra con 8 salas (2 por categoría),

la primera sirviendo como un punto de comparación con respecto a las mejoras encontradas por el modelo y para la evaluación de los casos de saturación, y la segunda sirviendo para el análisis de los escenarios con una mayor flexibilización de recursos.

Aunque el porcentaje promedio de utilización de las salas no superó el 50%, durante los primeros experimentos realizados se evidenció la necesidad del uso de las herramientas de programación planteadas en el proyecto (subutilización, aplazamiento y tiempo extra) debido al alto índice de cirugías no realizadas en cada uno de los casos donde estas no se tenían en cuenta (superior al 10% y provenientes en mayor medida de la sala de categoría 3), llegando a encontrar una mejora de más del 90% con la combinación del uso de tiempo extra y el aplazamiento de los procedimientos. Por otro lado, las combinaciones relacionadas con la subutilización de las salas, aunque presentan unos porcentajes de optimización considerables en todos los aspectos analizados, se muestran con un desempeño inferior con respecto al encontrado mediante el uso en conjunto de las dos herramientas anteriormente mencionadas, llegando incluso estas a ser las únicas tenidas en cuenta e ignorando los posibles beneficios de la subutilización, esto muy posiblemente debido a la restricción de que la herramienta sea de uso exclusivo para las cirugías pertenecientes a la sala de categoría 4.

Si bien, como se mencionó anteriormente, en un principio se muestra una preferencia por el uso de la combinación de aplazamiento y tiempo extra, el análisis de los casos de saturación manifestó que dependiendo de la disponibilidad de recursos físicos (salas) es imposible asegurar la utilización del aplazamiento, obligando de tal forma al modelo a utilizar combinaciones menos óptimas, situación que puede solucionarse con el aumento de las salas de cirugía disponibles. No obstante, aunque en el experimento realizado se habilitaron 4 salas adicionales, los porcentajes de ocupación y no realización evidenciaron que esto tan solo es necesario en salas de categoría 3 y 4 debido a la

duración y probabilidad de aparición de las cirugías relacionadas con estas salas, y en una menor medida en las salas de categoría 1 y 2, especialmente en la primera, donde incluso en los escenarios de saturación no presentaba mayor dificultad para la programación de los procedimientos.

En cuanto al uso del tiempo extra, se encontró que las salas con mayor beneficio de esta herramienta son las de categoría 4, esto debido a la corta duración de sus cirugías, lo que le facilita al modelo su programación, y por lo tanto un uso más eficiente de las ventanas de tiempo. Este factor es también por la cual en las demás salas el comportamiento es diferente, si bien, la habilitación del tiempo extra ayuda a la programación de varias cirugías en la semana (que en otro caso no podrían realizarse), las altas duraciones de los procedimientos relacionados con las demás categorías impiden programar más de una en este espacio, resultando en el desaprovechamiento de un gran porcentaje de ventanas de tiempo disponibles. Sin embargo, al igual que con los aplazamientos, este problema se soluciona en gran medida con la apertura de nuevas salas, llegando a reducir el uso de estos tiempos en más del 90% para todas las salas a excepción de las de categoría 3 cuyo porcentaje de mejora se mantiene con valor un poco por debajo del 85%.

Con respecto a la optimización de los costos, resulta evidente la mejora obtenida con la utilización de las herramientas de programación, llegando una reducción de casi el 100% con respecto a los escenarios iniciales, y si bien en un primer momento esto puede parecer poco realista, es necesario tener en cuenta que el modelo planteado no toma en consideración costos fijos de la operación de un hospital, sino tan solo los resultantes por las penalizaciones definidas en un principio (no realización, aplazamientos, uso de tiempo extra y subutilización). Debido a que gran parte de estos costos se deben a los procedimientos no realizados, una reducción en este factor provocará una mejora considerable en el resultado final de penalización.

En lo relacionado al uso de los doctores se encontró una distribución similar en cuanto a la cantidad requerida según su especialidad (la cual está relacionada con los 14 capítulos de las cirugías basadas en la clasificación única de procedimientos – CUPS) con un promedio de 2 cirujanos por capítulo, a excepción de los encargados de las cirugías relacionadas con las salas de categoría 1, con los cuales resultó necesario tan solo el uso de 1 cirujano. Aunque cabe resaltar que en todos los experimentos realizados se contó con al menos 1 doctor con una experiencia mayor a 10 años, lo que impidió que no se realizaran procedimientos por falta de personal capacitado.

Finalmente, el artículo de carácter publicable con los resultados obtenidos en esta investigación se adjunta en el apéndice G, así como una versión traducida al inglés (Apéndice H).

9. Recomendaciones

Uno de los principales inconvenientes encontrados para la realización de este proyecto fue la dificultad de la programación de la heurística, razón principal por la que en un principio se eligió trabajar con el lenguaje de programación “Python”, el cual resulta mucho más sencillo que las demás alternativas consideradas, sin embargo, la complejidad del problema y la poca aparición de investigaciones que se enfocaran al mismo tiempo en la planificación del uso de salas y del equipo de cirugía, conllevó a no tener en consideración factores que habrían resultado interesantes de analizar en una programación semanal de este tipo, entre las cuales se resaltan y se contemplan como mejoras a realizar en futuras investigaciones: la adición de nuevos costos de penalización, (como los resultantes de la habilitación de nuevas salas e ineficiencia en el uso del personal

disponible) así con el uso de nuevas opciones de programación, como el adelanto de cirugías o la aceptación de la subutilización en las cirugías de las salas de categorías 2 y 3.

Debido a que el algoritmo tiene una gran cantidad de aleatorización de factores, en caso de quererse utilizar en escenarios reales, se recomendaría realizar un cambio en el sistema de recepción de parámetros, permitiendo al modelo leer desde una fuente externa los datos relacionados a las cirugías a realizar, así como la información de los recursos disponibles en el hospital (cantidad y características de las salas y del personal). Adicional a esto, se consideraría conveniente adicionar una interfaz de usuario que facilitara el filtrado de información y la interpretación del resultado entregado por el programa.

Referencias Bibliográficas

- Akbarzadeh, Moslehi, Reisi-Nafchi, & Maenhout. (2019). The re-planning and scheduling of surgical cases in the operating room department after block release time with resource rescheduling. *European Journal of Operational Research*, 596-614. recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0377221719303741>
- Bautista, D. (2015). *ESTADO DEL ARTE DE LOS MODELOS DE OPTIMIZACIÓN EN LA LOGÍSTICA DE ATENCION HOSPITALARIA (Tesis de pregrado)*. Recuperado de <http://tangara.uis.edu.co/biblioweb/tesis/2015/156260.pdf>
- Bejarano, M. (2011). Evaluación cuantitativa de la eficiencia en las salas de cirugía. *Revista Colombiana de Cirugía*, 273-84. Recuperado de <https://www.revistacirugia.org/index.php/cirugia/article/view/189>
- Belien, & Demeulemeester. (2008). A branch-and-price approach for integrating nurse. *European Journal of Operational Research*, 652-66. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0377221706011751>
- Belkhamza, Jarboui, & Masmoudi. (2018). Two metaheuristics for solving no-wait operating room surgery scheduling. *Computers & Industrial Engineering*, 494-506. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S036083521830487X>
- Burke, Li, & Qu. (2010). A hybrid model of integer programming and variable neighbourhood search for highly-constrained nurse rostering problems. *European Journal of Operational*

- Research*, 484-493. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0377221709005396>
- Casanovas, N., Puig, E., Cruz, A., Martos, A., Mate, X. y García, JL. (2013). La gestión quirúrgica según niveles de complejidad: un modelo que se espera eficiente. *Gestión y evaluación de costes sanitarios*, 14(4), 675-83. Recuperado de https://www.fundacionsigno.com/archivos/publicaciones/05_GESTION_QUIRURGICA.pdf
- Costa A, Jr (2017). Assessment of operative times of multiple surgical specialties in a public university hospital. *Einstein (Sao Paulo, Brazil)*, 15(2), 200–205. Recuperado de <https://doi.org/10.1590/S1679-45082017GS3902>
- Cunquero, R. M. (s.f.). *Algoritmos Heurísticos en Optimización Combinatoria*. Recuperado de https://www.academia.edu/35236321/Algoritmos_Heur%C3%ADsticos_en_Optimizaci%C3%B3n_Combinatoria
- Díaz, Huecas, Allende, S., & Manso, G. (2007). Metaheurísticas de optimización combinatoria: Uso de simulated annealing para un problema de calendarización. *Tecnologi@ y Desarrollo*, 1-25. Recuperado de https://revistas.uax.es/index.php/tec_des/article/view/544/500
- Duenas, Martinelly, Yazgi, & Aguado. (2017). A Multi-objective Hospital Operating Room Planning and Scheduling Problem Using Compromise Programming. *Advances in Computational Intelligence*, 379-390. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0305054816302684>

- Guido, & Conforti. (2017). A hybrid genetic approach for solving an integrated multi-objective operating room planning and scheduling problem. *Computers and Operations Research*, 270-282. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0305054816302684>
- Hillier, & Lieberman. (2015). *INTRODUCTION TO OPERATION RESEARCH (EDITION 10)*. New York, Estados Unidos: McGraw-Hill.
- Lim, Mobasher, Bard, & Najjarbashi. (2016). Nurse scheduling with lunch break assignments in operating suites. *Operations Research for Health Care*, 35-48. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S2211692316300091>
- Martinelly, & Meskens. (2017). A bi-objective integrated approach to building surgical teams and nurse. *International Journal of Production Economics*, 323-334. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S0925527317301597>
- Meneses, C. (2017). *Estudio del Problema de Programación de Turnos de Enfermería*. (Tesis de Pregrado). Recuperado de <http://tangara.uis.edu.co/biblioweb/tesis/2017/168600.pdf>
- Muñoz, M. (2018). *Desarrollo de un Estudio para el Problema de Programación de Salas de Cirugía (Tesis de Pregrado)*. Recuperado de <http://tangara.uis.edu.co/biblioweb/tesis/2018/172315.pdf>
- Portafolio, R. (21 de Enero de 2018). El país tendría mínimo 10 años más con déficit en salud: Acemi. Recuperado de <https://www.portafolio.co/economia/el-pais-tendria-minimo-10-anos-mas-con-deficit-en-salud-acemi-513467>

Rodriguez, Peña, D. I., & Rincón. (2014). Análisis de los métodos heurísticos de resolución del problema de programación de proyectos con recursos limitados (RCPSP: Resource Constrained Project Scheduling Problem). *Tecnología y Desarrollo*, 1-21. Recuperado de https://revistas.uax.es/index.php/tec_des/article/view/591/547

Suarez, O. (2011). Una Aproximación a la Heurística y Metaheurísticas. *Inge@UAN*, 44-51. Recuperado de <http://csifsvr.uan.edu.co/index.php/ingean/article/view/198/170>

Taha, H. A. (2012). *Investigación de Operaciones*. Mexico: Pearson.

Xiang, Ma, & Wang. (2015). Operating Rooms Decision Optimization Integrating Surgery Planning and Nurse Rostering. *Industrial Engineering, Management Science and Applications*, 919-926. Recuperado de https://bibliotecavirtual.uis.edu.co:2142/chapter/10.1007/978-3-662-47200-2_96

Xiang, Yin, & Lim. (2015). A short-term operating room surgery scheduling problem integrating multiple nurses roster constraints. *Artificial Intelligence in Medicine*, 91-106. Recuperado de <https://bibliotecavirtual.uis.edu.co:2191/science/article/pii/S09333365714001420>

Colombia, Resolución Número 0005851 de 2018 (21 de diciembre de 2018)