

**UNIVERSIDAD DE CONCEPCIÓN - CHILE
FACULTAD DE INGENIERIA
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL**

***Aplicación del Problema de Asignación en Mantenimiento de
Equipos Médicos.***

por

Daniela Catalina del Pilar Hermosilla Aedo

Profesor Guía:

Dott.ssa Rosa Daniela Medina Durán

Concepción, Diciembre de 2017

Tesis presentada a la

**DIRECCIÓN DE POSTGRADO
UNIVERSIDAD DE CONCEPCIÓN**



Para optar al grado de

MAGISTER EN INGENIERÍA INDUSTRIAL

Financiado por proyecto FONDECYT de iniciación 11140244

RESUMEN

Aplicación del Problema de Asignación en Mantenimiento de Equipos Médicos.

Daniela Catalina del Pilar Hermosilla Aedo

Diciembre 2017

PROFESOR GUIA: Dott.ssa Rosa Medina Durán

PROGRAMA: Magíster en Ingeniería Industrial

El equipamiento médico es la representación de lo que la tecnología puede aportar en la recuperación de la salud y la mejora de la calidad de vida de las personas, apoyando ya sea con el diagnóstico, monitoreo o soporte de los pacientes. Es por esta razón que mantener este equipamiento en su mejor rendimiento evita que se generen errores tan graves como administración de dosis incorrecta o registro de parámetros biológicos erróneos. Este trabajo se enfoca en la asignación de monitores multiparámetros que requieren mantenimiento correctivo, a empresas que ofrezcan el servicio técnico.

El presente estudio contempla la posibilidad de que existan características no cuantificables y difíciles de incluir en el modelo. Por esta razón la metodología empleada entrega una lista ordenada de k soluciones, otorgando mayor libertad al tomador de decisiones.

Además, se considera la capacidad de cada empresa, es decir, el número de técnicos disponibles que pueden reparar la falla. Las empresas son representadas por grupos de elementos o *clusters*, y la asignación a elementos, del mismo *cluster* tiene el mismo costo asociado.

La lista no decreciente de costos, puede ser extensa, dependiendo del número de *clusters* y de elementos contenidos en éstos, por lo que el algoritmo propuesto extrae las soluciones que son diferentes, o bien, asignaciones con una variación a nivel de *clusters*.

Los resultados indican que existe una reducción en el tiempo de ejecución a un 0,0013%, y la reducción en dinero es CLP 46 316 806, lo que corresponde a un 54,20%. Basados en los resultados podemos decir que el algoritmo propuesto otorga un mejor uso de los recursos y en menor tiempo.

Palabras Claves: Semi-asignación, secuenciamiento de soluciones, mantenimiento de equipos médicos.

ABSTRACT

Application of assignment problem in maintenance of medical equipment.

Daniela Catalina del Pilar Hermosilla Aedo
December 2017

THESIS SUPERVISOR: Dott.ssa Rosa Medina Durán

PROGRAM: Master in Industrial Engineering

The medical equipment is the representation of what technology can contribute in the recovery of health and the improvement of the quality of life of people, helping with either the diagnosis, monitoring or support of patients. It is for this reason that maintaining this equipment at its best performance avoids generating such serious errors as incorrect dose infusion or recording of erroneous biological parameters. This work focuses on the assignment of multiparameter monitors that require corrective maintenance, to companies that offer technical service.

The present study contemplates the possibility that there are non-quantifiable characteristics that are difficult to include in the model. For this reason the methodology used provides an ordered list of solutions, giving more freedom to the decision maker.

In addition, the capacity of each company is considered, that is, the number of technicians available who can repair the fault. Companies are represented by groups of elements or clusters, and the assignment to elements of the same cluster has the same associated cost.

The non-decreasing list of costs can be extensive, depending on the number of clusters and elements contained in them, so the proposed algorithm extracts solutions that are different, or assignments with a variation at the level of clusters.

Results indicate that there is a reduction in the execution time to 0.0013%, and the reduction in money is CLP 46,316,806, which corresponds to 54.20%. Based on the results we can say that the proposed algorithm gives better use of resources and in less time.

Keywords: Semi-assignment, ranking of solutions, maintenance of medical equipment.

Tabla de Contenidos

RESUMEN	II
ABSTRACT	III
Tabla de Contenidos	IV
Lista de Figuras	VI
Lista de Tablas	VIII
Abreviaciones	IX
CAPÍTULO 1: Revisión Bibliográfica	1
<i>1.1 Introducción</i>	<i>1</i>
<i>1.2 Mantenimiento de equipos médicos</i>	<i>2</i>
<i>1.3 Modelo de asignación</i>	<i>4</i>
<i>1.4 Algoritmos para resolver el problema de asignación</i>	<i>6</i>
<i>1.4.1 Algoritmos Primitives-Duales</i>	<i>6</i>
<i>1.4.2 Algoritmo Dinic–Kronrod</i>	<i>8</i>
<i>1.4.3 Implementación “Shortest path” del algoritmo Húngaro</i>	<i>9</i>
<i>1.4.4 Algoritmos primales</i>	<i>9</i>
<i>1.4.5 Algoritmos duales</i>	<i>9</i>
<i>1.4.6 Otros trabajos</i>	<i>10</i>
<i>1.5 Algoritmos de secuenciamiento</i>	<i>10</i>
<i>1.5.1 Algoritmo de Murty</i>	<i>11</i>
<i>1.5.2 Algoritmo de Chegireddy-Hamacher</i>	<i>12</i>
<i>1.5.3 Algoritmo de Pedersen et al.</i>	<i>13</i>
<i>1.6 Problema de semi-asignación</i>	<i>13</i>
<i>1.7 Simetría y planos de corte</i>	<i>15</i>

CAPÍTULO 2: Propuesta	17
2.1 <i>Hipótesis</i>	17
2.2 <i>Objetivos</i>	17
2.2.1 <i>Objetivo general.....</i>	17
2.2.2 <i>Objetivos específicos.....</i>	17
2.3 <i>Alcances y limitaciones.....</i>	17
2.4 <i>Metodología.....</i>	18
CAPÍTULO 3: Desarrollo metodológico	19
3.1 <i>Algoritmo de asignación.....</i>	19
3.2 <i>Algoritmo de Murty.....</i>	21
3.3 <i>Cortes de simetría.....</i>	21
CAPÍTULO 4: Caso de estudio	23
4.1 <i>Descripción del caso.....</i>	23
4.2 <i>Monitores Multiparámetro</i>	23
4.3 <i>Supuestos</i>	24
4.4 <i>Instancias.....</i>	25
CAPÍTULO 5: Resultados	26
5.1 <i>Asignación actual</i>	26
5.2 <i>Asignación utilizando el secuenciamiento de soluciones</i>	26
5.3 <i>Discusión</i>	26
CAPÍTULO 6: Conclusiones	28
6.1 <i>Trabajo Futuro</i>	28
BIBLIOGRAFÍA.....	29
ANEXO	34

Lista de Figuras

Figura 1. Posibilidades de asignación en grafo bipartito $G(V1, V2, E)$. Fuente: Elaboración propia. .	5
Figura 2. Polígono de asignación. Los puntos representan todas las soluciones. El polígono delimita la región de soluciones factibles, y entre sus vértices se encuentra el óptimo. Los vértices del polígono de asignación corresponden sólo a matrices de permutación. Fuente: Elaboración propia.	5
Figura 3. Ejemplo de cálculo del elemento mínimo de Δ en cada fila (Dinitz, 2012).....	8
Figura 4. Ejemplo de selección de soluciones a partir de los nodos encontrados en el algoritmo de Murty (Burkard et al., 2009).	12
Figura 5. Pseudocódigo del algoritmo de reoptimización de Pedersen et al. (Pedersen et al., 2008).	13
Figura 6. Ejemplo de igualdad de costos dentro de los <i>clusters</i> en un problema de asignación. c_{ia} corresponde al costo de asignar el elemento i a algún elemento en el <i>cluster</i> a y c_{ib} corresponde al costo de asignar el elemento i a un elemento fuera del <i>cluster</i> . Lo mismo sucede con los otros elementos a asignar (diferentes a i), pero los costos asociados no fueron incluidos en la imagen para mejor visualización. Fuente: Elaboración propia.....	13
Figura 7. Representación del conjunto de simetrías de un triángulo equilátero («Triangle Fractals», 2005).....	15
Figura 8. Matriz de costos de problema usado para ejemplificar procedimiento del algoritmo JV. Fuente: Elaboración propia.	19
Figura 9. Asignación final luego de aplicar algoritmo JV. Fuente: Elaboración propia.	21
Figura 10. Ejemplo de monitor multiparámetro (“Monitor multiparámetro VISMO PVM-2701 - arquimed,” n.d.).....	24
Figura 11. Porcentaje de variación de costo para cada uno de los grupos (empresas). Fuente: Elaboración propia.	27
Figura 12. Costo por empresa en asignación utilizando el método actualmente empleado. Fuente: Elaboración propia.	43
Figura 13. Costo por empresa en la mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.	43
Figura 14. Costo por empresa en la 2ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.	44
Figura 15. Costo por empresa en la 3ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.	44

Figura 16. Costo por empresa en la 4ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 45

Figura 17. Costo por empresa en la 5ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 45

Figura 18. Costo por empresa en la 6ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 46

Figura 19. Costo por empresa en la 7ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 46

Figura 20. Costo por empresa en la 8ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 47

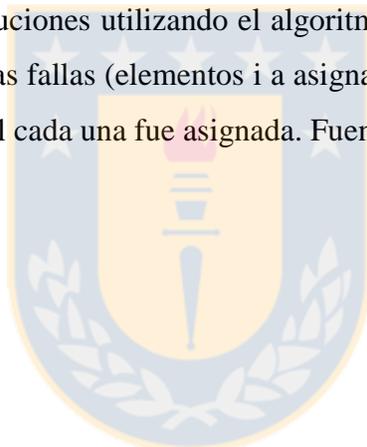
Figura 21. Costo por empresa en la 9ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 47

Figura 22. Costo por empresa en la 10ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia. 48



Lista de Tablas

Tabla 1. Variables duales luego del primer paso de la inicialización (1.a) del algoritmo JV. Fuente: Elaboración propia.	20
Tabla 2. Variables duales luego del tercer paso de la inicialización (1.c) del algoritmo JV. Fuente: Elaboración propia.	20
Tabla 3. Variables duales luego del segundo paso del algoritmo JV. Fuente: Elaboración propia. ...	20
Tabla 4. Variables duales luego del tercer paso del algoritmo JV. Fuente: Elaboración propia.....	21
Tabla 5. Primeros 10 costos ordenados de forma creciente. Fuente: Elaboración propia.....	26
Tabla 6. Asignación realizada con el método actualmente utilizado en el hospital. La columna F representa las fallas que debieron ser asignadas, numeradas en orden correlativo, y la columna G representa las empresas (grupos) a los cuales fueron asignadas. Fuente: Elaboración propia.....	34
Tabla 7. Primeras 10 mejores soluciones utilizando el algoritmo presentado. Los valores mostrados en la columna F corresponden a las fallas (elementos i a asignar). Para cada solución (S 1 a S 10) se muestra la empresa (grupo) al cual cada una fue asignada. Fuente: Elaboración propia.....	35



Abreviaciones

LSAP: Linear sum assignment problem.

k: Número de soluciones que se quiere encontrar con el algoritmo de secuenciamiento.

n: Cantidad de elementos a asignar.

JV: Jonker and Volgenant.

SAP: Shortest augmenting path.



CAPÍTULO 1: Revisión Bibliográfica

1.1 Introducción

El equipamiento médico es indispensable en todo centro hospitalario, ayuda en el diagnóstico, tratamiento y a dar soporte a pacientes de todo tipo. Existen algunas unidades que no pueden atender a los pacientes si no se encuentra algún equipo crítico como: monitor multiparámetro, bomba de infusión o ventilador mecánico. Por esta razón, el mantenimiento correctivo es una tarea primordial, para mantener la disponibilidad de éstos, y asegurar que funcione correctamente.

Cuando un equipo falla, debe ser asignado a un proveedor de servicio técnico para que éste sea reparado en el menor tiempo posible. Cuando el número de equipos es pequeño, la asignación se puede realizar manualmente, pero si aumenta, se convierte en un problema altamente combinatorio, por lo tanto, para el cálculo de la solución óptima se requiere una mayor cantidad de tiempo.

El caso de estudio está basado en un hospital de alta y mediana complejidad en Chile, uno de los mayores y más complejos del país. Por lo tanto, se requiere de un algoritmo para encontrar la solución que optimice el uso de los recursos.

Se debe considerar que cada empresa tiene más de un técnico disponible, donde cada uno puede reparar una falla. En consecuencia, este caso es tratado como un problema de semi-asignación con *clusters* (o grupos), donde el número de técnicos disponibles corresponde a la capacidad de la empresa, y por lo tanto, a la cantidad de fallas que puede reparar.

En la literatura, se encuentran trabajos que resuelven eficientemente el problema de asignación (Burkard & Çela, 1999), pero puede que éstos no sean suficientes para cubrir la totalidad del problema. Existen características no cuantificables, difíciles de incluir en el modelo, por lo que se requiere no solo la solución óptima, sino también las k mejores soluciones, solicitando alternativas de asignación cuando la primera no cumpla con todos los aspectos del contexto en que se aplica el modelo. En el caso de estudio se hace necesario incluir esta condición debido a que existe una variabilidad entre cada una de las instancias a resolver, y es la persona a cargo quien tiene acabado conocimiento de asignaciones previas, experiencias con las empresas, desempeño del equipamiento, interacción de los usuarios, etc. Debido a la relevancia del mantenimiento de equipos médicos, es que se debe dar mayor libertad al tomador de decisiones, otorgando diferentes asignaciones, y la diferencia de costo que implica realizar esa modificación.

A continuación se presenta bibliografía relacionada con el problema de asignación, mantenimiento de equipos médicos, semi-asignación, asignación con *clusters* y cortes de simetría, que

son utilizadas en el desarrollo de este trabajo.

1.2 Mantenimiento de equipos médicos

En el caso de estudio, se tiene una cantidad de equipamiento médico, los cuales han presentado algún defecto. Una vez detectada la falla, se solicita la reparación a la unidad encargada, y se debe decidir qué empresa será la encargada de realizar el servicio técnico.

El mantenimiento corresponde a las acciones necesarias para que el equipo continúe en funcionamiento, en las condiciones requeridas. Dentro de los objetivos del mantenimiento se pueden mencionar: cumplir con las normas; proteger a la persona que tenga contacto con el equipo; evitar mediciones incorrectas; disposición del equipo a ser utilizado; mejorar su desempeño; evitar que el equipo se dañe irreparablemente por forzarse a funcionar con un componente en mal estado; entre otros (Unidad de Servicios Generales y Mantenimiento, 2007).

Existen dos tipos de mantenimiento (World Health Organization, 2012), correctivo y preventivo. El mantenimiento correctivo (MC) es el proceso para restaurar la integridad, la seguridad o el funcionamiento de un dispositivo después de que se presenta una falla. En cambio, el mantenimiento preventivo (MP) se realiza para evitar una posible falla en el dispositivo y alargar su vida útil mediante revisiones frecuentes de su desempeño o estado. En este último se incluyen generalmente tareas como calibración, lubricación, limpieza o reemplazo de piezas que tienen mayor tasa de falla, o que tienen un mayor uso, por lo que se desgastan más rápido.

Dentro de la literatura disponible existen una serie de estudios efectuados en distintas áreas, desarrollados principalmente en el área de mantenimientos preventivos.

Respecto de las aplicaciones de programación de mantenimientos en el área de equipamiento médico, existe una serie de estudios, como el realizado en (Joseph & Madhukumar, 2010), donde se presenta el desarrollo de un índice de mantenimiento preventivo para cada instrumento en el inventario clínico, y disponen de una gestión de riesgo en equipos médicos, de manera de asignar un intervalo de mantenimiento preventivo, en los que las pruebas de seguridad deben ser aplicadas, como también, priorizar el procedimiento de mantenimiento preventivo. En este estudio se presenta un índice numérico, basándose en el coeficiente de nivel de riesgo de un instrumento (RLC por su sigla en inglés). De esta forma, el estudio presentó una calendarización de las mantenciones preventivas de forma adaptativa y comandada por los datos históricos.

En (Robson, Yeo, Riches, Carlisle, & Kitto, 2005) se presenta una metodología para clasificar los equipos médicos, en base a una serie de factores, como son: la frecuencia de uso, entorno en que se usa, naturaleza de su función, experiencia y conocimientos de los usuarios, confiabilidad de los

equipos, costo de los mantenimientos de soporte, gestiones previas de programación y reprogramación de mantenciones, disponibilidad de repuestos, riesgos asociados con la discontinuidad en su uso, entre otros. Esos factores, son esencialmente indicadores de la probabilidad que ocurra un problema en un equipo o permiten medir la consecuencia que un problema exista. De esta forma se puede determinar un riesgo relativo de cada dispositivo médico. Como no es posible disponer de toda esta información, los autores postulan que existen algunas características que tienen un gran nivel de relevancia, y que al ser medidos de forma aislada, siguen siendo indicadores útiles del riesgo relativo. Estos indicadores, son la condición del dispositivo médico, la utilización del dispositivo, y el cumplimiento del dispositivo médico, en relación a cumplir con estándares esenciales de eficiencia, eficacia y seguridad. Además, se muestra que las principales consecuencias relacionadas con problemas en los equipos médicos son los relacionados con las actividades desarrolladas con el equipo, la seguridad que presentan al ser usados, y el impacto financiero que tienen, por los costos de operación y de mantenimiento asociados.

El estudio desarrollado en (Manzini, Accorsi, Cennerazzo, Ferrari, & Maranesi, 2015) presenta un modelo de programación lineal entera mixta, para programar el mantenimiento preventivo basado en costos, confiabilidad y en la disponibilidad de recursos. El modelo presentado minimiza la función de costos totales obtenida desde una serie de elementos, como son el costo de la ejecución de las acciones preventivas y el costo de reparaciones adicionales en caso de fallas que no han sido previstas. Se incluye también el costo del personal de la empresa proveedora del equipo y/o de la empresa que provee servicios de mantenimiento. En particular, se requiere una serie de tareas, y para cada tarea, existe una frecuencia nominal de ejecución de acuerdo a los tiempos medios entre fallas de las partes y componentes involucrados. El objetivo es definir la programación que minimice el costo global de mantenimiento preventivo y planificarlos, en concordancia con las restricciones de capacidad. El modelo propuesto, puede ser aplicado en acciones preventivas de corto plazo, como son semanalmente, y el uso de recursos, repuestos, accesorios, equipos, y personal necesario para realizar las acciones de mantenimiento. Es principalmente un enfoque ex-ante, es decir, mantenimiento previo a que suceda la falla.

En el estudio desarrollado en (Bassani, Rocha, Luders, & Bizinotto, 2002), se describe la aplicabilidad de un modelo de gestión de control de procesos de mantenimiento de equipamiento médico en el Centro de Ingeniería Biomédica de la Universidade Estadual de Campinas (UNICAMP). El modelo se basa en diagramas de estado de todas las actividades de mantenimiento. El diagrama de estados involucra la ejecución de actividades en cinco sectores: gestión de ingeniería clínica, centro de servicio al cliente, proveedor de suministros, grupo ingenieril, y personal técnico. El primer paso

para analizar los datos del modelo es a través de un análisis en el dominio de los micro-procesos. Usado para estudiar el rendimiento de un grupo en particular, o de otros aspectos, como son la clase de equipamiento, o un departamento del Hospital en particular. Uno de los objetivos de este análisis es proveer a los ingenieros clínicos de un método simple para cambiar las especulaciones por datos reales, y transformar los datos en figuras simples, fáciles de obtener y de interpretar.

En estos trabajos se menciona la importancia de la planificación del mantenimiento correctivo y se demuestra la relevancia de definir qué factores influyen en el proceso y la variabilidad de éstos, decidir qué algoritmos y/o funciones son la mejor opción para nuestra instancia y considerar las posibles variantes. Al desarrollar un sistema que se adecúe a lo anterior, se tendrá una solución adaptable y capaz de gestionar de manera completa, apoyando eficientemente la toma de decisiones.

1.3 Modelo de asignación

Se tiene dos conjuntos con n entidades cada uno, representado por un grafo bipartito $G(V1, V2, E)$ (Wolsey, 1998), en el cual cada entidad de un grupo debe ser asignado a solo una entidad del otro, por ejemplo, n tareas (conjunto $V1$) que deben ser asignadas a n personas (conjunto $V2$). Cada posible asignación es representada con una arista (conjunto E), el cual tiene un costo asociado. El objetivo es minimizar el costo total luego de asignar cada una de las entidades, de tal manera que todas las personas tengan una tarea asignada. El modelo de programación entera para el problema de asignación es el presentado en ecuaciones (1)-(4):

$$z = \min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (1) \quad (\text{ec. 1.1})$$

$$\sum_{j=1}^n x_{ij} = 1 \quad \text{for } i = 1, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad \text{for } j = 1, \dots, n \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \text{for } i, j = 1, \dots, n \quad (4)$$

donde (1) es la función objetivo, que en este caso, consiste en minimizar el costo total de la asignación, en el cual c_{ij} corresponde a la matriz de costos de asignar cada elemento i a cada elemento j , (2) es el conjunto de restricciones que indican que a cada elemento i del grupo $V1$ se le puede asignar solo un elemento j del grupo $V2$, (3) es el conjunto de restricciones que indican que a cada elemento j del grupo $V2$ se le puede asignar sólo un elemento i del grupo $V1$, y (4) restringe la variable x_{ij} a valores 0 o 1. Si x_{ij} es igual a 1, se asigna el elemento i al elemento j . En la Figura 1 se observa un ejemplo del problema de asignación.

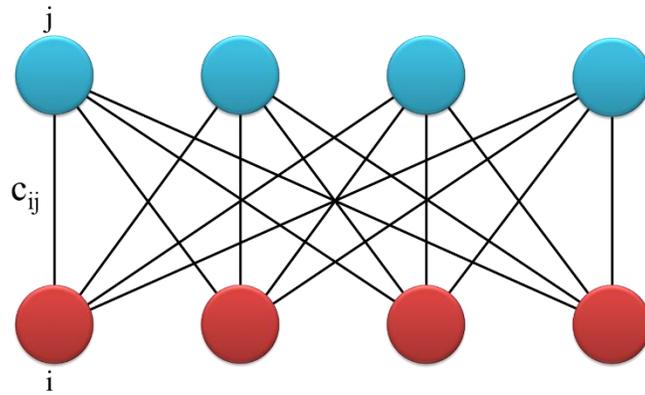


Figura 1. Posibilidades de asignación en grafo bipartito $G(V1, V2, E)$. Fuente: Elaboración propia.

Dado que la matriz de coeficientes del sistemas de ecuaciones mostrado en ec. 1.1 es totalmente unimodular, se puede resolver como un modelo de programación lineal (*linear sum assignment problem* (LSAP) en inglés) relajando el último grupo de restricciones (ver ec. 1.1 (4)) reemplazándolo por (Burkard & Çela, 1999):

$$x_{ij} \geq 0 \quad \text{for } i, j = 1, \dots, n \quad (\text{ec. 1.2})$$

Luego de esta modificación obtenemos una matriz unimodular doblemente estocástica, la cual forma un polígono de asignación que contiene al conjunto de soluciones factibles y que además sus puntos extremos coinciden con elementos en este conjunto. En la Figura 2 se muestra una representación de un polígono de asignación con estas características.

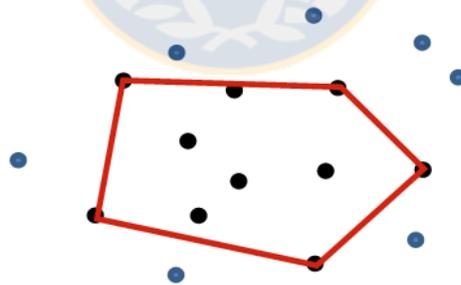


Figura 2. Polígono de asignación. Los puntos representan todas las soluciones. El polígono delimita la región de soluciones factibles, y entre sus vértices se encuentra el óptimo. Los vértices del polígono de asignación corresponden sólo a matrices de permutación. Fuente: Elaboración propia.

Para resolver el problema de asignación existen muchos métodos que utilizan algoritmos primales-duales, por ejemplo, el conocido método Húngaro (Kuhn, 1955) (ver sección 1.4.1), que tiene una complejidad $O(n^3)$.

1.4 Algoritmos para resolver el problema de asignación

Para resolver el problema de asignación existen en la literatura muchos algoritmos que pueden ser utilizados. La mayoría de los algoritmos secuenciales pueden ser clasificados en algoritmos primales-duales y algoritmos basados en simplex (Barr, Glover, & Klingman, 1997), (Cunningham, 1976), (Dantzig, 1963), siendo el método Húngaro el más conocido y utilizado. A continuación se presentarán algunos algoritmos agrupados según el tipo de enfoque con el cual es abordado.

1.4.1 Algoritmos Primales-Duales

En el libro (Burkard & Çela, 1999) se realiza una clasificación y definición de algunos de los algoritmos disponibles para resolver los LSAP. Los algoritmos primales duales trabajan con un par de soluciones primales no factibles x_{ij} , $x_{ij} \in \{0,1\}$, $1 \leq i, j \leq n$, y una solución dual factible u_i, v_j , $1 \leq i, j \leq n$, la cual cumple las condiciones de holguras complementarias (ec.1.3):

$$x_{ij}(c_{ij} - u_i - v_j) = 0, \quad \text{para } 1 \leq i, j \leq n \quad (\text{ec.1.3})$$

Dentro de los algoritmos primales-duales, tenemos:

(a) **Método Húngaro:** Fue el primer algoritmo primal-dual tratado en tiempo polinomial. En (Kuhn, 1955) Harold W. Kuhn presenta el conocido método Húngaro para la resolución de problemas de asignación. Basado en los trabajos de Dénes König y Jenő Egerváry, el algoritmo Húngaro construye una solución del problema primal partiendo de una solución no factible supraóptima (solución factible en el dual) buscando la factibilidad en cada iteración en base a las relaciones de las soluciones primal-dual.

(b) **Algoritmo de Jonker y Volgenant (JV):** Tiene una complejidad de $O(n^3)$ (Jonker & Volgenant, 1987). Esta metodología consta de 3 pasos: inicialización, aumento y ajuste de las variables duales.

1. La inicialización o preprocesamiento consta de tres pasos (Burkard, Dell'Amico, & Martello, 2009):

a. Reducción de columna: Se obtiene una primera solución parcial mediante la asignación de cada columna a una fila no asignada que corresponda a la columna de mínimo costo. Las columnas se recorren de manera inversa, es decir, de la columna n a la columna 1, de tal manera que las columnas con un índice menor tienen más posibilidades de no quedar asignadas, para que en pasos posteriores, en caso de empates, exista una mayor posibilidad de asignar el primer mínimo de la fila a una columna no asignada. Se inicializan las

variables u con valor cero, y las variables v con el costo mínimo de cada columna.

- b. Transferencia de reducción: Actualiza las variables duales v asociadas a las columnas ya asignadas, de tal manera que, cada fila asignada, tiene el costo reducido mínimo en al menos dos columnas diferentes. De esta forma, en el paso siguiente será más fácil mover la asignación de una fila para ampliar la solución.
 - c. Aumento de la reducción de filas: Se lleva a cabo una nueva iteración para cada fila perteneciente a un conjunto de filas no asignadas. En cada una de éstas se compara el menor y segundo menor costo reducido. Si tienen valores diferentes, la variable dual v que corresponde al mínimo es actualizada para que los dos costos reducidos resultantes sean iguales, y la fila en análisis sea asignada a la columna correspondiente a este costo. Si esta columna no había sido asignada anteriormente, entonces se continúa con la siguiente iteración en el conjunto de filas no asignadas. En caso contrario, se realiza una nueva iteración, pero con la fila a la cual esta columna había sido asignada.
2. Luego de la inicialización, el algoritmo JV utiliza una fase de aumento, en el que se recorren todas las filas no asignadas. Comienza encontrando un camino alternado, que es una secuencia de índices de filas y columnas alternadas. La primera fila no se encuentra asignada, la última es una columna, y entre estas dos hay columnas y filas asignadas en pares sucesivos. Si la última columna no ha sido asignada, se puede aumentar la solución parcial asignando todas las filas en el camino a las columnas subsiguientes, lo que conduce a una nueva asignación.
 3. Finalmente, en el tercer paso se ajustan las variables duales, ya que deben cumplir con las condiciones de holgura complementaria.

- **Otros algoritmos:**

- **Auction algorithms (Algoritmos de subasta):** (Bertsekas, 1981) Es similar al método Húngaro, pero desde un punto de vista de programación matemática, hay una diferencia significativa. En el método Húngaro el valor de la función objetivo dual disminuye en cada iteración, pero en el método propuesto por Bertsekas, este valor puede aumentar en algunas iteraciones. Otra diferencia es que en el algoritmo Húngaro un vértice asignado de $V1$ a $V2$ permanece asignado el resto del algoritmo, en cambio en este algoritmo, algunos vértices asignados del grupo $V2$, en alguna iteración siguiente puede ser liberado. El método de subasta es pseudopolinomial y su tiempo de ejecución es de orden $O(n^2(n+R))$, donde $R = \max\{|c_{ij}|: 1 \leq i, j \leq n\}$.

- **Algoritmos de pseudoflujo:** Están basados en la formulación del LSAP como un problema de flujo de costo mínimo. Un pseudoflujo es un flujo que cumple con las restricciones de capacidad, pero no necesariamente cumple con las de conservación de flujo. Los algoritmos de pseudoflujo trabajan con las relajaciones ε del problema de flujo de costo mínimo, donde las restricciones de conservación de flujo son violadas por a lo más ε . La solución inicial se transforma iterativamente en una solución óptima para el problema relajado. Luego, ε es disminuido y se repite el proceso hasta que alcanza un valor $1/n$. En ese punto la solución óptima del problema relajado, es también una solución óptima para el problema original. Algunos ejemplos son: (Orlin & Ahuja, 1992), (Goldberg, Plotkin, & Vaidya, 1993) y (Goldberg & Kennedy, 1995). Estos algoritmos son particularmente eficientes para problemas con matrices dispersas.

1.4.2 Algoritmo Dinic–Kronrod

Previo a la implementación $O(n^3)$ del algoritmo Húngaro, Dinic y Kronrod (Dinic & Kronrod, 1969) habían propuesto un enfoque diferente, totalmente independiente de la teoría de la dualidad de programación lineal. Aunque ignorado durante mucho tiempo (el algoritmo se presenta de una manera muy vaga, con varias imprecisiones en la traducción en Inglés), éste es el primer algoritmo de tiempo $O(n^3)$ para LSAP.

Dado un vector Δ (posiblemente cero), se selecciona un elemento a_{ij} en cada fila de la matriz de costos, denominado mínimo de Δ si cumple la condición de la ecuación 1.4 (ver Figura 3):

$$\forall 1 \leq k \leq n \quad a_{ij} - \Delta_j \leq a_{ik} - \Delta_k \quad (\text{ec.1.4})$$

	1	2	3	4
1	2	5	4	1
2	9	8	10	2
3	12	15	7	4
4	7	8	9	3
Δ	3	7	3	1

→

	1	2	3	4
1	2	5	4	1
2	9	8	10	2
3	12	15	7	4
4	7	8	9	3

Figura 3. Ejemplo de cálculo del elemento mínimo de Δ en cada fila (Dinitz, 2012).

Si todos los elementos seleccionados se encuentran en diferentes columnas entonces, la solución es óptima. Si no, se define la “deficiencia” de la solución como el número de columnas sin un elemento asignado. A continuación, se selecciona una de estas columnas, uno o más valores Δ_j son incrementados, y se obtiene una nueva solución con la deficiencia reducida en uno. Para ello

comenzamos utilizando un vector Δ de n ceros y mantenemos los mínimos encontrados. Se selecciona una de las columnas sin un elemento asignado, a la cual llamaremos s_1 . Luego incrementamos Δ_{s_1} con el valor máximo posible, sin que el elemento seleccionado anteriormente en la fila deje de ser mínimo. De esta manera, si recalculando con la ecuación (ec. 1.4) se obtiene otro posible mínimo, y se modifica. Paso siguiente, seleccionamos la columna del mínimo que acabamos de liberar y realizamos la misma operación. El proceso se itera hasta encontrar una columna con más de una fila asignada, y la solución se actualiza apropiadamente por el desplazamiento de las asignaciones alternativas.

1.4.3 Implementación “Shortest path” del algoritmo Húngaro

A principios de 1960, Hoffman y Markowitz (Hoffman & Markowitz, 1963) observaron que un LSAP puede ser resuelto a través de una secuencia de n caminos más cortos (shortest paths) en las matrices de costos de tamaño creciente desde 1×1 hasta $n \times n$. En su metodología, iteran resolviendo sub-matrices del problema original. Cada búsqueda del camino más corto requiere tiempo $O(n^3)$.

1.4.4 Algoritmos primales

Son tal vez menos eficientes que los algoritmos anteriores. Recorren las soluciones factibles hasta encontrar la mejor solución. El método Simplex, es un método iterativo que permite mejorar la solución en cada paso. La razón matemática de esta mejora radica en que el método consiste en caminar del vértice de un poliedro a un vértice vecino de manera que aumente o disminuya (según el contexto de la función objetivo, sea maximizar o minimizar), dado que el número de vértices que presenta un poliedro solución es finito siempre se encuentra solución. Dentro de esta clasificación se dispone de los siguientes algoritmos:

- **Algoritmos no basados en simplex:** (Balinski & Gomory, 1964; Cunningham & Marsh, 1978; Srinivasan & Thompson, 1977)
- **Algoritmos basados en simplex:** (Akgül, 1993; Barr et al., 1997; Cunningham, 1976; Orlin & Ahuja, 1992)

1.4.5 Algoritmos duales

Los métodos duales se usan con el objetivo de obtener una formulación de un problema dual que pueda conciliar dos aspectos aparentemente en conflicto: Por un lado, proporcionar información relevante del problema primal de forma que las cotas inferiores que obtenidas sean suficientemente

ajustadas, por otro lado, que la estructura del problema dual alcanzado resulte sencilla y permita obtener las cotas inferiores de forma eficiente. Dentro de esta clasificación se encuentran los siguientes algoritmos:

- **Algoritmos no basados en simplex:** (Hung & Rom, 1980; Thompson, 1981)
- **Algoritmos basados en simplex:** (Balinski, 1985; Goldberg, Plotkin, & Vaidya, 1993; Paparrizos, 1991)

1.4.6 Otros trabajos

En (Ji, Lee, & Li, 1997) los autores proponen un nuevo algoritmo para el problema de asignación basado en operaciones con matrices. El algoritmo es similar al método Simplex dual, pero se adopta un formato de matriz compacta en vez de la tabla simplex. Debido a esta similitud, su complejidad es exponencial. En comparación con el método Húngaro, el algoritmo no requiere encontrar el número mínimo de líneas en una matriz de costos reducidos. La desventaja del algoritmo es que se requiere una matriz $2n \times 2n$ con el fin de resolver un problema de asignación de $n \times n$. En (Mills-Tetty, Stentz, & Dias, 2007) presentan el algoritmo Húngaro dinámico, una nueva modificación al comúnmente utilizado algoritmo, aplicable a la solución óptima del problema de asignación en situaciones con cambios en los costos de los arcos o pesos. Finalmente, concluyen que la versión dinámica es más eficiente mediante la reparación de la solución inicial obtenida antes de que ocurran cambios en los costos.

En los siguientes trabajos se propone un algoritmo para encontrar la mejor asignación en un grafo bipartito. En (Fukuda & Matsui, 1994) los autores proponen un algoritmo para encontrar la asignación perfecta usando el método de partición binaria y en (Balakrishnarajan & Venuvanalingam, 1996) los autores proponen, un algoritmo basado en la búsqueda empleando nuevas heurísticas para la generación rápida. Esto se hace mediante una selección inteligente de los arcos para la generación de los nodos del árbol semántico asociado.

1.5 Algoritmos de secuenciamiento

Uno de los objetivos de este trabajo es encontrar no sólo la solución óptima al problema de semi-asignación, sino que una lista de k soluciones ordenadas según su resultado. De esta forma, luego de buscar minimizar el costo total de la asignación, se obtendría un conjunto de soluciones cuyo costo esté ordenado en orden no decreciente. Esta información puede ser necesaria en caso de que la asignación óptima no cumpla con todos los requerimientos del sistema donde se desea aplicar, y que

no pudieron ser incluidos en la formulación del problema (como por ejemplo, características cualitativas). En este caso, el tomador de decisiones puede evaluar otros aspectos de la solución que pueden no estar reflejados en la solución matemática entregada por el algoritmo de asignación.

En la literatura se pueden encontrar trabajos sobre algoritmos de secuenciamiento, muchos de éstos están basados en el algoritmo de Murty. A continuación presentamos los algoritmos más relevantes para este estudio, entre los que se elegirá el algoritmo base a utilizar para el secuenciamiento.

1.5.1 Algoritmo de Murty

A partir de una asignación de costo mínimo $\mathbf{a}(1)$ (ec. 1.5) encontrada con el método Húngaro u otro algoritmo que entregue la solución óptima del problema, se generan nodos de M_1 a M_{n-1} , también denominado particionamiento, imponiendo y prohibiendo asignaciones, como se muestra en ec. 1.6-1.8 para el primero nodo (Murty, 1968).

$$\mathbf{a}(1) = \{(i_1, j_1), \dots, (i_n, j_n)\} \quad (\text{ec. 1.5})$$

$$M_1 = \{(\overline{i_1, j_1})\}, \quad (\text{ec. 1.6})$$

$$M_2 = \{(i_1, j_1); (\overline{i_2, j_2})\}, \quad (\text{ec. 1.7})$$

⋮

$$M_r = \{(i_1, j_1); \dots; (i_{r-1}, j_{r-1}); (\overline{i_r, j_r})\} \text{ para } r=1, \dots, n-1 \quad (\text{ec. 1.8})$$

Estos nodos tienen las siguientes propiedades:

1. M_1, \dots, M_e son mutuamente excluyentes, y ninguno de éstos contiene a $\mathbf{a}(u)$, $u=1, \dots, k$, que corresponden a las asignaciones previamente encontradas.
2. $A = \bigcup_{u=1}^{u=k} \{\mathbf{a}(u)\} \cup \bigcup_{v=1}^{v=e} M_v$, donde A es el conjunto de todas las asignaciones.

Las celdas $(i_1, j_1), \dots, (i_r, j_r)$ son establecidas para ser contenidas en la asignación en el nodo M , y las celdas $(\overline{i_1, j_1}), \dots, (\overline{i_{n-1}, j_{n-1}})$ son establecidas para ser excluidas de cada asignación en el nodo M . La barra sobre la celda indica que debe ser excluida de la asignación.

Específicamente, la lista se genera prohibiendo el primer par asignado (por ejemplo $(\overline{i_1, j_1})$ en la ecuación 1.6), y encontrando la asignación de menor costo para el subproblema. Luego, en M_2 (ec.1.7), se impone el par (i_1, j_1) , y se prohíbe el siguiente par (i_2, j_2) buscando nuevamente la asignación de menor costo. A continuación, la lista se completa hasta prohibir el penúltimo par de la asignación $\mathbf{a}(1)$, fijando las anteriores. A partir de esta lista se selecciona la asignación con menor

costo, que pasa a ser parte de la secuencia devuelta por el algoritmo como asignación $a(2)$, al final de la ejecución. Esta se encuentra utilizando la ecuación (1.9):

$$a(k + 1) = b_{M_d} \quad (\text{ec. 1.9})$$

donde b_{M_d} representa una asignación de costo mínimo en el nodo M_d , y Z_{M_d} es su costo. M_d es cualquier nodo en la lista, determinado por la ecuación (1.10):

$$Z_{M_d} = \min_{i=1 \text{ a } e} [Z_{M_i}] \quad (\text{ec. 1.10})$$

A partir de esta asignación encontrada se genera una nueva lista, como se indicó anteriormente, la que se suma a las asignaciones que fueron encontradas, pero no seleccionadas, para el nodo anterior. En la Figura 4, se muestra un ejemplo de la selección entre los nodos generados por el algoritmo. El tiempo de ejecución para este algoritmo es $O(kn^4)$.

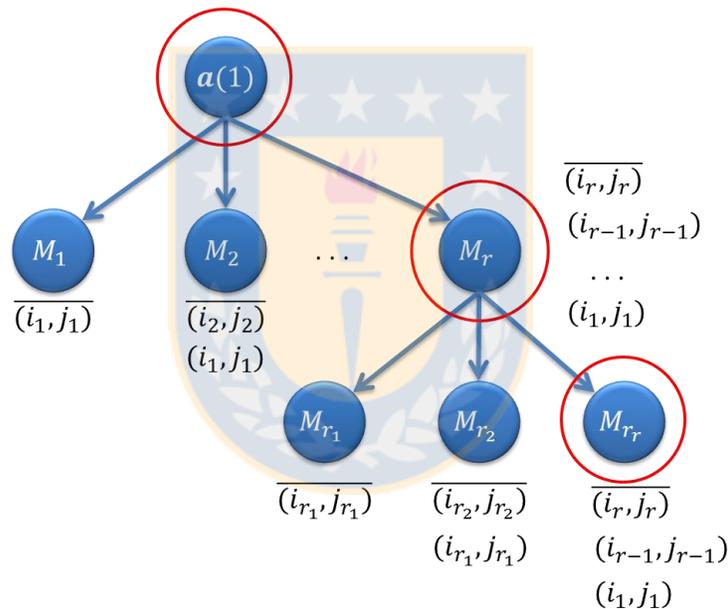


Figura 4. Ejemplo de selección de soluciones a partir de los nodos encontrados en el algoritmo de Murty (Burkard et al., 2009).

1.5.2 Algoritmo de Chegireddy-Hamacher

En la publicación (Chegireddy & Hamacher, 1987), primero explican el método general para el problema de k -best perfect matching, o las k mejores soluciones para el problema de emparejamiento, y luego presentan su propuesta de implementaciones alternativas para encontrar las segundas mejores soluciones.

Se utiliza el particionamiento del espacio de solución, prohibiendo uno a uno los arcos en la mejor asignación, e imponiendo todos los arcos considerados anteriormente en la solución. Crean las

ramas alternando entre el imponer y prohibir cada uno de estos arcos. La segunda mejor solución es encontrada dentro de una de estas ramas.

Este algoritmo presenta una complejidad $O(kn^3)$.

1.5.3 Algoritmo de Pedersen et al.

Pedersen en su trabajo (Pedersen, Relund Nielsen, & Andersen, 2008) utiliza un algoritmo de reoptimización a partir del algoritmo de particionamiento de Murty. El pseudocódigo se encuentra en la Figura 5. En éste, se calcula una asignación parcial $a(i)$ y sus valores duales (u, v) , para luego ser modificados en la tercera línea del pseudocódigo. Posteriormente, se construye la red restante que corresponde a $a(i)$ y a la solución dual. A continuación, se encuentra el camino de aumento mínimo (o *shortest augmenting path (SAP)*) y la solución correspondiente mediante el método de Dijkstra. Dado al uso de este método, la función *EncontrarOptimo* de la primera línea del pseudocódigo (ver Figura 5), se ejecuta en $O(kn^2)$. Finalmente este algoritmo tiene una complejidad $O(kn^3)$.

```

1  Procedimiento EncontrarOptimo ( $\bar{A}^i$ )
2     $a(i) := \text{CrarParcial}(\hat{a});$ 
3     $(u, v) := \text{ModificarDual}(\hat{u}, \hat{v});$ 
4     $\bar{G}(a(i)) := \text{ConstruirRedResidual}(a(i), u, v);$ 
5     $P := \text{EncontrarCaminoAumento}(\bar{G}(a(i)));$ 
6     $a := \text{AumentarSolucion}(P)$ 
7  Termino Procedimiento

```

Figura 5. Pseudocódigo del algoritmo de reoptimización de Pedersen et al. (Pedersen et al., 2008).

1.6 Problema de semi-asignación

El caso abordado en el presente trabajo contiene una variación, que consiste en que dentro del conjunto existen *clusters* o subgrupos, es decir, existe el mismo costo para más de una entidad (ver Figura 6).

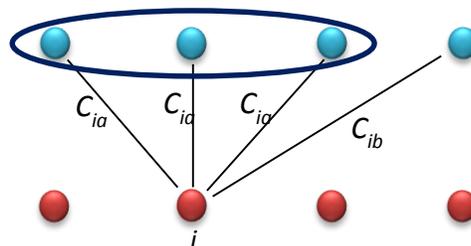


Figura 6. Ejemplo de igualdad de costos dentro de los *clusters* en un problema de asignación. c_{ia} corresponde al costo de asignar el elemento i a algún elemento en el *cluster* a y c_{ib} corresponde al costo de asignar el elemento i a un elemento fuera del *cluster*. Lo mismo sucede con los otros elementos a asignar (diferentes a i), pero los costos asociados no fueron incluidos en la imagen para mejor visualización. Fuente: Elaboración propia.

El problema de asignación corresponde a la asignación de n trabajos a m personas, donde n es igual a m , pero en el problema de semi-asignación, n es mayor a m . Éste se puede aplicar en áreas como planificación de mano de obra, programación de tareas, cálculo del presupuesto de capital y planificación, y planificación de proyectos (Kennington & Wang, 1992).

La formulación matemática del problema sería de la siguiente forma:

$$\text{Min} \quad \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \quad (\text{ec. 1.11})$$

$$\text{s. a.} \quad \sum_{j=1}^m x_{ij} = 1 \quad \forall i = 1, \dots, n \quad (\text{ec. 1.12})$$

$$\sum_{i=1}^n x_{ij} = b_j \quad \forall j = 1, \dots, m \quad (\text{ec. 1.13})$$

$$x_{ij} \geq 0 \quad \forall i = 1, \dots, n; \forall j = 1, \dots, m \quad (\text{ec. 1.14})$$

donde b_j representa la capacidad del grupo j . El modelo tiene una solución factible sólo si $\sum_{i=1}^m b_j = n$.

En el trabajo de Kennington y Wang (Kennington & Wang, 1992) se comienza explicando brevemente qué son los problemas de semi-asignación y sus aplicaciones, para luego continuar mencionando algoritmos que los resuelven como: el algoritmo simplex para redes, simplex primal, simplex dual, primal-dual, algoritmo “*out-of-killer*” o el camino de aumento mínimo (SAP). Posteriormente presentan un algoritmo del tipo “*shortest augmenting path*” y lo compara con los algoritmos existentes demostrando, que éste resulta ser más rápido que todos los demás, y tiene el mejor tiempo polinomial de ejecución en el peor de los casos.

El algoritmo consiste en cuatro fases: reducción de columna (*column reduction*), transferencia de reducción (*reduction transfer*), aumento de reducción de filas (*row reduction augmentation*) y camino de aumento más corto (*shortest augmenting path*). Si la primera fase resuelve el problema, entonces, el algoritmo se termina. De la misma manera ocurre con las siguientes fases.

A continuación se explica brevemente cada una de las fases:

- Reducción de columna: Heurística que entrega un conjunto de asignaciones parciales junto a variables duales factibles que satisfagan las condiciones de holguras complementarias. En esta fase, si un elemento i (del conjunto de los n trabajos) no ha sido asignado (en el comienzo del algoritmo ninguno ha sido asignado) y si el elemento del grupo j (del conjunto de las m personas) cuyo costo es el mínimo tampoco lo ha

vido, entonces asignamos el elemento i al grupo j .

- **Transferencia de reducción:** En esta fase se reducen las variables duales de grupos j saturados en alguna cantidad $\Delta > 0$, y esta misma cantidad es utilizada para incrementar las variables duales de los trabajos. La cota superior para Δ se determina en base a las condiciones de dualidad.
- **Aumento de reducción de fila:** El procedimiento básico reduce el número de tareas no asignadas, manteniendo las condiciones de dualidad y de holguras complementarias. En este trabajo se modifica este procedimiento, permitiendo la asignación a grupos j saturados para ser cambiados de un elemento i a otro. Además se agregó el procedimiento de reducción de fila de tal manera que un elemento i no asignado pudiera encontrar una persona j en la siguiente iteración de la aplicación de la reducción de filas.
- **Camino de aumento más corto:** Se usa para completar las últimas asignaciones restantes. En cada iteración el procedimiento selecciona un elemento no asignado i , y construye un camino alternado mínimo de i a un grupo j no saturado, usando el costo reducido como el largo de los arcos. Luego, se aumenta la asignación a través de este camino a sus sucesivos grupos j . En cada iteración se reduce en uno el número de elementos i no asignados.

1.7 Simetría y planos de corte

Un programa lineal entero (ILP) es simétrico si sus variables pueden ser permutadas sin cambiar la estructura del problema, cuyas permutaciones en conjunto forman el grupo simétrico (S_n). Por ejemplo, el triángulo equilátero tiene un grupo simétrico con 6 permutaciones (ver Figura 7). Éstas son: identidad; rotación 120° ; rotación 240° ; reflexión; reflexión más rotación 120° ; y reflexión más rotación de 240° . Cualquiera de estas transformaciones que utilicemos, mantienen la estructura inicial del triángulo.

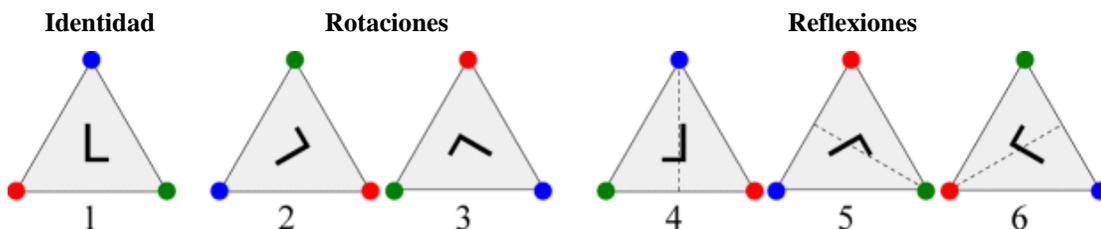


Figura 7. Representación del conjunto de simetrías de un triángulo equilátero (“Triangle Fractals,” 2005).

Los ILP simétricos frecuentemente aparecen cuando se formulan problemas clásicos en combinatoria u optimización. Aquellos relativamente pequeños son extremadamente difíciles de

resolver usando códigos *branch-and-cut*, los cuales no consideran la simetría del problema. Lógicamente, el problema se vuelve aún más complejo de resolver cuando aumenta la cantidad de elementos.

En algunos casos esta característica puede suponer una ventaja en la resolución del problema si se utiliza de forma correcta. Para esto en la literatura se han propuesto algunas herramientas tales como perturbación, fijación de variables, desigualdades de ruptura de simetría y la poda del árbol de enumeración (Jünger et al., 2010).

En el secuenciamiento de soluciones para el problema de semi-asignación tratado en este estudio también están presentes las simetrías, debido a que, existen grupos dentro de los cuales, el costo de asignación para cada uno de los elementos contenido en éste, es el mismo. Si permutamos cada elemento dentro del grupo, obtendremos el mismo costo total de asignación. La cantidad de soluciones con igual costo es $\prod_{j=1}^m b_j !$, donde m es la cantidad de *clusters* y b_j es el número de elementos en el *cluster* j . Este aspecto, correspondería a que las tareas serían asignadas al mismo grupo, lo cual no cumple con el objetivo que se busca en el secuenciamiento de soluciones, que es obtener una asignación distinta, es decir, asignando la falla a otra compañía y no a otro elemento dentro de la misma. Además, esto genera un aumento en el tiempo de ejecución con respuestas que no son las requeridas.

En el trabajo publicado por Medina Durán et al. (Medina Durán, Malaguti, & Duran-Faundez, 2017), se presentan técnicas para el secuenciamiento de soluciones en problemas de semi-asignación. Uno de ellos es una propuesta de un nuevo algoritmo basado en el algoritmo de Murty, aplicando cortes de simetría para evitar la generación de soluciones con igual costo. Estos cortes se realizan cuando se generan nuevos nodos, prohibiendo e imponiendo asignaciones contenidas en la mejor solución (ver sección 1.5.1). En lugar de prohibir solo una asignación, se prohíben todas aquellas que incluyan elementos dentro del mismo grupo.

CAPÍTULO 2: Propuesta

El problema abordado consiste en asignar monitores multiparámetros que requieren mantenimiento correctivo a proveedores externos de servicio técnico. Se eligió este equipo médico debido a que es indispensable para el monitoreo continuo de los signos vitales de los pacientes, por lo que es altamente solicitado. Cada empresa tiene una cantidad limitada de técnicos, a los cuales se les debe asignar la falla. A continuación se presenta la propuesta:

2.1 Hipótesis

Se puede reducir el costo total y el tiempo de reparación de los equipos médicos implementando un algoritmo de optimización para el secuenciamiento de soluciones del problema de semi-asignación.

2.2 Objetivos

2.2.1 Objetivo general

Asignar de manera eficiente los equipos médicos a los proveedores de servicio técnico utilizando un algoritmo para el secuenciamiento de soluciones y que resuelva el problema de priorización de asignaciones con grupos de igual costo y obtenga una mayor cantidad de asignaciones diferentes en menos tiempo, comparado con la metodología usada actualmente.

2.2.2 Objetivos específicos

- Estudio de los algoritmos de secuenciamiento de soluciones para la priorización de asignaciones existentes en la literatura.
- Modificación de algoritmos para considerar grupos con costos iguales.
- Evaluar el desempeño del algoritmo con una instancia derivada de un caso real de asignación de equipo médico para mantenimiento.

2.3 Alcances y limitaciones

Este proyecto propone una aplicación del algoritmo de Murty para problemas de semi-asignación con *clusters*. El hospital utilizado para obtener los datos del caso de estudio, es de mediana y alta complejidad. Se reserva su identidad debido a la confidencialidad solicitada por éste.

2.4 Metodología

El algoritmo de secuenciamiento basado en el algoritmo de Murty (Murty, 1968), descrito en la publicación de Medina Durán et al. (Medina Durán et al., 2017), y el algoritmo usado para obtener la solución óptima es el de Jonker and Volgenant (JV) (Jonker & Volgenant, 1987), de tipo *Shortest Augmenting Path*. El computador utilizado para correr las instancias es el servidor HP ProLiant ML310e Gen8, con procesador Intel® Xeon™ Quad-Core E3-1220v2 de 3.1 GHz, 8MB Level 3 cache - 69 Watts, E3-1200 Series, con 4 GB de memoria instalada (RAM) y sistema operativo Linux. Para la implementación del algoritmo se usó el lenguaje de programación C++. Para la recopilación de datos se realizaron entrevistas y se revisó documentación de la Unidad de Equipos Médicos del hospital en estudio.



CAPÍTULO 3: Desarrollo metodológico

El problema a tratar dispone de dos características principales: secuenciamiento de soluciones (sección 3.2) y semi-asignación (sección 3.3). Para ello se utilizan herramientas que tratarán cada una de éstas, las que serán explicadas a continuación.

3.1 Algoritmo de asignación

Se debe encontrar la solución óptima del problema de semi-asignación, utilizando un algoritmo que lo resuelva de forma eficiente. Para ello se recurre a la metodología presentada por Jonker y Volgenant, con una complejidad de $O(n^3)$ (Jonker & Volgenant, 1987). De acuerdo a lo mencionado en la sección 1.4.1, este algoritmo dispone de 3 etapas: inicialización o preprocesamiento, aumento y ajuste de las variables duales. Para mayor compresión se utiliza una instancia pequeña con *clusters*, para ejemplificar el procedimiento.

Ejemplo: Buscamos la solución óptima de la siguiente matriz (ver Figura 8) utilizando el algoritmo JV. La matriz es la representación de los costos asociados a asignar 7 tareas (filas) a 3 grupos (columnas), con 2, 3 y 2 personas respectivamente.

	0	1	2	3	4	5	6
0	54	54	57	57	57	43	43
1	36	36	41	41	41	46	46
2	35	35	32	32	32	28	28
3	23	23	29	29	29	37	37
4	50	50	48	48	48	52	52
5	38	38	33	33	33	32	32
6	37	37	40	40	40	42	42

Figura 8. Matriz de costos de problema usado para ejemplificar procedimiento del algoritmo JV. Fuente: Elaboración propia.

Todas las variables duales son inicializadas asignándoles el valor 0.

Durante la fase de inicialización, luego la reducción de columna, se realizan 2 asignaciones: $x_{34}=1$ y $x_{26}=1$. Como se mencionó, las columnas se recorren en orden inverso, y en cada una de ellas se busca la fila con el costo mínimo. Si esta fila no ha sido asignada, se asigna a la columna. Las variables duales actualizadas se muestran en la Tabla 1. Variables duales, donde los números en la primera fila corresponden al subíndice de cada uno de ellas. Éstas a su vez corresponden al índice de cada fila, en el caso de las variables duales u y a cada columna en el caso de las variables duales v .

Tabla 1. Variables duales luego del primer paso de la inicialización (1.a) del algoritmo JV. Fuente: Elaboración propia.

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>u</i>	0	0	0	0	0	0	0
<i>v</i>	23	23	29	29	29	28	28

En el segundo paso de la inicialización, se busca que las filas asignadas tengan el mismo costo reducido en al menos dos de ellas, pero debido a que el ejemplo, tiene *clusters*, esta condición ya se cumple. Lo mismo sucede con el tercer paso de la inicialización, ya que hay columnas con los mismos costos reducidos, no se produce la transferencia de reducción. Solo se actualizan las variables duales, las cuales se muestran en la Tabla 2.

Tabla 2. Variables duales luego del tercer paso de la inicialización (1.c) del algoritmo JV. Fuente: Elaboración propia.

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>u</i>	15	12	0	0	19	4	11
<i>v</i>	23	23	29	29	29	28	28

En la fase de aumento se recorren todas las filas no asignadas, es decir, las filas: 0, 1, 4, 5 y 6. A continuación se muestran las actualizaciones de las variables para cada una de ellas:

$$i=0: x_{05}=1$$

$$i=1: x_{12}=1$$

$$i=4: x_{43}=1$$

$$i=5: v_2=33; v_3=33; v_4=33; v_5=32; v_6=32; x_{30}=1; x_{34}=0; x_{54}=1$$

$$i=6: v_2=26; v_3=26; v_4=26; v_5=20; v_6=20; x_{11}=1; x_{12}=0; x_{62}=1$$

Por consiguiente, los valores de las variables duales luego del segundo paso del algoritmo JV son los siguientes (ver Tabla 3):

Tabla 3. Variables duales luego del segundo paso del algoritmo JV. Fuente: Elaboración propia.

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>u</i>	15	12	0	0	19	4	11
<i>v</i>	23	23	26	26	26	20	20

En la Tabla 4 se muestran los valores de las variables duales luego de la tercera etapa del algoritmo JV. En la Figura 9 se muestra la asignación resultante.

Tabla 4. Variables duales luego del tercer paso del algoritmo JV. Fuente: Elaboración propia.

	<i>0</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>
<i>u</i>	23	13	8	0	22	7	14
<i>v</i>	23	23	26	26	26	20	20

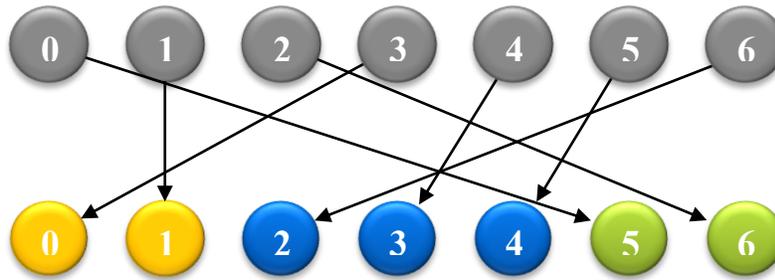


Figura 9. Asignación final luego de aplicar algoritmo JV. Fuente: Elaboración propia.

3.2 Algoritmo de Murty

Este algoritmo fue el primero en tratar el problema de secuenciamiento de soluciones (Murty, 1968), de ahí que muchos trabajos presentes en la literatura se basen en éste para desarrollar su metodología.

Tal como se muestra en la sección 1.5.1, a partir de una solución óptima, en este caso la entregada por el algoritmo de Jonker y Volgenant, se generan nodos, en los cuales se prohíben e imponen asignaciones, para crear nuevas soluciones.

3.3 Cortes de simetría

En una instancia donde existen grupos de elementos con el mismo costo asociado, la secuencia de soluciones entregada por el algoritmo anterior contiene costos repetidos, debido a la simetría del problema. Para obtener asignaciones con variaciones a nivel de *clusters* y reducir el tiempo de cómputo, al momento de prohibir una de las asignaciones, se prohíbe no solo una, sino todas aquellas que pertenecen al mismo grupo, es decir, que tienen el mismo costo asociado (Medina Durán et al., 2017). De esta forma nos aseguramos que la siguiente asignación no sea a uno de los elementos del mismo grupo, y la asignación a los *clusters* no sea la misma que la anterior.

Ejemplo: Continuando con el ejemplo planteado en la sección 3.1, se aplica el algoritmo de Murty con los cortes de simetría, seleccionando la segunda solución en base al primer nodo. Cabe recordar que se deben prohibir todas las asignaciones del mismo grupo, cuando corresponda, en lugar de solo prohibir la columna correspondiente en el nodo. En seguida se muestran los nodos generados a partir

de la mejor asignación:

$$a(1) = \{(0,5),(1,1),(2,6),(3,0),(4,3),(5,4),(6,2)\} = 251$$

$$M_1 = \{(\overline{0,5}), (\overline{0,6})\} = \{(0,4),(1,1),(2,6),(3,0),(4,3),(5,5),(6,2)\} = 264$$

$$M_2 = \{(0,5); (\overline{1,0}), (\overline{1,1})\} = \{(0,5),(1,2),(2,6),(3,0),(4,3),(5,4),(6,1)\} = 253$$

$$M_3 = \{(0,5),(1,1); (\overline{2,5}), (\overline{2,6})\} = \{(0,5),(1,1),(2,4),(3,0),(4,3),(5,6),(6,2)\} = 254$$

$$M_4 = \{(0,5),(1,1),(2,6); (\overline{3,0}), (\overline{3,1})\} = \{(0,5),(1,1),(2,6),(3,2),(4,3),(5,4),(6,0)\} = 254$$

$$M_5 = \{(0,5),(1,1),(2,6),(3,0); (\overline{4,2}), (\overline{4,3}), (\overline{4,4})\} = \text{Infinito}$$

$$M_6 = \{(0,5),(1,1),(2,6),(3,0),(4,3); (\overline{5,2}), (\overline{5,3}), (\overline{5,4})\} = \text{Infinito}$$

Los nodos M_5 y M_6 no tienen una solución factible, es por eso que su costo es mostrado como infinito.

Finalmente, la segunda asignación es:

$$a(2) = \{(0,5),(1,2),(2,6),(3,0),(4,3),(5,4),(6,1)\} = 253$$



CAPÍTULO 4: Caso de estudio

4.1 Descripción del caso

El hospital a usar en el estudio de caso dispone de un gran número de equipos médicos, los cuales cada cierto período deben ser sometidos a mantención para que su funcionamiento se encuentre acorde con sus requerimientos. Dependiendo de la relevancia del equipo, el no realizar una adecuada mantención puede producir, por ejemplo, errores en lecturas de signos vitales, de administración de medicamentos o de parámetros de medición de la seguridad eléctrica del equipo, entre otros, que pueden causar un grave daño a la salud del paciente, que en algunos casos podría ser fatal.

Cuando un equipo falla, se le debe realizar una mantención correctiva en el menor tiempo posible, para que el equipo vuelva a ser útil al centro. El decidir a quién se le solicitará el servicio técnico no es una tarea fácil, debido a que existen varias ofertas, las cuales pueden ser muy diferentes en cuanto al precio y a lo que incluyen. Además, la asignación de costo mínimo que pueda ser arrojado por un algoritmo de asignación, puede no ser la adecuada, ya que existen factores que son difíciles de incluir en el costo, y por lo tanto, no son considerados en la solución, como por ejemplo características cualitativas de las empresas. En base a esto, es adecuado usar algoritmos de secuenciamiento de soluciones, y permitir al tomador de decisiones elegir una mejor asignación.

En las instancias a usar para comprobar el algoritmo, las empresas de mantenimiento corresponderán a los *clusters*, y cada técnico disponible será considerado como un elemento dentro del *cluster*, quienes tendrán el mismo costo de asignación asociado.

4.2 Monitores Multiparámetro

También denominado monitor de signos vitales (ver Figura 10) es un dispositivo que permite detectar, procesar y desplegar en forma continua los parámetros fisiológicos del paciente. Incluye un sistema de alarmas que alertan cuando existe alguna situación adversa o fuera de los límites deseados (Centro Nacional de Excelencia Tecnológica en Salud, 2005). Es un dispositivo necesario en las salas UCI y UTI, así como lo son la bomba de infusión y el ventilador mecánico, por lo que sin ellos no se puede atender al paciente.

Se pueden encontrar muchos niveles de complejidad y de parámetros que muestra este dispositivo dependiendo de la unidad donde se use, o dependiendo de los signos vitales específicos a ser evaluados en el paciente. Sin embargo, comúnmente se muestran parámetros fisiológicos como: Electrocardiograma (ECG); frecuencia respiratoria; saturación de oxígeno (SpO₂); saturación venosa

de oxígeno (SvO₂); temperatura corporal; presión no invasiva; presión invasiva; dióxido de carbono (CO₂); gasto cardíaco; presión intracraneana (PIC), entre otros.

Los monitores multiparámetros pueden ser preconfigurados, modulares, o ambos. Los monitores preconfigurados tienen los parámetros a medir definidos por fábrica, por lo que no se pueden modificar; en cambio, si el monitor acepta la adición de módulos, se le puede añadir nuevos parámetros. La ventaja de los módulos, es que se puede obtener monitores menos complejos, y sólo en caso de ser necesario se agregan los módulos con las mediciones faltantes, lo que disminuye los costos al momento de adquirir el equipo. Esta variedad en las características de los monitores produce por consecuencia distintos tipos de fallas, pero aunque esto ocurra, debido a la relevancia de este equipamiento médico, existe la disponibilidad del servicio técnico que lo repare en caso de falla.



Figura 10. Ejemplo de monitor multiparámetro (“Monitor multiparámetro VISMO PVM-2701 - arquimed,” n.d.).

4.3 Supuestos

A modo de definir las características del problema, bajo las cuales se aplicará el algoritmo propuesto, se presentan los siguientes supuestos:

- Cada empresa puede tener más de un técnico disponible.
- Cada técnico puede revisar sólo un equipo a la vez.
- Hay empresas que son capaces de reparar distintas marcas y modelos de monitores.
- Las matrices de costos se crearán a partir de los costos de reparación de los monitores.
- Por protocolo del hospital, aquellos equipos cuyos precios de repuestos sean muy altos, sobre un 50% del costo del equipo, deben darse de baja, es decir, no necesitan reparación.
- Los costos de recibir mantención por distintos técnicos de una misma empresa son iguales.
- Los costos de los repuestos varían dependiendo de la marca del equipo.

4.4 Instancias

La instancia en la que se aplicó el algoritmo se basa en datos reales de un hospital de mediana y alta complejidad, en el que se debían asignar 329 monitores multiparámetros, que se encontraban con alguna falla y por lo tanto, se deben asignar a una empresa. El algoritmo de asignación contempla la capacidad de cada empresa (43 en total), en cuanto a la cantidad de técnicos que cada una posea a su disposición.

La asignación en el caso real es más bien empírica, asignando los primeros, que pueden ser los más complejos, a las empresas más grandes o con más trayectoria y que ofrezcan un menor costo, para luego asignar las fallas restantes a las empresas disponibles.



CAPÍTULO 5: Resultados

5.1 Asignación actual

En esta sección se muestra el resultado de una asignación realizada actualmente, sin el algoritmo de asignación. En el hospital se lleva a cabo una asignación más que nada intuitiva, y manual. En detalle, cada una de las fallas es asignada a una empresa, seleccionando primero las más conocidas y confiables para equipos complejos, y luego asignando las fallas restantes a empresas disponibles. Esto implica que existe además un componente aleatorio, debido al tamaño de la instancia, haciendo difícil decidir cuál es la mejor opción de asignación.

Para conseguir una respuesta se necesitó 2 horas y 19 minutos y el costo total fue \$85 450 405. La asignación se puede ver en la Tabla 6 del Anexo, y el gráfico con la información del costo por empresa se puede ver en la Figura 12, presente en el Anexo.

5.2 Asignación utilizando el secuenciamiento de soluciones

En la Tabla 5 se muestran los costos obtenidos luego de la ejecución del algoritmo (Medina Durán et al., 2017). Para esta instancia se solicitó un $k=10$. El tiempo de ejecución del algoritmo para obtener 10 costos diferentes, fue de 11,0836 segundos. Las 10 primeras soluciones se observan en la Tabla 7 del Anexo, y los gráficos con la información del costo por empresa se encuentran en las Figuras 13 - 22.

Tabla 5. Primeros 10 costos ordenados de forma creciente. Fuente: Elaboración propia.

<i>Solución</i>	<i>Costo</i>
1	39 133 599
2	39 133 989
3	39 134 194
4	39 134 584
5	39 134 700
6	39 134 799
7	39 134 934
8	39 134 995
9	39 135 189
10	39 135 295

5.3 Discusión

Aplicar el algoritmo disminuye a un 0,0013% el tiempo para asignar cada una de las fallas a las empresas correspondientes; además que, realizar la asignación de los monitores sin utilizar el

algoritmo, no asegura encontrar el costo mínimo, sino que solo asegura encontrar una solución posible. De esta forma, se logró reducir el costo de la asignación en un 54,20%, lo que significa una disminución de \$46 316 806 menos, con la mejor solución, con una diferencia de \$1696 con la décima mejor solución.

En la Figura 12 (ver Anexo) se observa que para 16 empresas se supera los \$2 000 000 en el costo, en cambio en las soluciones entregadas por la metodología propuesta (Figuras 13 - 22), solo 2 a 3 superan este valor. En las soluciones entregadas por el algoritmo propuesto (Figuras 13 - 22), los costos se modifican en algunos de los grupos (entre un 1% - 13%) y los otros grupos se mantienen con costo constante a lo largo de las 10 mejores soluciones. En la Figura 11 se observa qué grupos sufrieron modificaciones, y cuál fue su porcentaje de variación en el costo, calculado en base a su desviación estándar. Esto se debe a que algunas fallas pueden ser reparadas solo por algunas empresas, o porque el k no fue lo suficientemente grande para la variación de estos grupos.

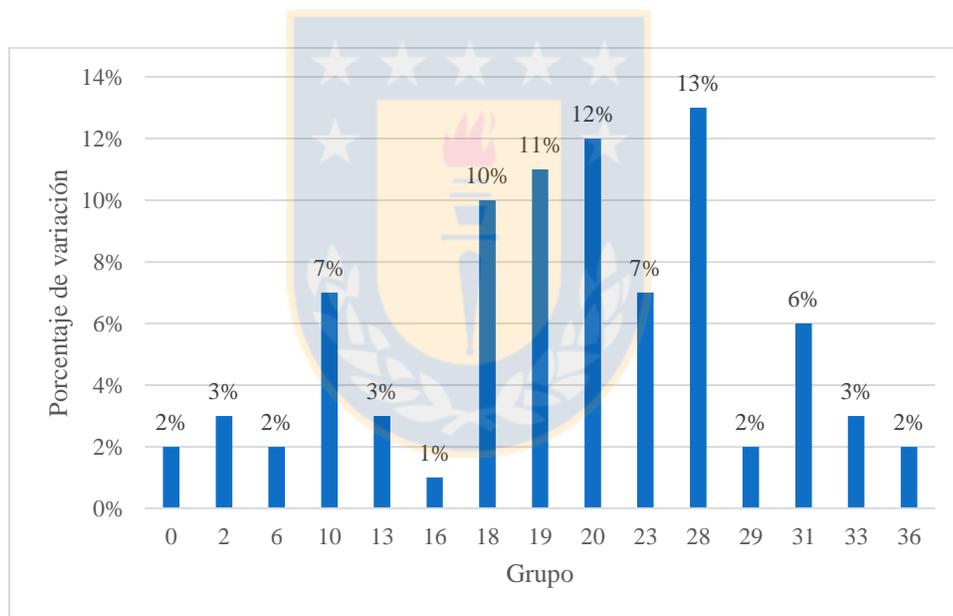


Figura 11. Porcentaje de variación de costo para cada uno de los grupos (empresas). Fuente: Elaboración propia.

De manera adicional, este algoritmo aplica cortes de simetría al momento de buscar la siguiente mejor asignación, por lo que comparado con los trabajos encontrados en el estado del arte, se utilizó una herramienta más eficiente.

El poder obtener no solo una, sino que 10 o más soluciones en segundos, otorga mayores posibilidades al tomador de decisiones. Si se presenta algún inconveniente, o duda que la primera asignación sea la mejor (por cualquier factor específico no cuantificable), éste tendrá la libertad de poder cambiar la selección, eligiendo la siguiente mejor en la lista, incrementando el costo en la menor cantidad posible.

CAPÍTULO 6: Conclusiones

El algoritmo propuesto logra una reducción considerable (de horas a menos de un segundo) del tiempo requerido para encontrar las k -ésimas asignaciones de menor costo, además de asegurar que las asignaciones propuestas sí representan la mejor combinación dentro de todas las posibilidades.

Al ser aplicados a un centro hospitalario, y aún más, de alta y mediana complejidad, se propicia un mejor uso de los recursos, de tal manera, que cuando se desee realizar mantenimiento correctivo a los monitores multiparámetros, puedan ser regresados en menos tiempo, y a su vez, el costo asociado se vea reducido.

Debido a que existen variables difíciles de cuantificar, existe la posibilidad de que una solución no se adapte a la situación del hospital en determinado momento. Por esta razón, la metodología utilizada nos entrega las k mejores asignaciones, lo que otorga una mayor flexibilidad al tomador de decisiones, y por lo tanto mayor control en cuanto a la selección de la opción que se adapte mejor al problema a resolver.

Por otro lado, en esta instancia solo se incluyeron monitores multiparámetros, pero puede ser aplicado a cualquier otro equipo médico, ya que para ellos solo se necesitaría realizar cotizaciones con las empresas para que pueda ser agregado a la instancia.

6.1 Trabajo Futuro

La siguiente fase más evidente, sería replicar el registro de costos de reparación de otros equipos, para que se ajuste a los equipos pertenecientes al centro hospitalario en el que se desee aplicar la metodología.

Para un informe más completo, la información entregada por el algoritmo propuesto se puede complementar con un registro de las fallas, fechas, costos y empresas asignadas, para poder generar gráficos y proyecciones de las posibles futuras fallas. Además, se puede crear una base de datos para que el algoritmo utilice costos actualizados por el usuario.

BIBLIOGRAFÍA

- Akgül, M. (1993). A genuinely polynomial primal simplex algorithm for the assignment problem. *Discrete Applied Mathematics*, 45, 93–115.
- Balakrishnarajan, M. M., & Venuvanalingam, P. (1996). Heuristic enhancements of the search for the generation of all perfect matchings. *Applied Mathematics Letters*, 9(2), 49–53. [https://doi.org/10.1016/0893-9659\(96\)00011-0](https://doi.org/10.1016/0893-9659(96)00011-0)
- Balinski, M. L. (1985). Signature methods for the assignment problem. *Operations Research*, 33, 527–537.
- Balinski, M. L., & Gomory, R. E. (1964). A primal method for the assignment and transportation problems. *Management Science*, 10, 578–593.
- Barr, R. S., Glover, F., & Klingman, D. (1997). The alternating basis algorithm for assignment problems. *Mathematical Programming*, 13, 1–13.
- Bassani, J. W. M., Rocha, L. S., Luders, M. L., & Bizinotto, W. J. (2002). Micro-process based management of medical equipment maintenance. In *Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint* (Vol. 3, pp. 1942–1943 vol.3). <https://doi.org/10.1109/IEMBS.2002.1053105>
- Bertsekas, D. P. (1981). A New Algorithm for the Assignment Problem. 1981. <https://doi.org/10.1007/BF01584237>
- Burkard, R. E., & Çela, E. (1999). Linear Assignment Problems and Extensions. In D.-Z. Du & P. M. Pardalos (Eds.), *Handbook of Combinatorial Optimization* (pp. 75–149). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4757-3023-4_2
- Burkard, R. E., Dell’Amico, M., & Martello, S. (2009). *Assignment Problems*. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104).

- Centro Nacional de Excelencia Tecnológica en Salud. (2005, February). Guía Tecnológica No. 13: Monitor de Signos Vitales.
- Chegireddy, C. R., & Hamacher, H. W. (1987). Algorithms for finding K-best perfect matchings. *Discrete Applied Mathematics*, 18(2), 155–165. [https://doi.org/10.1016/0166-218X\(87\)90017-5](https://doi.org/10.1016/0166-218X(87)90017-5)
- Cunningham, W. H. (1976). A network simplex method. *Mathematical Programming*, 11, 105–116.
- Cunningham, W. H., & Marsh, A. B. (1978). A primal algorithm for optimum matching. *Mathematical Programming Study*, 8, 50–72.
- Dantzig, G. B. (1963). Linear Programming and Extensions. *Princeton University Press*.
- Dinic, E. A., & Kronrod, M. A. (1969). An algorithm for the solution of the assignment [sic] problem. *Sov. Math. Dokl.*
- Dinitz, Y. (2012, January). *The Assignment Problem*. Presented at the Student Seminar, Ben-Gurion University of Negev. Retrieved from https://www.cs.bgu.ac.il/~dinitz/Course/SS-12/Assignment_Problem-Noa-Yoav/presentation.pdf
- Fukuda, K., & Matsui, T. (1994). Finding all the perfect matchings in bipartite graphs. *Applied Mathematics Letters*, 7(1), 15–18. [https://doi.org/10.1016/0893-9659\(94\)90045-0](https://doi.org/10.1016/0893-9659(94)90045-0)
- Goldberg, A. V., & Kennedy, R. (1995). An efficient cost scaling algorithm for the assignment problem. *Mathematical Programming*, 75, 153–177.
- Goldberg, A. V., Plotkin, S. A., & Vaidya, P. (1993). Sublinear-time parallel algorithms for matching and related problems. *Journal of Algorithms*, 14, 180–213.
- Hoffman, A. J., & Markowitz, H. M. (1963). A note on shortest path, assignment, and transportation problems. *Naval Research Logistics Quarterly*, 10(1), 375–379. <https://doi.org/10.1002/nav.3800100132>
- Hung, M. S., & Rom, W. O. (1980). Solving the assignment problem by relaxation. *Operations Research*, 28(4), 969–982.

- Ji, P., Lee, W. B., & Li, H. (1997). A new algorithm for the assignment problem: An alternative to the Hungarian method. *Computers & Operations Research*, 24(11), 1017–1023. [https://doi.org/10.1016/S0305-0548\(97\)00019-1](https://doi.org/10.1016/S0305-0548(97)00019-1)
- Jonker, R., & Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4), 325–340. <https://doi.org/10.1007/BF02278710>
- Joseph, J., & Madhukumar, S. (2010). A novel approach to Data Driven Preventive Maintenance Scheduling of medical instruments. In *2010 International Conference on Systems in Medicine and Biology (ICSMB)* (pp. 193–197). <https://doi.org/10.1109/ICSMB.2010.5735370>
- Jünger, M., Liebling, T. M., Naddef, D., Nemhauser, G. L., Pulleyblank, W. R., Reinelt, G., ... Wolsey, L. A. (Eds.). (2010). *50 Years of Integer Programming 1958-2008*. Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from <http://link.springer.com/10.1007/978-3-540-68279-0>
- Kennington, J., & Wang, Z. (1992). A Shortest Augmenting Path Algorithm for the Semi-assignment Problem. *Oper. Res.*, 40(1), 178–187. <https://doi.org/10.1287/opre.40.1.178>
- Kuhn, H. W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 83–97. <https://doi.org/10.1002/nav.3800020109>
- Manzini, R., Accorsi, R., Cennerazzo, T., Ferrari, E., & Maranesi, F. (2015). The scheduling of maintenance. A resource-constraints mixed integer linear programming model. *Computers & Industrial Engineering*, 87, 561–568. <https://doi.org/10.1016/j.cie.2015.06.006>
- Medina Durán, R., Malaguti, E., & Duran-Faundez, C. (2017). Techniques for finding a list of solutions with increasing costs for the Semi-Assignment Problem. Presented at the Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON).
- Mills-Tetty, G. A., Stentz, A., & Dias, M. B. (2007). The Dynamic Hungarian Algorithm for the Assignment Problem with Changing Costs. *Robotics Institute*. Retrieved from Magíster en Ingeniería Industrial, Dirección de Postgrado – Universidad de Concepción

<http://repository.cmu.edu/robotics/149>

Monitor multiparámetro VISMO PVM-2701 - arquimed. Retrieved December 22, 2017, from

http://www.arquimed.cl/productos/index.php?id_product=212&controller=product

Murty, K. G. (1968). An Algorithm for Ranking all the Assignments in Order of Increasing Cost.

Operations Research, 16(3), 682–687. <https://doi.org/10.1287/opre.16.3.682>

Orlin, J. B., & Ahuja, R. K. (1992). New scaling algorithms for the assignment and minimum cycle

mean problems. *Mathematical Programming*, 54, 41–56.

Paparrizos, K. (1991). A relaxation column signature method for assignment problems. *European*

Journal of Operational Research, 50, 211–219.

Pedersen, C. R., Relund Nielsen, L., & Andersen, K. A. (2008). An algorithm for ranking assignments

using reoptimization. *Computers & Operations Research*, 35(11), 3714–3726.

<https://doi.org/10.1016/j.cor.2007.04.008>

Robson, J., Yeo, P., Riches, M., Carlisle, T., & Kitto, N. (2005). Risk Management and Biomedical

Devices. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the* (pp. 166–169).

<https://doi.org/10.1109/IEMBS.2005.1616368>

Srinivasan, V., & Thompson, G. L. (1977). Cost operator algorithms for the transportation problem.

Mathematical Programming, 12, 372–391.

Thompson, G. L. (1981). A recursive method for solving assignment problems, in Studies on Graphs

and Discrete Programming. In *Annals of Discrete Mathematics* (Vol. 11, pp. 319–343). North

Holland, Amsterdam: P. Hansen.

Triangle Fractals. (2005). Retrieved December 20, 2017, from

<http://ecademy.agnesscott.edu/~lriddle/ifs/siertri/triangleVariation.htm>

Unidad de Servicios Generales y Mantenimiento. (2007). Plan Mantenimiento y reparación de los

equipos biomédicos, transporte, comunicaciones y de computo de la red de salud Tacna.

Wolsey, L. A. (1998). *Integer Programming* (1 edition). New York: Wiley-Interscience.

World Health Organization. (2012). *Introducción al programa de mantenimiento de equipos médicos*.



ANEXO

Tabla 6. Asignación realizada con el método actualmente utilizado en el hospital. La columna F representa las fallas que debieron ser asignadas, numeradas en orden correlativo, y la columna G representa las empresas (grupos) a los cuales fueron asignadas. Fuente: Elaboración propia.

<i>F</i>	<i>G</i>										
0	13	55	40	110	34	165	17	220	4	275	39
1	13	56	6	111	0	166	17	221	38	276	35
2	13	57	4	112	15	167	16	222	2	277	41
3	13	58	37	113	15	168	16	223	2	278	11
4	33	59	39	114	11	169	28	224	2	279	3
5	13	60	15	115	11	170	28	225	2	280	3
6	13	61	37	116	4	171	28	226	2	281	3
7	13	62	18	117	4	172	28	227	2	282	41
8	20	63	37	118	35	173	0	228	2	283	28
9	16	64	18	119	35	174	6	229	36	284	28
10	20	65	18	120	23	175	7	230	36	285	12
11	20	66	37	121	34	176	7	231	2	286	32
12	23	67	21	122	15	177	16	232	2	287	12
13	23	68	2	123	11	178	16	233	34	288	42
14	23	69	11	124	5	179	34	234	36	289	34
15	23	70	10	125	5	180	34	235	36	290	12
16	32	71	21	126	9	181	38	236	36	291	41
17	32	72	25	127	31	182	38	237	41	292	24
18	32	73	10	128	31	183	38	238	41	293	33
19	32	74	25	129	31	184	38	239	35	294	33
20	20	75	39	130	31	185	38	240	35	295	33
21	20	76	11	131	31	186	38	241	35	296	6
22	20	77	9	132	31	187	38	242	29	297	6
23	20	78	21	133	31	188	38	243	3	298	6
24	20	79	4	134	7	189	38	244	3	299	0
25	20	80	4	135	30	190	38	245	3	300	4
26	20	81	10	136	18	191	4	246	3	301	22
27	35	82	10	137	5	192	8	247	3	302	26
28	35	83	21	138	11	193	28	248	12	303	4
29	24	84	33	139	8	194	7	249	12	304	2
30	16	85	21	140	7	195	11	250	29	305	2
31	42	86	21	141	7	196	18	251	29	306	3
32	42	87	21	142	7	197	8	252	22	307	2

33	19	88	21	143	4	198	9	253	4	308	2
34	11	89	27	144	8	199	8	254	34	309	3
35	11	90	27	145	42	200	8	255	4	310	15
36	4	91	27	146	17	201	8	256	22	311	15
37	14	92	27	147	17	202	4	257	9	312	5
38	14	93	27	148	17	203	9	258	22	313	5
39	14	94	27	149	30	204	9	259	33	314	5
40	30	95	27	150	4	205	9	260	39	315	26
41	30	96	27	151	0	206	9	261	33	316	26
42	19	97	15	152	0	207	6	262	11	317	7
43	19	98	27	153	4	208	18	263	11	318	15
44	19	99	34	154	0	209	38	264	6	319	1
45	19	100	34	155	0	210	38	265	6	320	1
46	1	101	34	156	0	211	3	266	42	321	1
47	37	102	34	157	0	212	3	267	25	322	1
48	37	103	15	158	0	213	3	268	26	323	1
49	37	104	24	159	0	214	3	269	11	324	1
50	30	105	24	160	4	215	4	270	11	325	42
51	37	106	24	161	11	216	1	271	11	326	21
52	40	107	24	162	12	217	19	272	16	327	3
53	40	108	34	163	17	218	0	273	16	328	3
54	40	109	9	164	17	219	23	274	39	-	-

Tabla 7. Primeras 10 mejores soluciones utilizando el algoritmo presentado. Los valores mostrados en la columna F corresponden a las fallas (elementos i a asignar). Para cada solución (S 1 a S 10) se muestra la empresa (grupo) al cual cada una fue asignada. Fuente: Elaboración propia.

<i>F</i>	<i>S 1</i>	<i>S 2</i>	<i>S 3</i>	<i>S 4</i>	<i>S 5</i>	<i>S 6</i>	<i>S 7</i>	<i>S 8</i>	<i>S 9</i>	<i>S 10</i>
<i>0</i>	13	13	13	13	13	13	13	13	13	13
<i>1</i>	18	18	18	18	20	18	18	18	18	20
<i>2</i>	10	10	10	10	10	10	10	10	10	10
<i>3</i>	18	18	18	18	18	18	18	18	18	18
<i>4</i>	33	33	33	33	33	33	33	33	33	33
<i>5</i>	18	18	18	18	18	18	18	18	18	18
<i>6</i>	38	38	38	38	38	38	38	38	38	38
<i>7</i>	18	18	18	18	18	18	18	18	18	18
<i>8</i>	35	35	35	35	35	35	35	35	35	35
<i>9</i>	31	31	31	31	31	31	31	31	31	31
<i>10</i>	35	35	35	35	35	35	35	35	35	35
<i>11</i>	27	27	27	27	27	27	27	27	27	27
<i>12</i>	42	42	42	42	42	42	42	42	42	42
<i>13</i>	25	25	25	25	25	25	25	25	25	25

14	31	31	31	31	31	31	31	31	31	31
15	31	31	31	31	31	31	31	31	31	31
16	32	32	32	32	32	32	32	32	32	32
17	26	26	26	26	26	26	26	26	26	26
18	41	41	41	41	41	41	41	41	41	41
19	31	0	31	0	0	31	31	31	0	0
20	2	2	2	2	10	2	2	2	2	10
21	40	40	40	40	40	40	40	40	40	40
22	13	13	13	13	18	13	13	13	13	18
23	10	10	10	10	13	10	10	10	10	13
24	38	38	38	38	38	38	38	38	38	38
25	39	39	39	39	39	39	39	39	39	39
26	39	39	39	39	39	39	39	39	39	39
27	38	38	38	38	38	38	38	38	38	38
28	13	13	13	13	13	13	13	13	13	13
29	27	27	27	27	27	27	27	27	27	27
30	34	34	34	34	34	34	34	34	34	34
31	35	35	35	35	35	35	35	35	35	35
32	38	38	38	38	38	38	38	38	38	38
33	28	28	28	28	28	28	28	28	28	28
34	11	11	11	11	11	11	11	11	11	11
35	11	11	11	11	11	11	11	11	11	11
36	0	0	0	0	0	0	0	0	0	0
37	41	41	41	41	41	41	41	41	41	41
38	35	35	35	35	35	35	35	35	35	35
39	23	23	23	23	23	23	23	23	23	23
40	0	0	0	0	0	0	0	0	0	0
41	4	4	4	4	4	4	4	4	4	4
42	19	19	19	19	19	19	19	19	19	19
43	27	27	27	27	27	27	27	27	27	27
44	23	19	23	19	23	23	19	23	19	23
45	20	20	20	20	20	20	2	20	20	20
46	19	19	19	19	19	19	19	19	19	19
47	27	27	27	27	27	27	27	27	27	27
48	20	20	20	20	20	20	20	20	20	20
49	34	34	34	34	34	34	34	34	34	34
50	4	4	4	4	4	4	4	4	4	4
51	20	20	20	20	20	20	20	20	20	20
52	23	23	23	23	23	23	23	23	23	23
53	19	19	19	19	19	19	19	19	19	19
54	39	39	39	39	39	39	39	39	39	39
55	31	31	31	31	31	31	31	31	31	31
56	6	6	6	6	6	6	6	6	6	6
57	9	9	9	9	9	9	9	9	9	9
58	27	27	27	27	27	27	27	27	27	27

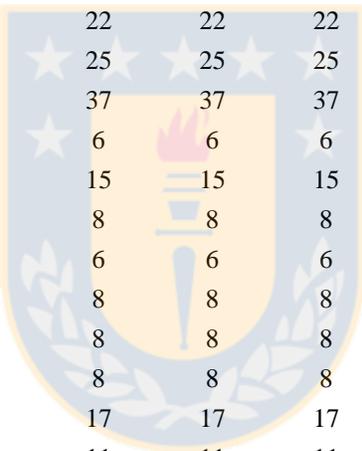
59	35	35	35	35	35	35	35	35	35	35
60	26	26	26	26	26	26	26	26	26	26
61	21	21	21	21	21	21	21	21	21	21
62	35	35	35	35	35	35	35	35	35	35
63	2	2	2	2	2	2	2	2	2	2
64	10	10	10	10	10	10	10	10	10	10
65	35	35	35	35	35	35	35	35	35	35
66	34	34	34	34	34	34	34	34	34	34
67	21	21	21	21	21	21	21	21	21	21
68	35	35	35	35	35	35	35	35	35	35
69	11	11	11	11	11	11	11	11	11	11
70	21	21	21	21	21	21	21	21	21	21
71	42	42	42	42	42	42	42	42	42	42
72	11	11	11	11	11	11	11	11	11	11
73	39	39	39	39	39	39	39	39	39	39
74	40	40	40	40	40	40	40	40	40	40
75	24	24	24	24	24	24	24	24	24	24
76	33	33	33	33	33	33	33	33	33	33
77	11	11	11	11	11	11	11	11	11	11
78	19	19	19	19	19	19	19	19	19	19
79	30	30	30	30	30	30	30	30	30	30
80	34	34	34	34	34	34	34	34	34	34
81	42	42	42	42	42	42	42	42	42	42
82	38	38	38	38	38	38	38	38	38	38
83	27	27	27	27	27	27	27	27	27	27
84	30	30	30	30	30	30	30	30	30	30
85	19	20	19	20	19	19	20	19	20	19
86	42	42	42	42	42	42	42	42	42	42
87	34	34	34	34	34	34	34	34	34	34
88	34	34	34	34	34	34	34	34	34	34
89	2	23	2	23	2	2	23	2	23	2
90	28	28	28	28	28	28	28	28	28	28
91	2	2	2	2	2	2	2	2	2	2
92	34	34	34	34	34	34	34	34	34	34
93	20	20	20	20	20	20	20	20	20	20
94	34	34	34	34	34	34	34	34	34	34
95	23	23	23	23	23	23	23	23	23	23
96	19	19	19	19	19	19	19	19	19	19
97	8	8	8	8	8	8	8	8	8	8
98	21	21	21	21	21	21	21	21	21	21
99	37	37	37	37	37	37	37	37	37	37
100	2	2	2	2	2	2	2	2	2	2
101	23	23	23	23	23	23	23	23	23	23
102	28	28	20	20	28	28	28	28	28	20
103	6	6	6	6	6	6	6	6	6	6

104	39	39	39	39	39	39	39	39	39	39
105	28	28	28	28	28	28	28	28	28	28
106	15	15	15	15	15	15	15	15	15	15
107	29	29	29	29	29	36	29	29	36	29
108	2	2	2	2	2	2	2	2	2	2
109	5	5	5	5	5	5	5	5	5	5
110	21	21	21	21	21	21	21	21	21	21
111	4	4	4	4	4	4	4	4	4	4
112	41	41	41	41	41	41	41	41	41	41
113	6	6	6	6	6	6	6	6	6	6
114	11	11	11	11	11	11	11	11	11	11
115	38	38	38	38	38	38	38	38	38	38
116	4	4	4	4	4	4	4	4	4	4
117	3	3	3	3	3	3	3	3	3	3
118	37	37	37	37	37	37	37	37	37	37
119	41	41	41	41	41	41	41	41	41	41
120	30	30	30	30	30	30	30	30	30	30
121	0	2	0	2	2	0	0	0	2	2
122	9	9	9	9	9	9	9	9	9	9
123	5	5	5	5	5	5	5	5	5	5
124	5	5	5	5	5	5	5	5	5	5
125	11	11	11	11	11	11	11	11	11	11
126	41	41	41	41	41	41	41	41	41	41
127	20	20	16	16	20	20	20	20	20	16
128	16	16	16	16	16	16	16	16	16	16
129	3	3	3	3	3	3	3	3	3	3
130	2	2	2	2	2	2	2	2	2	2
131	40	40	40	40	40	40	40	40	40	40
132	27	27	27	27	27	27	27	27	27	27
133	34	34	34	34	34	34	34	34	34	34
134	9	9	9	9	9	9	9	9	9	9
135	24	24	24	24	24	24	24	24	24	24
136	1	1	1	1	1	1	1	1	1	1
137	11	11	11	11	11	11	11	11	11	11
138	38	38	38	38	38	38	38	38	38	38
139	11	11	11	11	11	11	11	11	11	11
140	2	2	2	2	2	2	2	2	2	2
141	16	16	16	16	16	16	16	16	16	16
142	3	3	3	3	3	3	3	3	3	3
143	4	4	4	4	4	4	4	4	4	4
144	5	5	5	5	5	5	5	5	5	5
145	20	20	20	20	20	20	20	20	20	20
146	21	21	21	21	21	21	21	21	21	21
147	21	21	21	21	21	21	21	21	21	21
148	3	3	3	3	3	3	3	3	3	3

149	0	0	0	0	0	0	0	0	0	0
150	4	4	4	4	4	4	4	4	4	4
151	27	27	27	27	27	27	27	27	27	27
152	0	0	0	0	0	0	0	0	0	0
153	30	30	30	30	30	30	30	30	30	30
154	0	0	0	0	0	0	0	0	0	0
155	4	4	4	4	4	4	4	4	4	4
156	4	4	4	4	4	4	4	4	4	4
157	0	0	0	0	0	0	0	0	0	0
158	0	0	0	0	0	0	0	0	0	0
159	4	4	4	4	4	4	4	4	4	4
160	11	11	11	11	11	11	11	11	11	11
161	11	11	11	11	11	11	11	11	11	11
162	8	8	8	8	8	8	8	8	8	8
163	1	1	1	1	1	1	1	1	1	1
164	38	38	38	38	38	38	38	38	38	38
165	38	38	38	38	38	38	38	38	38	38
166	16	16	16	16	16	16	16	16	16	16
167	20	20	20	20	20	20	20	20	20	20
168	34	34	34	34	34	34	34	34	34	34
169	38	38	38	38	38	38	38	38	38	38
170	20	20	20	20	20	20	20	20	20	20
171	1	1	1	1	1	1	1	1	1	1
172	1	1	1	1	1	1	1	1	1	1
173	2	2	2	2	2	2	2	2	2	2
174	5	5	5	5	5	5	5	5	5	5
175	31	31	31	31	31	31	31	31	31	31
176	3	3	3	3	3	3	3	3	3	3
177	3	3	3	3	3	3	3	3	3	3
178	7	7	7	7	7	7	7	7	7	7
179	34	34	34	34	34	34	34	34	34	34
180	3	3	3	3	3	3	3	3	3	3
181	38	38	38	38	38	38	38	38	38	38
182	16	16	16	16	16	16	16	16	16	16
183	2	2	2	2	2	2	2	2	2	2
184	7	7	7	7	7	7	7	7	7	7
185	7	7	7	7	7	7	7	7	7	7
186	42	42	42	42	42	42	42	42	42	42
187	3	3	3	3	3	3	3	3	3	3
188	40	40	40	40	40	40	40	40	40	40
189	20	31	20	31	31	20	20	20	31	31
190	3	3	3	3	3	3	3	3	3	3
191	9	9	9	9	9	9	9	9	9	9
192	5	5	5	5	5	5	5	5	5	5
193	28	28	28	28	28	28	28	28	28	28

194	11	11	11	11	11	11	11	11	11	11
195	11	11	11	11	11	11	11	11	11	11
196	11	11	11	11	11	11	11	11	11	11
197	4	4	4	4	4	4	4	4	4	4
198	4	4	4	4	4	4	4	4	4	4
199	4	4	4	4	4	4	4	4	4	4
200	9	9	9	9	9	9	9	9	9	9
201	33	33	33	33	33	33	33	33	33	33
202	9	9	9	9	9	9	9	9	9	9
203	23	23	23	23	23	23	23	23	23	23
204	9	9	9	9	9	9	9	9	9	9
205	33	33	33	33	33	33	33	33	33	33
206	3	3	3	3	3	3	3	3	3	3
207	32	32	32	32	32	32	32	32	32	32
208	13	13	13	13	13	13	13	13	13	13
209	17	17	17	17	17	17	17	17	17	17
210	20	20	20	20	20	20	20	20	20	20
211	17	17	17	17	17	17	17	17	17	17
212	13	13	13	13	13	13	13	13	13	13
213	24	24	24	24	24	24	24	24	24	24
214	17	17	17	17	17	17	17	17	17	17
215	8	8	8	8	8	8	8	8	8	8
216	17	17	17	17	17	17	17	17	17	17
217	32	32	32	32	32	32	32	32	32	32
218	1	1	1	1	1	1	1	1	1	1
219	1	1	1	1	1	1	1	1	1	1
220	4	4	4	4	4	4	4	4	4	4
221	16	16	16	16	16	16	16	16	16	16
222	17	17	17	17	17	17	17	17	17	17
223	7	7	7	7	7	7	7	7	7	7
224	7	7	7	7	7	7	7	7	7	7
225	7	7	7	7	7	7	7	7	7	7
226	16	16	16	16	16	16	16	16	16	16
227	7	7	7	7	7	7	7	7	7	7
228	31	31	31	31	31	31	31	31	31	31
229	15	15	15	15	15	15	15	15	15	15
230	12	12	12	12	12	12	12	12	12	12
231	2	2	2	2	2	2	2	2	2	2
232	3	3	3	3	3	3	3	3	3	3
233	4	4	4	4	4	4	4	4	4	4
234	12	12	12	12	12	12	12	12	12	12
235	36	36	36	36	36	29	36	28	29	36
236	14	14	14	14	14	14	14	14	14	14
237	12	12	12	12	12	12	12	12	12	12
238	12	12	12	12	12	12	12	12	12	12

239	36	36	36	36	36	36	36	36	36	36
240	12	12	12	12	12	12	12	12	12	12
241	24	24	24	24	24	24	24	24	24	24
242	12	12	12	12	12	12	12	12	12	12
243	24	24	24	24	24	24	24	24	24	24
244	22	22	22	22	22	22	22	22	22	22
245	14	14	14	14	14	14	14	14	14	14
246	28	28	28	28	28	28	28	28	28	28
247	15	15	15	15	15	15	15	15	15	15
248	3	3	3	3	3	3	3	3	3	3
249	3	3	3	3	3	3	3	3	3	3
250	15	15	15	15	15	15	15	15	15	15
251	37	37	37	37	37	37	37	37	37	37
252	22	22	22	22	22	22	22	22	22	22
253	21	21	21	21	21	21	21	21	21	21
254	0	0	0	0	0	0	0	0	0	0
255	21	21	21	21	21	21	21	21	21	21
256	22	22	22	22	22	22	22	22	22	22
257	25	25	25	25	25	25	25	25	25	25
258	37	37	37	37	37	37	37	37	37	37
259	6	6	6	6	6	6	6	6	6	6
260	15	15	15	15	15	15	15	15	15	15
261	8	8	8	8	8	8	8	8	8	8
262	6	6	6	6	6	6	6	6	6	6
263	8	8	8	8	8	8	8	8	8	8
264	8	8	8	8	8	8	8	8	8	8
265	8	8	8	8	8	8	8	8	8	8
266	17	17	17	17	17	17	17	17	17	17
267	11	11	11	11	11	11	11	11	11	11
268	16	16	16	16	16	16	16	16	16	16
269	25	25	25	25	25	25	25	25	25	25
270	26	26	26	26	26	26	26	26	26	26
271	9	9	9	9	9	9	9	9	9	9
272	11	11	11	11	11	11	11	11	11	11
273	9	9	9	9	9	9	9	9	9	9
274	3	3	3	3	3	3	3	3	3	3
275	15	15	15	15	15	15	15	15	15	15
276	14	14	14	14	14	14	14	14	14	14
277	36	36	36	36	36	36	36	36	36	36
278	6	6	6	6	6	6	6	6	6	6
279	2	2	2	2	2	2	2	2	2	2
280	13	13	13	13	13	13	13	13	13	13
281	17	17	17	17	17	17	17	17	17	17
282	37	37	37	37	37	37	37	37	37	37
283	29	29	29	29	29	29	29	29	29	29



284	15	15	15	15	15	15	15	15	15	15
285	28	28	28	28	28	28	28	36	28	28
286	7	7	7	7	7	7	7	7	7	7
287	3	3	3	3	3	3	3	3	3	3
288	1	1	1	1	1	1	1	1	1	1
289	30	30	30	30	30	30	30	30	30	30
290	3	3	3	3	3	3	3	3	3	3
291	24	24	24	24	24	24	24	24	24	24
292	29	29	29	29	29	29	29	29	29	29
293	33	33	33	33	33	33	33	33	33	33
294	18	18	18	18	18	18	18	18	18	18
295	33	33	33	33	33	33	33	6	33	33
296	6	6	6	6	6	6	6	33	6	6
297	6	6	6	6	6	6	6	6	6	6
298	33	33	33	33	33	33	33	33	33	33
299	10	10	10	10	10	10	10	10	10	10
300	0	0	0	0	0	0	0	0	0	0
301	22	22	22	22	22	22	22	22	22	22
302	15	15	15	15	15	15	15	15	15	15
303	4	4	4	4	4	4	4	4	4	4
304	4	4	4	4	4	4	4	4	4	4
305	4	4	4	4	4	4	4	4	4	4
306	36	36	36	36	36	36	36	36	36	36
307	27	27	27	27	27	27	27	27	27	27
308	4	4	4	4	4	4	4	4	4	4
309	37	37	37	37	37	37	37	37	37	37
310	15	15	15	15	15	15	15	15	15	15
311	36	36	36	36	36	36	36	36	36	36
312	13	13	13	13	13	13	13	13	13	13
313	32	32	32	32	32	32	32	32	32	32
314	0	0	0	0	0	0	0	0	0	0
315	37	37	37	37	37	37	37	37	37	37
316	32	32	32	32	32	32	32	32	32	32
317	26	26	26	26	26	26	26	26	26	26
318	37	37	37	37	37	37	37	37	37	37
319	16	16	28	28	16	16	16	16	16	28
320	38	38	38	38	38	38	38	38	38	38
321	1	1	1	1	1	1	1	1	1	1
322	38	38	38	38	38	38	38	38	38	38
323	42	42	42	42	42	42	42	42	42	42
324	34	34	34	34	34	34	34	34	34	34
325	2	2	2	2	2	2	2	2	2	2
326	0	0	0	0	0	0	0	0	0	0
327	2	2	2	2	2	2	2	2	2	2
328	18	18	18	18	18	18	18	18	18	18



Figura 12. Costo por empresa en asignación utilizando el método actualmente empleado. Fuente: Elaboración propia.



Figura 13. Costo por empresa en la mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 14. Costo por empresa en la 2ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 15. Costo por empresa en la 3ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.

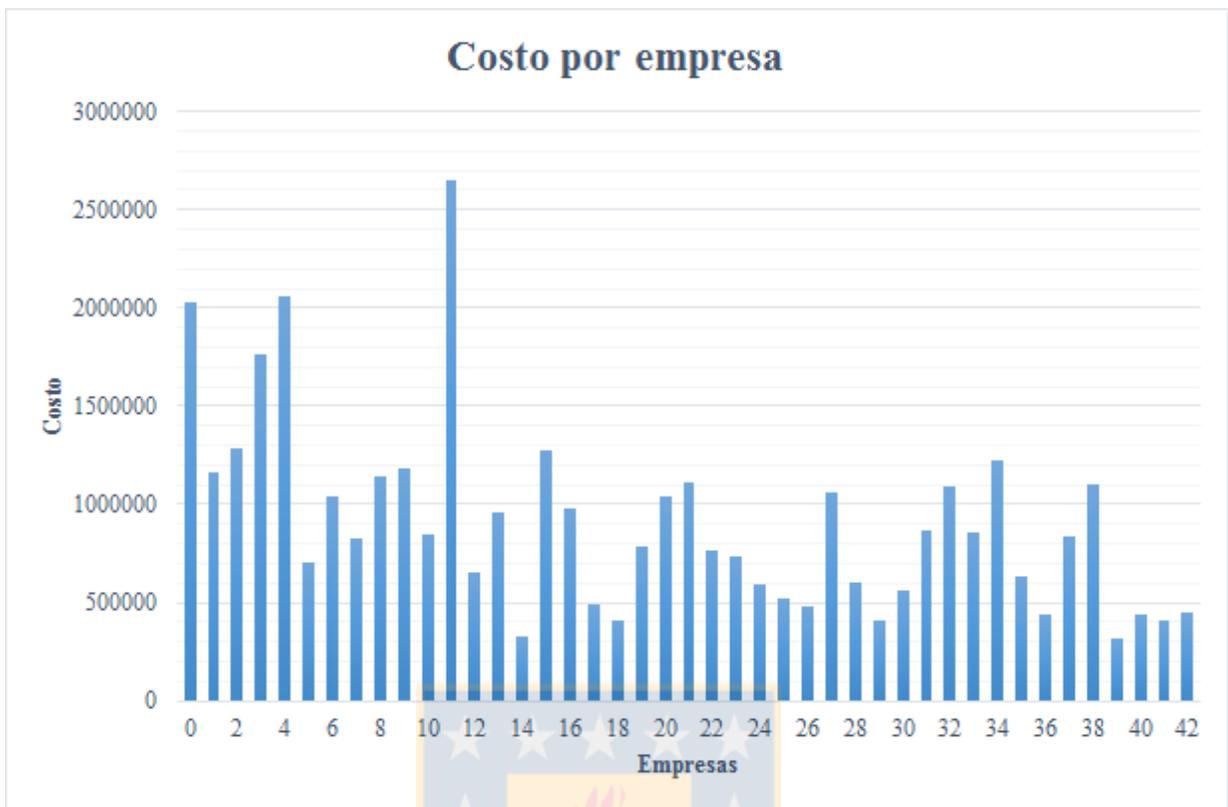


Figura 16. Costo por empresa en la 4ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 17. Costo por empresa en la 5ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 18. Costo por empresa en la 6ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 19. Costo por empresa en la 7ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 20. Costo por empresa en la 8ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.



Figura 21. Costo por empresa en la 9ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.

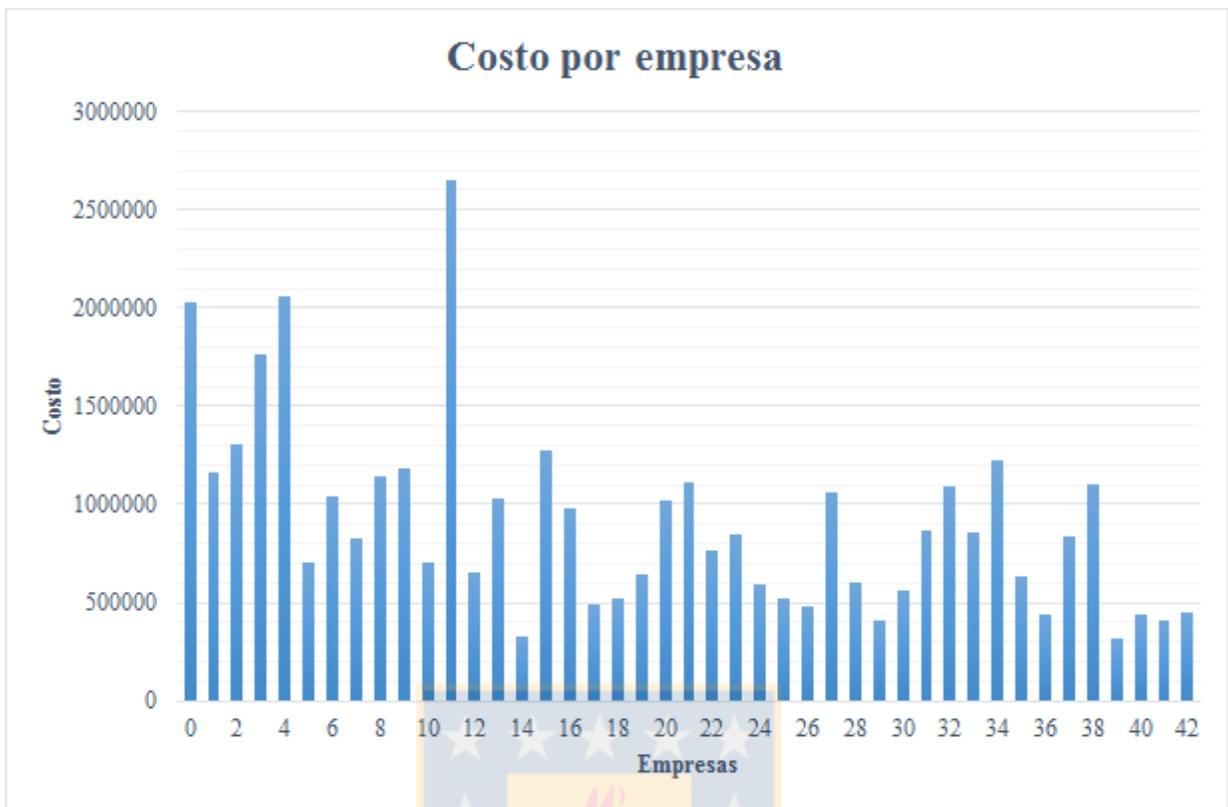


Figura 22. Costo por empresa en la 10ª mejor solución entregada por la metodología propuesta. Fuente: Elaboración propia.

