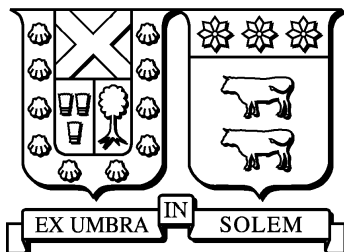


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA

DEPARTAMENTO DE INFORMÁTICA

SANTIAGO – CHILE



“UN ACERCAMIENTO HEURÍSTICO
MULTI-OBJETIVO PARA EL PROBLEMA DE
DISTRIBUCIÓN DE MATERIALES PELIGROSOS
CON MEZCLA”

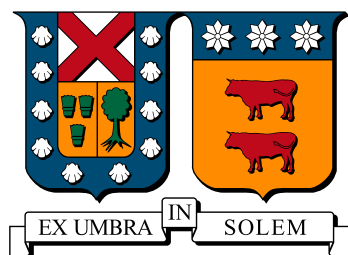
ALEXANDER MAXIMILIANO RUZ ROJAS

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO

PROFESOR GUÍA: ELIZABETH MONTERO U.

AGOSTO 2021

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



**“UN ACERCAMIENTO HEURÍSTICO
MULTI-OBJETIVO PARA EL PROBLEMA DE
DISTRIBUCIÓN DE MATERIALES
PELIGROSOS CON MEZCLA”**

ALEXANDER MAXIMILIANO RUZ ROJAS

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INFORMÁTICO**

PROFESOR GUÍA: ELIZABETH MONTERO U.

PROFESOR CORREFERENTE: JOSÉ LUIS MARTÍ L.

AGOSTO 2021

MATERIAL DE REFERENCIA, SU USO NO INVOLUCRA RESPONSABILIDAD DEL AUTOR O DE LA INSTITUCIÓN

Agradecimientos

A las primeras personas que quiero agradecer son a mis padres Alejandro y Myriam, mis hermanas Allison y Katherine, mi hermano Felipe, mi sobrino Máximo y a mi cuñado Rodrigo. Con la presencia de ellos en mi vida, este proceso no fue tan agobiante.

Gracias a mis compañeros de universidad con los que compartí bastantes años de forma presencial y online (modo pandemia). Fueron momentos de entretenimiento, estudios y risas. Espero siga así!

Gracias a mis amigos que conozco hace años y con los que aún sigo compartiendo y teniendo buenos e inolvidables momentos.

Y también agradecer a la profesora Elizabeth Montero, muchas gracias por la ayuda, por el ánimo y por su apoyo en este gran y difícil proceso!

Resumen

En el presente trabajo de memoria, se aborda el problema *distribución de materiales peligrosos con mezcla*. En este se busca asignar rutas a camiones, para que distribuyan los materiales peligrosos a los clientes. Se considera mezcla, porque durante el transporte el camión puede llevar más de un tipo de material peligroso, siempre y cuando estos sean compatibles entre sí. El problema es un problema multi-objetivo ya que, por un lado se busca minimizar los costos del transporte asociados al proceso de distribución, es decir, abaratar costos. Por otro lado, dado el transporte de materiales que pueden ser riesgosos para la salud de la población se busca minimizar su exposición a posibles reacciones de los materiales peligrosos, debidas a accidentes durante que se puedan dar durante el proceso.

Para la resolución del problema, se propone utilizar el algoritmo Greedy para la construcción inicial de la soluciones factibles al problema. Además, para proceso de búsqueda local se propone un algoritmo basado en Tabu Search que incorpora dos movimientos. Estos operadores son 2-OPT e Insert. El movimiento 2-opt se encarga de buscar secuencias de visitas diferentes dado un conjunto de clientes mientras que el operador insert es capaz de modificar el conjunto de clientes visitados por cada ruta. Además, se considera el uso de la lista tabú para impedir ciclos durante el proceso de búsqueda que lleven a re-visitarse zonas del espacio de búsqueda ya analizadas.

El algoritmo es aplicado a un caso real de la ciudad de Santiago de Chile, la cual es dividida en siete zonas, cada una funcionando de forma independiente. Las instancias solo comparten el depósito desde donde se distribuyen los materiales y salen los camiones de reparto. La data provista considera la información de la red de Santiago de Chile que considera todas las intersecciones como nodos de un gran grafo que es recorrido usando un algoritmo Dijkstra

para determinar los mejores caminos entre cada par de nodos y bajo ciertas condiciones de importancia relativa de los dos objetivos.

Los resultados de las pruebas muestran que el algoritmo es capaz de conseguir un amplio conjunto de soluciones no dominadas para la mayoría de los casos evaluados. Por otro lado, al comparar los resultados con los resultados de un modelo resuelto a través del solver CPLEX 12.8 se observa que los resultados del método heurístico presentan un peor nivel de convergencia que se ve contrarrestado por su capacidad de encontrar soluciones de calidad en gran parte de los frentes.

Abstract

This work studied the problem of distribution of hazardous materials with blending. The problem consists of assigning routes to trucks in order to distribute hazardous materials to customers. It is considered a blending problem, because during transportation trucks can carry more than one type of dangerous material, if they are compatible with each other. On the other hand, the problem is a multi-objective problem because it seeks to minimize the transportation costs associated with the distribution process, that is, reducing costs. On the other hand, given the transportation of materials that may be risky for the health of the population, it is sought to minimize their exposure to possible reactions of hazardous materials due to accidents that may occur during the process.

To solve the problem, here we proposed to use the Greedy algorithm for the initial construction of feasible solutions to the problem. In addition, an algorithm based on Tabu Search that incorporates two movements is proposed for the local search process. These operators are 2-OPT and Insert. The 2-opt movement is in charge of searching for different visit sequences given a set of clients, while the insert operator is capable of modifying the set of clients visited by each route. In addition, the use of the tabu list is considered to prevent cycles during the search process that lead to re-visit areas of the search space already analyzed.

The algorithm is applied in a real case of the city of Santiago, Chile, which is divided into seven zones, each one operating independently. The instances only share the warehouse from where the materials are distributed and the delivery trucks leave. The data provided considers the information from Santiago, network that considers all intersections as nodes of a large graph that is traversed using a Dijkstra algorithm to determine the best paths between each pair of nodes and under certain conditions of relative importance of the two objectives.

The test results show that the algorithm can achieve a wide set of non-dominated solutions for most of the cases evaluated. On the other hand, when comparing the results with the results of a model resolved through the CPLEX 12.8 solver, it is observed that the results of the heuristic method show a poor level of convergence due to its ability to find quality solutions to a large extent of the Pareto fronts.

Índice de Contenidos

Agradecimientos	III
Resumen	IV
Abstract	VI
Índice de Contenidos	VIII
Lista de Tablas	XI
Lista de Figuras	XII
Introducción	1
1. Definición del Problema	3
1.1. Objetivo General	3
1.2. Objetivos Específicos	3
1.3. Distribución de materiales peligrosos	4
1.3.1. Problemas Multi-Objetivo	10
1.4. Modelo Matemático	11
1.4.1. Parámetros	11
1.4.2. Variables	13

1.4.3.	Restricciones	13
1.4.4.	Naturaleza de las variables	15
1.4.5.	Objetivos	15
1.5.	Resumen	16
2.	Estado del Arte	18
2.1.	Problema de enrutamiento de vehículos	18
2.2.	Enrutamiento de vehículos multi-objetivo	21
2.3.	Transporte de materiales peligrosos	26
2.4.	Resumen	31
3.	Propuesta	33
3.1.	Representación	34
3.2.	Función evaluación	36
3.3.	Estructura general	37
3.3.1.	Algoritmo Greedy	38
3.3.2.	Operadores de búsqueda	41
3.3.3.	Insert	42
3.3.4.	2-opt	44
3.3.5.	Lista Tabú	46
3.3.6.	Actualización archivo de soluciones no dominadas	48
3.4.	Resumen	50
4.	Experimentos y resultados	51
4.1.	Objetivos	51
4.1.1.	Instancias de Prueba	52
4.1.2.	Indicadores de calidad	54

4.1.3. Sintonización de parámetros	56
4.2. Resultados	58
4.2.1. Zona 1	59
4.2.2. Zona 2	61
4.2.3. Zona 3	63
4.2.4. Zona 4	66
4.2.5. Zona 5	69
4.2.6. Zona 6	70
4.2.7. Zona 7	72
4.3. Resumen	75
5. Conclusiones	77
Conclusiones	77
Bibliografía	82

Índice de tablas

1.1. Compatibilidad y resultado de la combinación de materiales peligrosos. . .	5
1.2. Clasificación del material peligroso de acuerdo a las leyes chilenas.	8
4.1. Características de instancias de pruebas obtenidas de [9]	53
4.2. Material producido por los clientes en las instancias de prueba obtenidas de [9]	53
4.3. Puntos de referencia para la evaluación del hipervolumen en cada zona. . .	56
4.4. Parámetros para el proceso de sintonización	57
4.5. Configuración de parámetros sintonizada por instancia y en conjunto	58
4.6. Valores de hipervolumen de ambas configuraciones para la zona 1.	60
4.7. Valores de hipervolumen de ambas configuraciones para la zona 2.	63
4.8. Valores de hipervolumen de ambas configuraciones para la zona 3.	65
4.9. Valores de hipervolumen de ambas configuraciones para la zona 4.	67
4.10. Valores de hipervolumen de ambas configuraciones para la zona 5.	70
4.11. Valores de hipervolumen de ambas configuraciones para la zona 6.	71
4.12. Valores de hipervolumen de ambas configuraciones para la zona 7.	74

Índice de figuras

1.1. Ejemplo de una solución del problema sin considerar materiales peligrosos.	6
1.2. Ejemplo de una solución del problema de este trabajo de memoria.	9
1.3. Frente de Pareto	12
3.1. Representación de la solución.	34
3.2. Representación de la solución de la zona 3.	35
3.3. Representación de materiales de la solución de la zona 3.	36
3.4. Ejemplo uso del movimiento insert.	43
3.5. Ejemplo uso movimiento 2-OPT.	46
3.6. Representación del almacenamiento de la lista tabú.	47
4.1. Hipervolumen para un problema de minimización de dos objetivos.	55
4.2. Frente de Pareto heurístico para ambas configuraciones: zona1 (en azul) y ALL (en rojo) para la zona 1.	60
4.3. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en rojo) para la zona 1.	61
4.4. Frente de Pareto heurístico para ambas configuraciones, zona2 (en azul) y ALL (en rojo) para la zona 2.	62

4.5. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 2.	64
4.6. Frente de Pareto heurístico para ambas configuraciones, zona3 (en azul) y ALL (en rojo) para la zona 3.	65
4.7. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 3.	66
4.8. Frente de Pareto heurístico para ambas configuraciones, zona4 (en azul) y ALL (en rojo) para la zona 4.	67
4.9. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 4.	68
4.10. Frente de Pareto heurístico para ambas configuraciones, zona5 (en azul) y ALL (en rojo) para la zona 5.	69
4.11. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 5.	71
4.12. Frente de Pareto heurístico para ambas configuraciones, zona6 (en azul) y ALL (en rojo) para la zona 6.	72
4.13. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 6.	73
4.14. Frente de Pareto heurístico para ambas configuraciones, zona6 (en azul) y ALL (en rojo) para la zona 7.	74
4.15. Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 7.	75

Introducción

En el presente trabajo de memoria, se propone un algoritmo heurístico multi-objetivo para resolver el problema de distribución de materiales peligrosos de la ciudad de Santiago de Chile. En este problema, la ciudad de Santiago está dividida en siete zonas. Cada una de ellas considera una cantidad de camiones y también una cantidad de clientes, a los cuales se les debe distribuir una carga de material peligroso. Para todas las zonas existen los mismos tipos de materiales, pero cada cliente determina qué tipo de material necesita. Todos los camiones comienzan el proceso de distribución en el depósito y deben distribuir los materiales a sus zonas respectivas. Una vez finalizado este proceso, los camiones deben volver al depósito.

Debido a que el transporte es de materiales peligrosos, el diseño del algoritmo utilizado para resolver el problema, debe considerar dos objetivos. Uno de ellos trata sobre el costo del transporte, que corresponde al traslado de los materiales entre los clientes. Por lo tanto lo que se va a buscar con este objetivo es minimizarlo, para abaratar costos. El segundo objetivo trata sobre la población expuesta a posibles accidentes agravados por el transporte de dichos materiales. Debido a que los materiales peligrosos pueden ser perjudiciales para la salud de las personas, se va a buscar trasladar los materiales por rutas donde se exponga a la menor cantidad de personas y además que el trayecto sea lo más seguro posible. Cabe destacar que cada tipo de material peligroso, tiene un nivel de riesgo diferente y, además, los materiales transportados en un camión particular, deben ser compatibles entre sí, es decir, no reaccionar al llevarlos en un mismo camión. Al igual que el primer objetivo, se va a buscar minimizar la población expuesta al transporte de los materiales peligrosos. Si se busca minimizar ambos objetivos, en gran parte de los casos puede ocurrir que las rutas para un objetivo sean muy diferentes a las que optimizan el otro objetivo.

Para la resolución de este problema se propone un algoritmo de búsqueda heurística basado en Greedy y Tabu Search, sumado a dos operadores de búsqueda local. Se utiliza el algoritmo Greedy para construir la solución inicial del problema, utilizando solo el objetivo de costos. Esto se hace para simplificar el proceso de construcción de esta solución. Luego, el algoritmo Tabu Search es aplicado utilizando los operadores 2-OPT e Insert. El primero de ellos invierte una sección de la ruta de un camión, de modo de cambiar la secuencia de visitas definida por dicha ruta. El segundo movimiento, remueve un cliente de una ruta de un camión y lo inserta en alguna posición de la ruta de otro camión, siempre y cuando se sigan cumpliendo las restricciones de compatibilidad y capacidad de camiones del problema. Una lista tabu es implementada para restringir la visitada de espacios ya explorados durante la búsqueda.

El acercamiento se evalúa utilizando las siete instancias de prueba del problema de distribución de materiales peligroso de Santiago de Chile. Dicha data usa la red completa de Santiago de Chile en la cual se requiere realizar búsqueda de caminos de costo mínimo para determinar caminos entre clientes para, posteriormente, construir las rutas de visitas del problema. Los resultados obtenidos se comparan con los obtenidos por un acercamiento de búsqueda completa.

Este trabajo de memoria, tendrá la siguiente estructura. En el capítulo 1, se da a conocer la definición del problema, junto al modelo matemático de éste. En el capítulo 2, se presenta el estado del arte y consiste en una revisión de resoluciones similares de problemas semejantes. Luego en el capítulo 3, se presenta la propuesta para resolver el presente problema, es decir, el algoritmo utilizado. En el capítulo 4, se presentan las instancias en que se ejecutará el algoritmo y los resultados de éste. Y finalmente, en el capítulo 5 se presenta la conclusión y futuros trabajos.

Capítulo 1

Definición del Problema

En este capítulo se presenta la definición del problema distribución de materiales peligrosos bi-objetivo. El capítulo comienza presentando el objetivo general y los objetivos específicos. A continuación se presenta una descripción del problema a resolver, el cual está basado en un caso real de la ciudad de Santiago de Chile. Se detallan las principales características del problema. Además se presenta una descripción de problemas multi-objetivo. Por último se presenta el problema resuelto.

1.1. Objetivo General

Desarrollar un algoritmo eficiente para resolver el problema de distribución de materiales peligrosos bi-objetivo basado en un caso de estudio.

1.2. Objetivos Específicos

Identificar el problema a trabajar y sus variantes a lo largo del tiempo.

Investigar sobre soluciones ya definidas para el problema a trabajar de esta propuesta.

Diseñar una técnica de búsqueda eficiente con el fin de obtener resultados de buena calidad, en tiempos acotados.

Evaluar los resultados obtenidos utilizando el algoritmo propuesto.

Analizar y comparar los resultados con artículos que hayan resuelto el mismo caso de estudio.

1.3. Distribución de materiales peligrosos

Una gran cantidad de empresas y compañías de transporte, trabajan con materiales nocivos para la salud de las personas y el medio ambiente. Por lo que deben tomar los resguardos necesarios para que el transporte de estos materiales sea lo más seguro posible y así disminuir la posibilidad de que ocurra un accidente durante el proceso. Posibles accidentes como fugas de gas o químicas, explosiones, incendios, radiación de los materiales, entre otros. Es por esto que investigadores han estudiado el problema y han propuesto posibles modelos matemáticos con el fin de encontrar rutas que disminuyan los riesgos de accidentes. Tomar decisiones para evitar el transporte en rutas inseguras, como curvas en carretera, pavimento en mal estado o minimizando la población expuesta a un posible accidente de un material peligroso disminuye el riesgo.

El problema a desarrollar en este trabajo de memoria, es basado en un caso real de la ciudad de Santiago de Chile. En donde el objetivo principal es distribuir distintos materiales peligrosos, por rutas seguras y rápidas, mediante una flota de vehículos que transportarán esta carga a los distintos clientes. Para este problema en particular se consideran tipos materiales con distintos niveles de riesgo. Este nivel de riesgo es interpretado como la cantidad de población que ponen en riesgo en caso de un accidente durante el transporte de la carga. Los vehículos deben cumplir con tres condiciones al distribuir la carga. La primera restricción y más importante es que los materiales deben ser compatibles entre sí, es decir, que el vehículo solo puede transportar materiales en conjunto que no generen un riesgo extra si es que se llevan en la misma carga. Esto se relaciona con el transporte de carga conocido en la literatura como transporte con mezclas. Tal como se observa en la tabla 1.1, en donde se muestra los

Tipo de material	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>
<i>A</i>	A	-	C	D	-
<i>B</i>	-	B	C	D	E
<i>C</i>	C	C	C	-	E
<i>D</i>	D	D	-	D	E
<i>E</i>	-	E	E	E	E

Tabla 1.1: Compatibilidad y resultado de la combinación de materiales peligrosos.

cinco tipos de materiales peligrosos A, B, C, D, y E y cuales son compatibles entre sí y en caso de que sí lo sean, cual es el material a considerar para el cálculo del riesgo. Por ejemplo, para el caso del material A, solo es compatible con los materiales C y D. Si se toma la decisión de llevar el material A con el material C, por dominancia, en el cálculo se considera el riesgo del material C. Si se toma la otra decisión, que es llevar el material A junto al material D, por la misma razón de dominancia, se considera el riesgo del material D.

La segunda restricción es que la cantidad de material a recolectar no se puede dividir, si a un vehículo transporta la carga de un material exigido por un cliente, el mismo vehículo debe distribuir el total de la carga para el cliente y no puede ser distribuida entre dos o más vehículos.

La tercera restricción es que los vehículos tienen una capacidad que no debe ser sobrepasada.

Como se mencionó, existen incompatibilidades entre los materiales peligrosos y para este caso deben regirse bajo la legislación chilena. En la tabla 1.1 se muestra la compatibilidad y el resultado de la mezcla entre los materiales. En este caso, si es que un vehículo transporta material tipo A, puede también llevar los materiales C y D por separado, imposible que pueda llevar ambos juntos porque según la tabla 1.1 C es incompatible con D. Si es que lleva también el material C, sus rutas serán consideradas con el nivel de riesgo C como lo muestra la tabla 1.1. Y si es que lleva el material D, sus rutas considerarán el riesgo de este material. Los materiales B y E a pesar que pertenecen al mismo grupo, se diferencian debido al radio

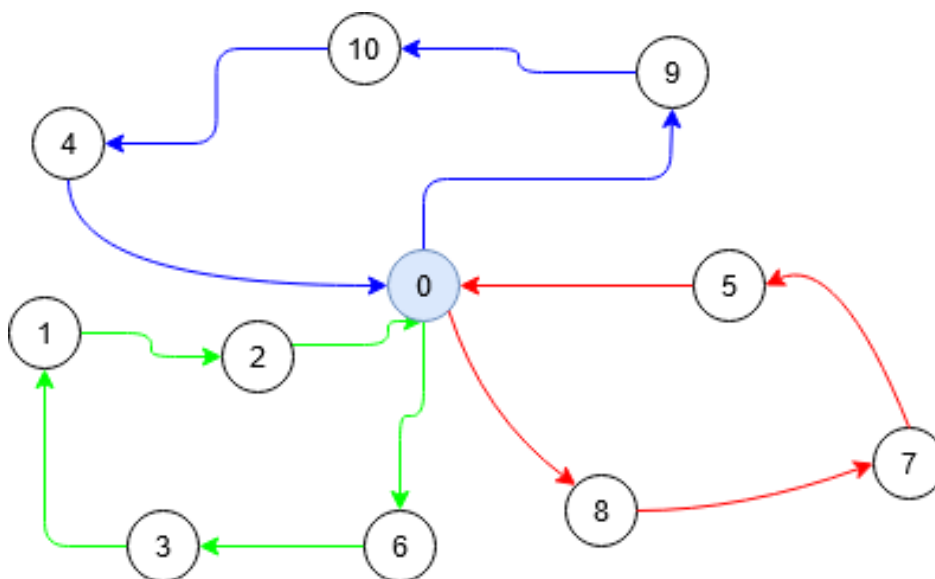


Figura 1.1: Ejemplo de una solución del problema sin considerar materiales peligrosos.

Fuente: Elaboración Propia

de riesgo que tienen, por ejemplo si ocurre un accidente donde el vehículo transporte carga y el material dominante es el B el riesgo será mucho menor que si el vehículo transporte carga y tiene un accidente en el mismo lugar pero con material dominante E. Esto se debe a que con el material B, el radio de riesgo es de 600 metros y el del material E es de 1000 metros.

El caso real a trabajar esta compuesto por un total de 2204 nodos más el lugar de origen, que denominaremos depósito. El depósito corresponde a la fábrica en donde se encuentran todos los materiales para su distribución. Entre los clientes y el depósito, existen diferentes caminos que el vehículo puede tomar para el proceso de distribución y para volver al depósito.

Así descrito, el problema corresponde a un problema de ruteo de vehículos con mezclas.

La figura 1.1 corresponde a un grafo, el cual es una representación de un conjunto de objetos nodos unidos a través aristas. En el grafo, el conjunto de nodos corresponde al conjunto de clientes más el depósito, que en este caso es el nodo con el número 0. Existen distintos caminos entre los nodos que corresponden a las flechas de colores del grafo. Las flechas muestran solo los caminos elegidos por un conjunto de tres camiones que realizan las visitas

a los clientes. Cada color representa a un vehículo distinto. El vehículo que representa el color rojo comenzó su recorrido en el depósito, visitó al cliente número 8, seguido del número 7 y así hasta llegar al nodo número 5 y regresar al depósito. En el caso particular de la figura todos los vehículos comienzan y terminan su recorrido en el depósito.

Como se dijo anteriormente, existen distintos tipos de materiales peligrosos. Para el caso estudiado en este trabajo de memoria se consideran cinco materiales distintos. Cada camino tiene costos y riesgos asociados al material transportado para cada par de clientes, por ejemplo si el vehículo lleva un tipo de material y pasa por la ruta que une a los clientes 7 y 5, el costo o el riesgo o ambos pueden ser distintos, si es que el mismo vehículo con el mismo material pasa por la ruta que une a los clientes 8 y 7 o incluso el costo o el riesgo pueden ser iguales.

Esto, debido a que se transportan materiales peligrosos que afectan a la población expuesta en caso de accidentes, se deben respetar ciertas legislaciones que pueden depender tanto del país o la ciudad en que se realiza el transporte de la carga. En [9], los autores indican que debido a la legislación chilena, decreto supremo 148, en el artículo 87 [21], se estipula que existen incompatibilidades entre los residuos peligrosos. Así, ciertos materiales no se pueden transportar en el mismo vehículo, por posibles explosiones, generación de gases inflamables o tóxicos, reacciones violentas, entre otros. Los autores definen en [9] dos grupos de residuos peligrosos, denominados grupo A (A1 a A7) y grupo B (B1 a B7). Para el presente trabajo de memoria, los materiales peligrosos fueron clasificados según el Decreto Supremo 298[22] de Chile, la Norma Chilena Oficial [5] y al Decreto Supremo 148 [21].

La tabla 1.2 muestra la clasificación bajo las normas chilenas para los materiales a transportar. El grupo A corresponde a materiales que no se pueden reutilizar mediante alguna operación, por lo que deben ser desechados de alguna forma y el grupo B son aquellos que si se pueden reutilizar mediante alguna operación. El material A corresponde a un residuo acuoso, puede ser desechado a través de depósitos permanentes dentro o sobre la tierra o por biodegradación. El material B corresponde a residuo hidrudo, que es un metal o compuesto metálico el cual es posible recuperarlo o regenerarlo. El material C corresponde a un ácido mineral, con el cual es posible recuperar sus componentes para reducir la contaminación. El residuo D corresponde a un residuo derivado del cloruro que debe ser almacenado

Tipo de material	Grupo	Radio de riesgo [metro]
A	A-4	200
B	B-4	600
C	B-7	400
D	A-7	800
E	B-4	1000

Tabla 1.2: Clasificación del material peligroso de acuerdo a las leyes chilenas.

por períodos prolongados. El residuo E corresponde a residuo metálico que al igual que el material B, puede ser recuperado o regenerado.

La figura 1.2 representa un grafo similar a la figura 1.1, tanto en la cantidad de nodos, donde también se incluye un nodo depósito. Los colores rojo, azul y verde corresponden a la representación de los vehículos, que para este ejemplo también son 3. Las principales diferencias están en que la figura 1.2 incluye el transporte de materiales peligrosos, lo que podría modificar o no las rutas respecto al grafo de la figura 1.1, pero para un mejor entendimiento se dejarán las mismas rutas.

En la figura 1.2, se agregaron 3 materiales peligrosos, A, B Y C. Los cuadrados con el tipo de material, apegados a los nodos, corresponden al material que los clientes necesitan, por lo tanto el vehículo debe entregarlo. Las letras apegadas a las flechas corresponden al material dominante, según la tabla 1.1, es decir, que riesgo del material predomina durante ese trayecto.

En la ruta del vehículo rojo, los materiales a transportar son C y A que son distribuidos para los clientes 8, 7 y 5. Durante el trayecto desde el depósito hasta el cliente 8, al cual se le distribuye el material A y el material dominante, debido a la tabla 1.1. Una vez visitado el cliente 8, el vehículo rojo debe visitar el cliente 7, al cual se le distribuye el material C y durante ese trayecto el material dominante sigue siendo C. Una vez distribuido el material C para el cliente 7, se debe transportar el material A para el cliente 5 y durante ese trayecto el material dominante es A. Una vez visitado el cliente 5, el vehículo vuelve sin materiales peligrosos. Al realizar dicho recorrido el riesgo es cero.

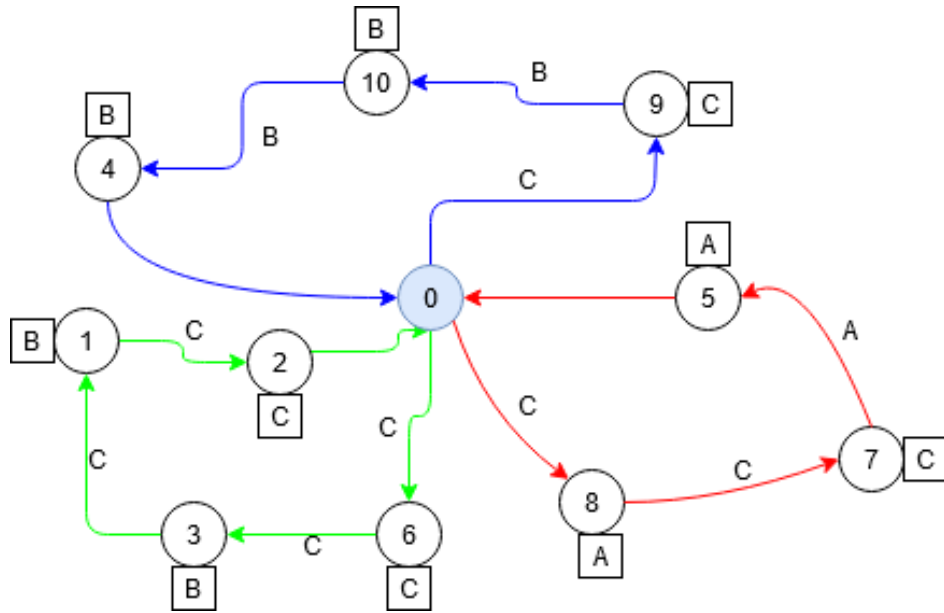


Figura 1.2: Ejemplo de una solución del problema de este trabajo de memoria.

Fuente: Elaboración Propia

En la ruta del vehículo verde, los materiales a transportar son B y C que son distribuidos para los clientes 6, 3, 1 y 2. El vehículo al salir del depósito debe visitar al cliente 6 al cual se le distribuye el material C, durante ese trayecto el material dominante es C. Una vez visitado el cliente 6, el vehículo recorre el trayecto entre este cliente y el 3, al cual se le distribuye el material B, donde el material dominante sigue siendo C. Una vez visitado el cliente 3, el vehículo se dirige hacia el cliente 1 para distribuirle el material B y el material dominante durante ese trayecto sigue siendo C. Luego el vehículo desde el cliente 1 debe visitar al cliente 2 para distribuirle el material C y durante ese trayecto el dominante sigue siendo C. Una vez visitado el cliente 2, el vehículo vuelve sin materiales peligrosos. Al realizar dicho recorrido el riesgo es cero.

En la ruta del vehículo azul, los materiales a transportar son B y C, para los clientes 9, 10 y 4. El vehículo al salir del depósito debe visitar al cliente 9 para distribuirle el material C, durante ese trayecto el material dominante es C. Una vez visitado el cliente 9, el vehículo se dirige hacia el cliente 10, al cual se le distribuye el material B, durante ese trayecto el material dominante es B. Luego el vehículo desde el cliente 10 debe visitar al cliente 4 para

distribuirle el material B, durante ese trayecto el dominante es B. Una vez visitado el cliente 4, el vehículo vuelve sin materiales peligrosos. Al realizar dicho recorrido el riesgo es cero.

Es importante destacar que debido a la incompatibilidad entre materiales o incluso la dominancia de un material sobre otro, el vehículo debe tomar otras rutas que pueden resultar más costosas, en términos de distancia.

Así descrito el problema consiste en determinar un conjunto de rutas que visiten a todos los clientes respetando las capacidades de los vehículos, las restricciones de compatibilidades y minimizando tanto la distancia y el riesgo total de las rutas. El problema puede entonces considerarse como un problema multi-objetivo. En la siguiente sección se describen algunos aspectos fundamentales de los problema multi-objetivo.

1.3.1. Problemas Multi-Objetivo

Los problemas multi-objetivo, son aquellos en que se desea optimizar dos o más objetivos al mismo tiempo. En estos problemas, en la mayoría de los casos no existe forma de encontrar una solución que sea la más óptima para los objetivos que se desean cumplir. Esto se debe a que existen conflictos entre los objetivos o pueden ser opuestos entre sí. Un ejemplo ocurre en un problema donde alguna empresa desee aumentar la producción, pero para lograrlo, debe aumentar la inversión lo que implica que los costos de la empresa aumentan. En este caso se contradicen los objetivos que son aumentar la producción y disminuir los costos de la empresa.

En términos matemáticos, los problemas multi-objetivo definen una solución encontrada como un vector $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T$ que satisfaga las restricciones del problema y optimice la función objetivo, que se puede plantear de forma vectorial:

$$f(x) = [f_1(x), f_2(x), \dots, f_m(x)]$$

Donde x^* pertenece a un conjunto S , que está formado por todas las soluciones que satisfagan las restricciones.

Una forma de tratar el problema de que los objetivos tengan conflictos entre sí es con la *dominancia de Pareto*. Al comparar las diferentes soluciones del conjunto S , es posible determinar que una solución es dominante sobre otra si es que en alguno de los objetivos es de mejor calidad y en el resto de los objetivos la calidad es igual o mejor. De forma matemática, el vector solución $x^* = [x_1^*, x_2^*, \dots, x_n^*]^T \in S$ domina al vector solución $y^* = [y_1^*, y_2^*, \dots, y_n^*]^T$, si y solo si $f_i(x) \leq f_i(y); \forall i \in \{1, \dots, k\}$ y $\exists i \in \{1, \dots, k\}$ tal que $f_i(x) < f_i(y)$. De esta forma es posible decir que $x^* < y^*$ (el vector x^* domina al vector y^*)

Así es posible obtener un conjunto de todas las soluciones que no son dominadas. Una solución x^* se dice que es Pareto óptima si y solo si $\nexists x \in S$ tal que $v = f_i(x) \forall i \in \{1, \dots, k\}$ domine a $u = f_i(x^*) \forall i \in \{1, \dots, k\}$. Dichas soluciones que son Pareto óptimas conforman un conjunto que es llamado Frente de Pareto, definido como $P = \{x \in S, \nexists y \in S : x < y\}$. El Frente de Pareto se presenta en el espacio de objetivos. La figura 1.3 muestra un ejemplo de frente de Pareto para un problema de minimización de dos objetivos.

Las soluciones, cuya imagen se encuentra en el frente de Pareto se denomina conjunto de Pareto.

Así, la meta en los problemas multi-objetivo es encontrar en tiempos acotados el conjunto de Pareto que permite encontrar un frente de Pareto que muestre un alto nivel de convergencia y distribución.

1.4. Modelo Matemático

A continuación se detallan los parámetros, variables, restricciones, naturaleza de las variables y la función objetivo del problema.

1.4.1. Parámetros

K : Conjunto de vehículos.

N : Conjunto de clientes incluyendo el nodo depósito.

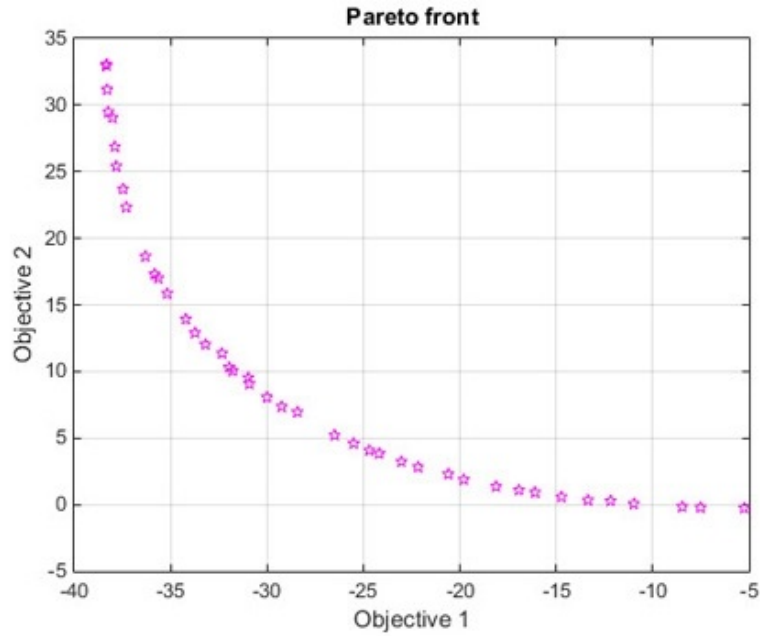


Figura 1.3: Frente de Pareto

Fuente: https://in.mathworks.com/help/gads/multiobj_plot2.png

N_c : Conjunto de clientes sin el nodo depósito.

M : Conjunto con los tipos de materiales peligrosos.

MC : Conjunto de cantidad de materiales peligrosos, donde el i -ésimo elemento corresponde a la cantidad que generó el cliente $n_i \in N$.

T : Conjunto del par: (nodo cliente, tipo de material peligroso), donde el nodo cliente necesita un tipo de material.

A : Conjunto de arcos (rutas entre los nodos).

COM : Conjunto de pares de materiales peligrosos del tipo (m, r) que son compatibles entre sí, es decir, un vehículo puede transportar en la misma carga dichos materiales.

D_0 : Lista de distancia desde el nodo depósito hacia los clientes.

D_m : Matriz de distancia entre los distintos nodos (clientes), si el vehículo transporta el

material $m \in M$.

R_m : Matriz de riesgo entre los distintos nodos (clientes), si el vehículo transporta el material $m \in M$.

Q : Capacidad del vehículo.

1.4.2. Variables

x_{ij}^k = Variable binaria que indica si el vehículo k viaja desde el nodo i al nodo j .

y_i^k = Variable binaria que indica si el vehículo k carga el producto del cliente i .

w_{ij}^{km} = Variable binaria que indica si el vehículo k transporta el material m , a través de la ruta que une a los nodos i y j .

C_i^k = Cantidad de material peligroso que lleva el vehículo $k \in K$ antes de visitar al nodo $i \in N$

1.4.3. Restricciones

Restricción 1: No se puede superar la capacidad de carga de los vehículos.

$$\sum_{i \in N_c} y_i^k \cdot MC_i \leq Q \quad \forall k \in K \quad (1.1)$$

Restricción 2: Indica que la carga que necesita el cliente debe ser distribuida por un solo vehículo.

$$\sum_{k \in K} y_i^k = 1 \quad \forall i \in N \quad (1.2)$$

Restricción 3: El vehículo puede salir del depósito solo una vez. Además permite o no

usar cada vehículo.

$$\sum_{j \in N_c} x_{0j}^k \leq 1 \quad \forall k \in K, 0 \text{ representa el nodo depósito.} \quad (1.3)$$

Restricción 4: Todos los vehículos que entren a un nodo cliente, deben salir de éste.

$$\sum_{i \in N_c} x_{ij}^k = \sum_{h \in N_c} x_{jh}^k \quad \forall k \in K, \forall j \in N_c \quad (1.4)$$

Restricción 5: Si el vehículo k viaja desde el nodo i al j , entonces se le entrega la carga al cliente j , que necesita el material m .

$$\sum_{i \in N} x_{ij}^k = y_j^k \quad \forall k \in K, \forall j \in N_c, (j, m) \in T \quad (1.5)$$

Restricción 6: Un vehículo con carga de un material tipo r , no puede salir del nodo i (nodo actual) con un riesgo mayor a r .

$$\sum_{j \in N} w_{ij}^{km} \geq 1 - y_i^k \quad \forall k \in K, \forall i \in N_c, m, r \in M, (i, r) \in T \quad (1.6)$$

Restricción 7: Se evita la combinación de materiales incompatibles en el mismo vehículo.

$$y_i^k + y_j^k \leq 1 \quad \forall k \in K, \forall i \in N_c, (i, m), (j, r) \in T, (m, r) \in COM \quad (1.7)$$

Restricción 8: Se registra el volumen de materiales después de que el vehículo haya entregado los materiales a un nodo cliente. Además se evita que el vehículo quede atrapado en un sub-tour (ciclos del grafo).

$$C_j^k \leq C_i^k + MC_i - Q \cdot (1 - x_{ij}^k) \quad \forall k \in K, i \in N, j \in N_c \quad (1.8)$$

Restricción 9: Si un vehículo viaja desde el nodo i al nodo j , solo tendrá un tipo de riesgo.

$$\sum_{m \in M} w_{ij}^{km} \geq x_{ij}^k \quad \forall k \in K, \forall i, j \in N \quad (1.9)$$

Restricción 10: Se suma a la restricción 9, si un vehículo viaja desde el nodo i al nodo j , solo tendrá un tipo de riesgo.

$$\sum_{m \in M} w_{ij}^{km} \leq 1 \quad \forall k \in K, \forall i, j \in N \quad (1.10)$$

1.4.4. Naturaleza de las variables

$$x_{ij}^k, y_i^k, w_{ij}^{km} \in \{0, 1\} \quad \forall k \in K, i, j \in N, m \in M \quad (1.11)$$

Al ser variables binarias, solo pueden estar activas o inactivas. Por lo tanto solo pueden tomar los valores 0 o 1.

$$C_i^k \geq 0 \quad \forall k \in K, j \in N \quad (1.12)$$

La cantidad de material peligroso debe ser mayor o igual a cero, debido a que el vehículo puede distribuir o no materiales.

1.4.5. Objetivos

Debido a que el problema es bi-objetivo, se tendrán dos funciones que se deberán minimizar. La primera corresponde al objetivo de minimizar la población expuesta a riesgos del

material peligroso transportado.

$$\text{Min } F_1 = \sum_{k \in K} \sum_{m \in M} \sum_{i \in N_c, j \in N} EP_{ij}^m \cdot w_{ij}^{mk} + \sum_{k \in K} \sum_{m \in M} \sum_{j \in N_c} b_{j0}^m \cdot w_{j0}^{mk} \quad (1.13)$$

La segunda función corresponde al objetivo de minimizar el costo de transporte de materiales peligrosos.

$$\text{Min } F_2 = \sum_{k \in K} \sum_{m \in M} \sum_{i \in N_c, j \in N} c_{ij} \cdot w_{ij}^{mk} + \sum_{k \in K} \sum_{m \in M} \sum_{j \in N_c} c_{j0} \cdot w_{j0}^{mk} \quad (1.14)$$

Donde:

EP_{ij}^M : Cantidad de población en riesgo expuesta al material m entre los nodos i, j .

c_{ij} : Costo de transporte desde el nodo i al nodo j .

$\alpha \in [0, 1]$

α corresponde a una constante que representa el peso que se le da al objetivo de la población expuesta entre un par de nodos. Y el complemento de esta variable es el peso que se le da al objetivo de costo de transporte entre un par de nodos.

1.5. Resumen

El problema de distribución de materiales peligrosos bi-objetivo, trata de una flota de camiones, que buscan transportar materiales peligrosos a distintos clientes, considerando que se desean trasladar los materiales por rutas que sean de bajo costo y además disminuyendo la población expuesta a posibles accidentes durante los viajes. De acuerdo a la legislación chilena, los materiales son clasificados en cinco grupos. Además de que existen incompatibilidades entre ellos, las cuales se traducen en que un vehículo no puede transportar materiales que son incompatibles entre sí. Aquellos vehículos que se utilizan, se les dan trayectorias que cumplan con las funciones objetivos, buscando minimizar la relación entre el costo y

la población expuesta entre un par de nodos y un material transportado. También recordar que los materiales tienen distintos rangos de riesgo, por lo tanto una posible buena opción es transportar los materiales con mayor rango de riesgo durante las primeras rutas de transporte y así evitar que en rutas futuras afecten a la población expuesta en caso de accidente.

Capítulo 2

Estado del Arte

Este capítulo presenta la revisión bibliográfica relacionada al problema multi-objetivo de enrutamiento de vehículos para distribución de materiales peligrosos con mezcla.

Este capítulo se divide en tres secciones. En la primera sección se revisarán acercamientos relacionados al problema de enrutamiento de vehículos general, desde sus inicios, como han ido evolucionando los algoritmos de resolución y cómo han mejorado las soluciones y los tiempos de cómputo gracias al avance de la investigación. En una segunda sección se revisan acercamientos relacionados al problema de enrutamiento de vehículos multi-objetivo. En una última sección se revisan artículos relacionados al problema resuelto en el presente trabajo de memoria.

2.1. Problema de enrutamiento de vehículos

El problema de enrutamiento de vehículos fue introducido por primera vez por Dantzig y Ramser en el año 1959 [8]. En su primera publicación los autores presentan un modelo matemático y para un problema de ruteo de vehículos capaz de atender la demanda del petróleo de un conjunto de estaciones de servicio mediante una flota de camiones. Estos camiones debían transportar el combustible desde un nodo central a varias estaciones de gas, minimizando la distancia recorrida para abaratar costos. El modelo era bastante simple comparado

con los modelos actuales. Dantzig y Ramser llamaron al problema *problema de despacho de camiones* y fue considerado como una generalización del problema del vendedor viajero. Las principales diferencias entre estos problemas se refieren a la existencia de una flota de camiones y además que la capacidad total para almacenar los productos a transportar es mayor que la carga total de todos los productos.

Los autores Wren y Holliday en el año 1972 [1], propusieron un modelo matemático para resolver el problema de *multi-depot vehicle routing problem*. Aquí consideraron restricciones de carga, distancia y de tiempo. Los autores propusieron un nuevo algoritmo que consistía en construir una solución inicial factible y luego mejorar su calidad a través de heurísticas de refinamiento. Para obtener la solución inicial, se debía especificar la cantidad de vehículos por nodo depósito. De esta manera a cada cliente se le asignaba un vehículo que cumpliera con las restricciones de carga y distancia. Por lo tanto al primer cliente se le asignaba la ruta más corta desde el nodo depósito más cercano, lo mismo con el segundo cliente y así hasta completar toda la demanda. La solución inicial se podía refinar a través de procesos, como por ejemplo, intercambio de un nodo de una ruta hacia otra, eliminar ciclos en las rutas de transporte, reducir el número de camiones utilizados, entre otros. Dichos procesos eran utilizados hasta que la solución no se pudiera mejorar más o bien hasta un tiempo de uso prudente o máximo de la computadora que ejecuta el algoritmo.

En los años 90, se comenzó a aplicar métodos de optimización al problema de enrutamiento de vehículos. En [20] los autores utilizaron el algoritmo de optimización de búsqueda tabú. En este artículo, utilizaron una nueva heurística de búsqueda tabú, la cual llamaron *tabroute* y utilizaron para resolver el problema de generación de rutas de vehículos, con restricciones de capacidad y longitud de rutas. En síntesis, a partir de una solución inicial, el algoritmo genera múltiples soluciones al eliminar algún nodo de alguna ruta y reinsertarlo en alguna otra ruta. Esto produce soluciones factibles e infactibles, es decir, que cumplen con las restricciones del problema o no. Los autores indican que producir soluciones infactibles, les permite no quedar estancados en una solución factible, ya que desde las infactibles es posible obtener soluciones de mejor calidad.

Otro método de optimización frecuentemente utilizado para resolver problemas de enrutamiento de vehículos es un enfoque basado en la optimización de colonias de hormigas.

En [18] los autores proponen el *MACS-VRPTW*. Este enfoque simula una jerarquía de colonia de hormigas artificiales inspirado en el comportamiento real de las hormigas [2], capaces de encontrar los caminos más cortos para llegar a su objetivo. Este comportamiento se obtiene gracias a las feromonas depositadas por las hormigas utilizada como fuente de comunicación indirecta. A mayor número de hormigas transitando por un camino, más corto debiera ser dicho camino y una cantidad alta de feromonas depositadas por las hormigas. En [18] los autores utilizaron este enfoque para resolver el problema de enrutamiento de vehículos con ventanas de tiempo, es decir, se le suman restricciones en donde por ejemplo el vehículo debe realizar un viaje entre un par de nodos en un cierto tiempo determinado. Además utilizan un enfoque metaheurístico basado en el comportamiento de búsqueda de alimento de colonias reales de hormigas. La idea principal de este enfoque es que a través de estos agentes (hormigas) artificiales simples es posible construir soluciones para este tipo de problemas. Debido a que estos, cooperan utilizando una memoria común que corresponde a las feromonas que es utilizada para construir nuevas soluciones. Los resultados que se obtuvieron fueron calificados como buenos en comparación a los algoritmos de la época, tanto en calidad de la solución como en tiempos de cómputo del algoritmo.

En la década del 2000, Berger y Barkaoui [15] resolvieron el problema de enrutamiento de vehículos clásico utilizando algoritmos genéticos. Debido a que en esa época, los algoritmos genéticos aún no estaban al nivel de las mejores técnicas, los autores decidieron utilizar un algoritmo híbrido. El enfoque principal de este algoritmo estaba en minimizar la distancia recorrida de todos los vehículos. El algoritmo trabaja con dos poblaciones de soluciones, que intercambian los mejores individuos para que estas poblaciones evolucionen a través de operadores genéticos y además de otras estrategias de búsqueda. Estos operadores genéticos son recombinaciones y mutaciones. El primero de estos lo que hace es basarse en segmentos prometedores de dos soluciones principales y así generar una nueva solución, con esto se intenta mejorar las rutas de uno de los padres (soluciones principales) utilizando buenos genes, que corresponden a los segmentos de rutas prometedores.

Para el segundo operador genético, que es mutación, los autores definen un conjunto de cuatro operadores de mutación. De esta forma se crea un equilibrio entre la búsqueda de soluciones diversas y búsqueda de soluciones de alta calidad. Los resultados obtenidos por

los autores de este artículo, fueron considerados como muy competitivos en comparación a las más técnicas utilizadas para resolver el mismo problema.

El problema de enrutamiento de vehículos ha sido ampliamente estudiado en el área de investigación de operaciones e inteligencia artificial. Los primeros acercamientos se hicieron mayormente en versiones del problema que consideran un único objetivo a optimizar. La naturaleza del problema permite definir más objetivos interesantes en la mayoría de las variantes. A continuación se describen algunos acercamientos relacionados al problema de enrutamiento multi-objetivo.

2.2. Enrutamiento de vehículos multi-objetivo

En [6] los autores proponen un modelo de programación lineal entera mixta bi-objetivo, para Pollution-Routing Problem (PRP), que es una variante del problema VRP, que considera las condiciones del tráfico y la velocidad del vehículo como dos factores importantes, así se puede determinar el consumo de combustible y la contaminación durante el transporte. Además los autores utilizan un caso de estudio que corresponde a una cadena de suministros de lácteos, que cuenta con cuatro proveedores, dos puntos de transbordo y 33 clientes. El primer objetivo considera minimizar el costo total, en el que consideran los costos de contaminación y costos de enrutamiento. El segundo objetivo es maximizar la confiabilidad del suministro.

Los autores presentan dos algoritmos para comparar los resultados obtenidos entre estos para instancias grandes del problema. Uno es el algoritmo Simulated Annealing para problemas multi-objetivo y el otro es un algoritmo genético NSGA II. A diferencia de un simulated annealing tradicional, el algoritmo multi-objetivo simulated annealing propuesto va generando soluciones no dominadas a través de una función de probabilidad. De esta manera se generan soluciones en el frente óptimo de Pareto, para así obtener la mayor cantidad posible de soluciones no dominadas y bien dispersas. Además este algoritmo no requiere de mucha memoria para mantener la información de la población.

La segunda técnica utilizada, NSGA II, está basada en la combinación de dominancia de

Pareto y un algoritmo genético. Lo que este algoritmo hace es determinar la densidad de una solución en el espacio de búsqueda, selección de la solución y así generar otra población utilizando la última solución. La densidad de una solución, es descrita como una medida para ajustar la variedad de la población, ya que considera la solución promedio más cercana para todas las funciones de aptitud. Para la selección de la solución, primero se debe generar una población parental inicial y se evalúa la calidad de cada solución a través del algoritmo de ordenamiento. Se escogen los padres (mejores soluciones del paso anterior) de la población y se les aplican los operadores de mutación y cruzamiento para generar nuevas soluciones. Luego nuevamente se busca la mejor solución evaluando la calidad de las soluciones, pero en este caso es la de los padres y la de los hijos generados. Y se escogen nuevamente los padres para la siguiente generación. Esto se hace hasta alcanzar el criterio de parada. La representación que utilizaron los autores para la solución fue utilizar dos vectores. En uno de ellos almacenaban cantidad de proveedores visitadas por vehículo, origen de cada vehículo en la fase de recolección y destino de cada vehículo. En el segundo vector almacenaban las rutas de los vehículos.

Para comparar los resultados entre ambos algoritmos, los autores utilizan el caso de estudio nombrado anteriormente. ¿Cuál? Además, se contaba con dos y cuatro vehículos para las operaciones de entrega y recogida, respectivamente. Los autores hicieron pruebas con pequeñas instancias para comparar los resultados entre los algoritmos y concluyeron que NSGA II obtiene mejores resultados y, además, en un menor tiempo de computación que MOSA. Por lo tanto utilizaron NSGA II para resolver el caso de estudio y obtuvieron soluciones cercanas a lo que es el frente óptimo de Pareto.

En el año 2013, se propone el algoritmo Simulated Annealing-based parallel multi-objective (MT-PSA) para el problema de enrutamiento de vehículos con ventanas de tiempo [23]. En este problema, existían múltiples restricciones, entre ellas industriales, económicas y ambientales.

El enfoque multi-objetivo que se le da al problema es porque se busca minimizar la distancia recorrida de los vehículos y, además, minimizar el desequilibrio en las distancias recorridas y en las cargas entregadas. El algoritmo Simulated Annealing es una técnica que

imita calentar y luego enfriar un material, esos procesos simulan el hecho de buscar soluciones en un espacio y de encontrar la solución de mejor calidad en un espacio de soluciones respectivamente. Para que existe una alta diversificación en la búsqueda de soluciones, los autores proponen diez operadores, por ejemplo algunos de ellos son, crear o eliminar rutas, modificar el vehículo asignado a un cliente, etc. operadores de cruzamiento, que consiste en crear soluciones factibles a partir del cruzamiento de soluciones padres, que corresponderían a dos rutas padres. Para la comparación entre las soluciones, para aceptar una nueva solución si es que es mejor que la actual, los autores utilizan el criterio de Metrópolis, donde se utilizan parámetros como la suma de las variaciones de la calidad de la soluciones según la función objetivo y la temperatura actual del algoritmo. Los resultados que obtuvieron los autores, los calificaron como positivos debido a que alcanzaban soluciones de alta calidad y se producía una ganancia significativa en los tiempos de ejecución en comparación con algoritmo SPEA2, que es una implementación paralela secuencial y basada en islas.

Los autores del artículo [25] buscan soluciones para el problema de ruteo de vehículos en transporte de valores (*cash in transit vehicle routing problem*) utilizando un enfoque basado en teoría de juegos y toma de decisiones. El modelo utilizado en este artículo, busca aumentar la seguridad del problema enrutamiento de vehículos en tránsito que manejan efectivo. Este recorrido lo hacen entre una tesorería central y las sucursales bancarias, por lo que el interés está en minimizar la distancia total recorrida por los vehículos y además minimizar el riesgo total durante el transporte. Por lo tanto, los objetivos del problema son minimizar la distancia y minimizar el riesgo para optimizar la seguridad del transporte. Similar a lo visto en este trabajo de memoria con la diferencia de que el riesgo en el artículo [25] se interpreta como riesgo de robo de los productos, es decir, es necesario reducir la vulnerabilidad respecto a intentos de robo. Los autores utilizaron un enfoque basado en teoría de juegos y la toma de decisiones (MCDM), para así medir el nivel de riesgo. Un nuevo algoritmo genético inteligente multi-objetivo (MOIGA), que es una variación del algoritmo genético para resolver problemas multi-objetivo que se caracteriza por seleccionar la heurística más eficaz entre las que componen al algoritmo. La representación utilizada para almacenar la solución es un cromosoma, que corresponde a una cadena de strings, de largo igual a la cantidad de clientes. Para los operadores, los autores utilizaron cruzamiento y mutación. Para el primer caso, se escoge de forma aleatoria entre dos operadores los cuales son *best cost-best route crossover*)

y *order crossover*. Y para mutación, se escoge de forma aleatoria entre seis operadores, los cuales son *reubicación externa*, *intercambio externo*, *intercambio cruzado o externo*, *intercambio interno* y *reubicación interna*. En un paso siguiente clasificaban todas las soluciones usando la dominancia de Pareto y aplican un algoritmo de selección inteligente. Y por último, crean una nueva población en base a la descendencia. Estos pasos los repiten hasta un criterio de parada.

Las instancias utilizadas por los autores, son las mismas que se utilizaron en [26]. Estas tienen 100 clientes y cuentan con una cantidad de vehículos entre 9 y 19 dependiendo de la instancia. Obtuvieron resultados iguales o muy cercanos a la mejor solución conocida en todas las instancias. Para el peor caso, la solución con MOIGA, fue un 11 % peor que la mejor solución conocida. El algoritmo MOIGA obtuvo tiempos de ejecución similares comparado con tres algoritmos MOIGA pero sin un operador. Estos eran algoritmos MOIGA sin mutación, otro sin selección con hill-climbing y el último sin cruzamiento. Siempre en alguna instancia el tiempo de ejecución utilizando MOIGA era similar a alguno de los tres algoritmos MOIGA sin algún operador.

En [11], los autores abordan el problema de enrutamiento de vehículos bi-objetivo para el transporte de materiales peligrosos. Similar al problema abordado en este trabajo de memoria, pero donde los vehículos solo pueden transportar un tipo de material peligroso en su ruta. Los autores proponen dos enfoques para tratar el problema y así comparar los resultados entre los dos métodos propuestos. Uno de ellos es un algoritmo de búsqueda local basado en dominancia multi-objetivo. Y el otro corresponde a un algoritmo metaheurístico de restricción ϵ . El primero de ellos es un enfoque que se basa en la selección de un conjunto de puntos que se va a explorar. Y explorar la vecindad de este conjunto basadas en parámetros del algoritmo. El método de búsqueda utilizado es el *Variable Neighborhood Descent*. Utilizan movimientos como shift, swap y 2-opt. El algoritmo restricción ϵ es un algoritmo de búsqueda local, que produce solo soluciones eficientes, es decir utilizando el mejor vecino eficiente, aquel que menos perjudique a la función objetivo. Los algoritmos se prueban en veinte instancias propuesta en [3], y cada enfoque se ejecuta 30 veces por instancia. Cada instancia cuenta con un máximo de hasta cien clientes. Los autores obtuvieron como resultados que en todas las instancias el algoritmo basado en dominancia de vecindad

multi-objetivo fue mejor que el algoritmo metaheurístico de restricción ϵ , con la excepción de una sola instancia.

En [12], los autores realizan un modelo para el problema de enrutamiento de ubicación de objetivo múltiples para logística de materiales peligrosos con restricción de tráfico en carreteras interurbanas. En el artículo se propone un algoritmo genético único, cuyos cromosomas poseen dos tipos de genes, los cuales representan almacenes y las rutas de transporte. Los objetivos del problema son minimizar el riesgo total, entre los cuales se incluye el riesgo del sitio y del transporte. Además de minimizar los costos y maximizar la satisfacción al cliente. La representación utilizada es una estructura cromosómica que contiene la orden del servicio de los clientes, elección de camino entre par de nodos y la señalización por parte del cliente según que material desee, en relación con los diferentes servicios de almacén. El algoritmo del artículo presentado por los autores, como primer paso es obtener una solución factible, es decir, que cumpla con todas las restricciones del problema, se repite este proceso una cantidad definida de veces y se evalúan según la calidad respecto a la función objetivo todas las soluciones factibles creadas en el paso anterior. Se realiza un proceso de selección y los cromosomas pasan por dos operadores, los cuales son cruzamiento y mutación. El operador de cruzamiento, lo que hace es escoger de forma aleatoria genes de dos cromosomas padres y se intercambian entre si, generando hijos. Se seleccionan dos cromosomas con mejor calidad según función objetivo entre los padres e hijos. Para el caso del operador de mutación, se seleccionan al azar dos genes del cromosoma y se cambian sus posiciones para obtener a otro hijo. El algoritmo fue aplicado a un caso real sobre la logística de materiales peligrosos en Shandong, China. Se contaba con tres almacenes de materiales peligrosos y nueve clientes ubicados en lugares distintos. Los autores resolvieron el problema en cuatro casos distintos, los cuales son: sin restricciones de tráfico, con restricciones de tráfico, satisfacción del cliente y solo una ruta por par origen-destino. Los autores concluyen que el modelo es flexible, ya que es aplicable a diversas medidas de costo y riesgo. También que el modelo es completo, al considerar aspectos importantes de la vida real de la gestión de la logística de materiales peligrosos, ya que en la literatura no existían modelos de este tipo. Y además que el trabajo realizado en el artículo es muy útil para las decisiones corporativas, incluso para las que puedan tomar departamentos gubernamentales.

2.3. Transporte de materiales peligrosos

Los primeros estudios relacionados al transporte de materiales peligrosos, datan de principios de la década del 80. Los autores de [27], en una extensa investigación, desarrollan un criterio con el fin de determinar rutas designadas para el transporte de materiales peligrosos. Estas rutas eran designadas según qué tan probable era que ocurriera un accidente en una ruta cualquiera, es decir, entre menor es la probabilidad de accidente, menor es el riesgo de asignar dicha ruta para el transporte de materiales peligrosos. Para el cálculo de la probabilidad de que ocurriera un accidente, los autores consideraron factores como tasa de accidentes en cierta ruta, tipo de material transportado, radio que pueda afectar el accidente y esto conlleva a población afectada, escuelas o residencias cercanas, entre otros. Los autores modelaron la situación para un caso real de la ciudad de Washington D.C. Los materiales peligrosos fueron calificados entre siete tipos: explosivos, gases comprimidos, inflamables, materiales venenosos, materiales radioactivos, sólidos inflamables y materiales corrosivos.

Otro de los primeros estudios relacionados al transporte de materiales peligrosos data del año 1985, en [16] los autores desarrollan un modelo para un caso real en Arizona. Al igual que en [27], los autores utilizaron la probabilidad de que ocurriera un accidente con algún material peligroso para designar rutas, pero no fue lo único que consideraron. También agregan tres variables más, que son: número de eventos ocurridos en una ruta, población en riesgo frente a un posible accidente y volumen de materiales peligrosos transportados. Los autores interpretan dichos eventos como factores de riesgo, por lo tanto cada ruta, tendrá una calificación numérica respecto a la posibilidad de que ocurran dichos eventos. Mediante un cálculo del producto entre los cuatro eventos, se obtiene una calificación general del riesgo de una ruta. Los autores también destacan que, en base a la Agencia Federal para el Manejo de Emergencias, se ponderó cada evento multiplicándolo por un valor de importancia. De esta manera se podía reflejar la importancia de cada evento en relación al riesgo.

En [17], los autores estudian un problema complementario al transporte de materiales peligrosos, que es el combinar los criterios de ubicación y ruta los cuales son: minimizar el riesgo de eliminación de un material peligroso, minimizar el riesgo de ruta y minimizar el tiempo de viaje. En este artículo se propuso un modelo combinado de ubicación y ruta

que utiliza como objetivos los tres criterios mencionados anteriormente. Dado un conjunto de posibles sitios de eliminación de desechos, el modelo combinado selecciona un número predeterminado de sitios para: maximizar la distancia total entre los centros de población, minimizar el riesgo de transporte y minimizar el tiempo de viaje. Los autores no aplican su modelo a un caso real, pero si aplican el modelo en un problema hipotético de 16 nodos.

En [10] se propone un acercamiento similar al de [17], pero incorporando variantes al problema de transporte de materiales peligrosos. En [10] se presenta un modelo multi-objetivo para ubicar instalaciones de disposición o tratamiento de materiales peligrosos. Para esto consideran cuatro objetivos, los cuales son: minimizar el costo operativo total, minimizar el riesgo total percibido, distribución equitativa del riesgo entre los centros de la población y distribución equitativa de la utilidades causada por la operación de las instalaciones del tratamiento. Los autores utilizaron la programación por metas, que permiten representar los cuatro objetivos en solo uno. Y utilizaron funciones de penalización para obtener soluciones de mejor calidad, ya que grandes desviaciones de los objetivos son penalizadas. A pesar de que se toman supuestos bastantes simplificados, como por ejemplo, obviar las decisiones gubernamentales o las leyes que puedan modificar las restricciones o la función objetivo, los autores califican su modelo como una herramienta de ayuda a la toma de decisiones. Basan su anterior afirmación en que el modelo puede entregar una serie de soluciones, pero la decisión final siempre la tomará el encargado de tomar las decisiones, como por ejemplo el dueño de la empresa que está encargada de la flota de vehículos.

Para la década del 2000, los autores del artículo [7] revisan los modelos de riesgo más comunes de la literatura. Además consideran dos axiomas que comprueban si es que los modelos satisfacen dichos axiomas. Uno de éstos es la *monotonidad de la evaluación de la ruta*, es decir, que al agregar arcos a una ruta, su valor no debe disminuir. Las funciones de evaluación de rutas de uso común, satisfacen dicho axioma. El otro axioma es el *atributo de monotonidad*, es decir, si es que un valor de cualquier atributo de un camino aumenta, el valor del camino no puede disminuir. Los autores establecen un modelo que cumpla con los axiomas, con el fin de considerar que el vehículo llega o no a su destino, debido a que durante el trayecto pudo haber tenido algún accidente. El modelo que presentan para cumplir con ambos axiomas propuestos es que consideran que se necesiten múltiples viajes para satisfacer

la demanda, debido a posibles accidentes. Estos son medidos utilizando probabilidades en la función objetivo, en donde se mide la probabilidad de que el transporte a lo largo del camino resulte en al menos un accidente.

En el año 2009, en [4] los autores presentan un nuevo modelo de decisión para el transporte de materiales peligrosos. En donde calibran pesos para cada criterio de selección, dependiendo de lo que desea la entidad interesada en abordar el problema (cliente, empresa, etc). El modelo propuesto estima el riesgo de cada ruta y escoge la ruta con menor riesgo en base a un conjunto de criterios y según los pesos (riesgos) de las rutas. Una vez que se obtienen todas las rutas con sus respectivos riesgos, lo siguiente que hacen los autores es elegir los criterios de evaluación de las funciones objetivos y los indicadores para poder medir el desempeño de las diferentes alternativas. En el artículo se definen cinco funciones objetivos, los cuales no son fijos, son adaptables para cualquier tipo de problema. A través de un proceso matemático, los autores definen una matriz en donde las filas corresponden a las alternativas (rutas) y las columnas corresponden a los criterios de evaluación. De esta manera es posible ver que ruta es mejor comparada con las demás, y para esto se define un criterio, en donde la mejor ruta es aquella que sus elementos de la matriz (que para una ruta corresponden a la fila) sumen más. Para los autores la importancia de este modelo es que se puede personalizar fácilmente para otros estudios y además adaptarse a cualquier problema de enrutamiento. Los autores evalúan el modelo propuesto en un caso de estudio muy pequeño.

En [13] el autor analiza distintos modelos de riesgo para el problema de enrutamiento de vehículos de materiales peligrosos en áreas urbanas. El autor compara cinco acercamientos. El acercamiento LI corresponde a los modelos matemáticos que consideran la variable riesgo pero son independientes de la cantidad del material que se lleva, es decir, se considerará siempre el mismo valor de riesgo para cualquier cantidad de material. El acercamiento LD corresponde a los modelos matemáticos que son dependientes de la cantidad del material que se lleva, es decir, el valor de riesgo puede variar dependiendo de la cantidad de material que se transporta. El acercamiento LI-TR considera el modelo LI, con el vehículo totalmente cargado mientras que el acercamiento LD-TR considera el producto entre LI-TR y el porcentaje de carga de materiales durante el transporte. El acercamiento LI-AP considera el modelo LI, representando la probabilidad de un accidente independiente de la carga y el acercamiento

LD-AP, considera el modelo LD, representando la probabilidad de un accidente dependiente de la carga. Los modelos fueron testeados para el problema de enrutamiento de vehículos de materiales peligrosos, llamado en inglés *Hazardous Materials Vehicle Routing Problem* (HMVRP) y para el problema de la ruta más corta, llamado en inglés *Shortest Path Problem* (SPP). Si bien, no existe un modelo mejor que otro, dado que se centran en distintos objetivos, el autor rescata puntos positivos de cada objetivo. Por ejemplo para el modelo de probabilidad de accidente, se concluye que éste ofrece la mejor compensación económica y es el mejor cuando las consecuencias son inciertas. También concluye que, para los modelos LD, un pequeño aumento en el tamaño de la flota puede mejorar considerablemente la calidad de las soluciones. Por último, en el caso específico de HMVRP, el autor afirma que los resultados muestran un claro conflicto entre los objetivos de sustentabilidad ambiental y social.

En [24], se aborda el problema de enrutamiento de vehículos de materiales peligrosos utilizando el algoritmo sistema de colonia de hormigas. En este acercamiento se considera como criterio heurístico la reducción de costos y también la reducción de la cantidad de población en riesgo ante posibles accidentes durante el transporte de los materiales peligrosos. Por lo tanto consideran como función objetivo la minimización de: cantidad de vehículos utilizados, tiempo de funcionamiento de los vehículos y exposición total al riesgo asociado con el proceso de transporte en la población.

Las instancias para testear el algoritmo fueron obtenidas del artículo [26] en donde utilizaron 6 instancias y cada una de ellas contaba con una cantidad de clientes igual a 25, 50 y 100. El algoritmo da resultados muy buenos y que son cercanos a los óptimos globales, el sistema de colonia de hormigas en un VRP presenta soluciones entre un 0-10 % de discrepancia respecto a las soluciones óptimas globales, pero la mayoría de los resultados tenía una diferencia entre 0-7 %. Convergió a buenas soluciones en aproximadamente 1000 iteraciones, para una instancia de 50 clientes y utilizando 6 vehículos. Cabe destacar que en ninguna instancia utilizada se llegó al óptimo global. Los autores concluyen que sus resultados son tan competitivos como los obtenidos en los enfoques exactos.

En el año 2016 los autores del artículo [19], abordan el problema de enrutamiento de ubicación de desechos peligrosos industriales con múltiples objetivos considerando diferentes

tipos de desechos incompatibles. Al igual que en este trabajo de memoria, los autores consideran la restricción de incompatibilidad entre algunos desechos además de otros factores. Los objetivos que plantean los autores para este tipo de problemas son tres. El primero de ellos es minimizar el costo de transporte de los desechos entre los distintos nodos, por ejemplo entre los nodos de origen, el costo de enviar los desechos desde los nodos de origen a centros de tratamiento, etc. El segundo objetivo es minimizar el riesgo total del sitio para la población que vive alrededor de los centros de tratamiento y eliminación de desechos. Y el último objetivo es minimizar el riesgo de transporte entre las rutas. Al igual que en [6], los autores utilizan el algoritmo genético de clasificación no dominado (NSGA II) y la optimización del enjambre de partículas multi-objetivo (MOPSO). Utilizan dos algoritmos para comparar los resultados y observar cuál funciona mejor para las instancias utilizadas. Los objetivos para este problema son minimizar el costo de la recolección de desechos y el riesgo total para las personas que viven alrededor de los centros de tratamiento y eliminación de desechos. La representación para este problema es una matriz de $(W + N) \times G$, en donde W corresponde al número de tipos de desechos, N al número de instalaciones del sistema y G corresponde al número de nodos de origen. El algoritmo NSGA-II utiliza el mismo procedimiento que en [6], utilizan operadores de cruzamiento y mutación. El primero de ellos, es una recombinación de la información de un par de cromosomas durante el cruce, es decir, los cromosomas hijos pueden tener mejores características en comparación a sus padres. El segundo de ellos, es utilizado para garantizar la diversidad de soluciones y así evitar que se caiga en un óptimo local, y lo que se hace en este artículo es un intercambio al azar entre dos espacios de cromosomas. El algoritmo MOPSO está basado en el movimiento en el espacio de un enjambre de partículas. Cada partícula posee una posición actual y una velocidad que le permiten moverse en el espacio. Cada una de éstas conoce su mejor posición previa, es decir, de la mejor solución encontrada en su proceso y además, también conoce la mejor solución global (de todo el proceso). La actualización de la velocidad y posición de las partículas se hace mediante ecuaciones que están presentadas en el artículo. Los algoritmos fueron testeados en 15 instancias, las cuales tenían entre 6 a 100 nodos, entre 2 y 15 nodos de tratamiento, entre 2 y 6 nodos de reciclaje y entre 2 y 12 nodos de eliminación. Los autores concluyen que el algoritmo NSGA-II tuvo un mejor rendimiento que el algoritmo MOPSO.

Además presenta menores tiempos de ejecución para las instancias grandes, con una cantidad de clientes de 30 o más (hasta 100) presentando un promedio de 180,25 segundos de ejecución entre las 6 instancias más grandes. Y el algoritmo MOPSO un tiempo de 254,375 segundos promedio entre las 6 instancias más grandes. Por lo tanto hay una diferencia significativa de 74,125 segundos, es decir, para instancias grandes NSGA-II es casi un 30 % más rápido.

En el artículo [9], los autores realizan un modelo para aplicarlo al caso real que también se verá en el presente trabajo de memoria. El problema es llamado recolección de materiales peligrosos con carga de múltiples productos. Los objetivos del problema son minimizar la población total expuesta y el costo total del transporte. El estudio es aplicado a un caso real, para la ciudad de Santiago de Chile. El método utilizado es crear un nuevo grafo auxiliar sin conexiones entre los nodos. Además para todos los materiales peligrosos obtienen el valor de cada arco, que consideraría el costo y el riesgo asociado entre todos los pares de clientes que están conectadas entre sí. Con la información anterior, les fue posible encontrar el camino más óptimo entre cada par de clientes utilizando cualquier algoritmo que haga ese cálculo. Luego al grafo auxiliar que estaba sin conexiones van agregando los arcos que forman el camino entre los pares de nodos escogidos. Obteniendo un grafo totalmente conectado. Con este grafo resuelven el problema con las restricciones planteadas, como por ejemplo no sobrepasar la carga total del vehículo, cada cliente es visitado una única vez, etc. Como último paso resuelven la búsqueda de la ruta de cada vehículo sobre la red real, donde no necesariamente todos los clientes tienen un camino entre sí, esto se obtiene reemplazando cada arco del grafo auxiliar con su trayectoria correspondiente de la red real. Los resultados de este artículo no los comparan con algún otro algoritmo, pero si pudieron observar que su solución se acerca bastante a lo que es el frente de Pareto.

2.4. Resumen

El problema de enrutamiento de vehículos es un problema de optimización combinatoria, en donde se busca que una flota de vehículos atienda a un grupo de clientes a través de las rutas más óptimas. El problema tiene una amplia gama de variantes, debido a que al

ser aplicados al mundo real, se ve la necesidad de agregar nuevos objetivos o restricciones al problema. Como por ejemplo la satisfacción del cliente, el hecho de agregar materiales peligrosos, añadir ventanas de tiempo para los transportes, etc. En presente capítulo se vieron 3 secciones del problema enrutamiento de vehículos. La primera de ellas la original en donde se citaron los primeros artículos en tratar sobre este problema. En una segunda sección se presentó lo que son los problemas de enrutamiento de vehículos multi-objetivo, en donde ya no solo se consideraba la minimización del costo (objetivo más común), sino que se agregaban otros objetivos. Y en una tercera sección se ven los tipos de problema similares a los de este trabajo de memoria, en donde se agregan materiales peligrosos, lo que conlleva a agregar otro objetivo, que es minimizar la población expuesta a posibles accidentes de estos materiales peligrosos durante en transporte.

Capítulo 3

Propuesta

Para resolver el problema de distribución de materiales peligrosos con mezcla, se propone la utilización de los algoritmos Greedy y Tabu Search. El uso del algoritmo Greedy se justifica debido a que para inicializar el algoritmo Tabu Search (TS), se necesita un punto de partida. El algoritmo Greedy es capaz de construir soluciones seleccionando, paso a paso, la mejor componente a incluir en la solución actual. Para este problema, esto se traduce en que para cada asignación de un cliente a una ruta de un camión, se escoge el camino con menor peso de arco respecto a la función objetivo. Cabe destacar que en el algoritmo Greedy propuesto, el criterio utilizado para escoger la siguiente componente es escoger la opción óptima en cada paso local. Esto significa que una vez visitado un cliente, se dirige al próximo cliente para el cual el camino sea el de menor costo posible. Es por esto que el algoritmo Greedy utiliza solo la información de las distancias. Una vez obtenida la solución inicial con Greedy, se utiliza el algoritmo TS para realizar la búsqueda local del espacio de soluciones promoviendo la diversificación con el criterio de aceptación de peores soluciones que implementa. El acercamiento implementa también un archivo de almacenamiento de soluciones no dominadas que permite resolver el problema de naturaleza multi-objetivo.

En este capítulo se describe el acercamiento propuesto. Comienza presentando la representación de las soluciones, la función de evaluación utilizada para determinar la calidad de las soluciones, los algoritmos utilizados junto con sus componentes en detalle, el uso de la lista tabú y el archivo de soluciones no dominadas.

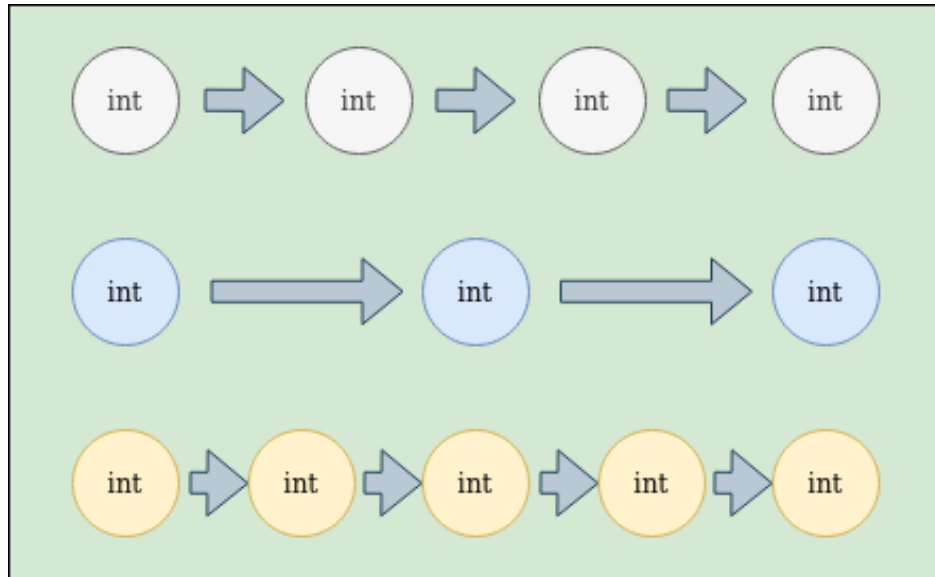


Figura 3.1: Representación de la solución.

Fuente: Elaboración Propia

3.1. Representación

Lo que se busca en este problema, es encontrar rutas óptimas y factibles para la distribución de los materiales peligrosos, transportados por un conjunto de camiones con destino un grupo de clientes distribuidos geográficamente. Debido a lo anterior, la estructura de datos que se utilizó para la representar las soluciones es una listas de listas. En donde cada lista interior, representa la ruta de un camión. En cada ruta se almacenan los identificadores de los clientes visitados, los cuales son representados como valores enteros. Por lo tanto, el largo de la lista general es igual a la cantidad de camiones. Y el largo de las listas que representan las rutas de cada camión es variable. Un ejemplo de solución utilizando la representación indicada se puede observar en la figura 3.1. En dicha figura se pueden observar tres filas que representan las rutas de los camiones. En cada ruta se encuentran nodos que representan a los clientes. Por ejemplo la ruta del primer camión, contiene cuatro clientes representados con color gris, la ruta del segundo camión, contiene tres clientes representados con color azul y el último camión contiene cinco clientes representados con el color amarillo.

El vector que representa la ruta del camión no considera el nodo depósito ni al principio,

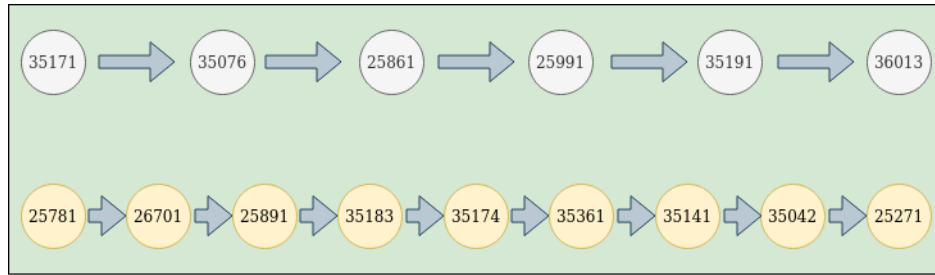


Figura 3.2: Representación de la solución de la zona 3.

Fuente: Elaboración Propia

ni al final de la ruta. Debido a que es un gasto innecesario en almacenamiento guardar dicho nodo en cada ruta, tanto al inicio como al final de éstas. De esta manera solo se almacenan los clientes visitados y el orden en que se visitan. Así se ahorra almacenamiento en los vectores y también tiempo, lo que se traduce en eficiencia y eficacia respectivamente.

Cabe destacar que el algoritmo trabaja solo con soluciones factibles, por lo que todas las soluciones representadas siempre cumplen con todas las restricciones del problema. Por ejemplo, se debe cumplir con que todos los materiales transportados son compatibles y se respeta la capacidad de los camiones.

En las figuras 3.2 y 3.3 se muestra un ejemplo de solución para la instancia real del conjunto de pruebas denominada zona 3. Dicha instancia considera 15 clientes sin incluir el nodo depósito y 2 camiones de recolección. En la imagen es posible ver las dos rutas de los dos camiones. La primera ruta cuenta con la distribución de materiales para seis clientes y la segunda ruta para nueve clientes. En la figura 3.2, en cada ruta se observan los identificadores de los clientes. Y en la figura 3.3 se muestra la misma solución, pero se detallan los materiales que fueron distribuidos a cada cliente junto a la cantidad de éste. Se denota en color verde junto con un signo +, ya que el camión entrega dicha cantidad de material al cliente que visita, disminuyendo la carga que lleva para el transporte, lo que se traduce en un aumento de su capacidad.

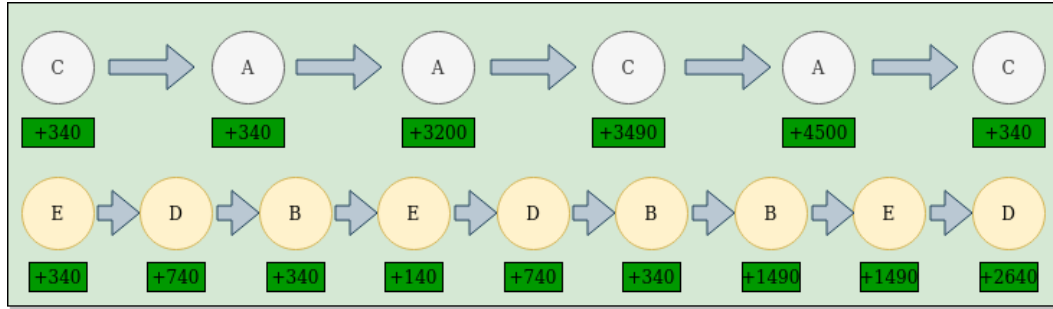


Figura 3.3: Representación de materiales de la solución de la zona 3.

Fuente: Elaboración Propia

3.2. Función evaluación

Para este problema multi-objetivo, la calidad de la solución se mide a través de las funciones objetivo. Esta función de evaluación contempla la relación entre los objetivos que son minimizar los riesgos y minimizar las distancias recorridas. Sea Z la calidad de la solución, Z_1 el costo total de las distancias y Z_2 el costo total de los riesgos. En este caso la función de evaluación se calcula como la suma ponderada de los objetivos. La función de evaluación usada se muestra en la ecuación 3.1,

$$Z = \alpha \cdot Z_1 + (1 - \alpha) \cdot Z_2 \quad (3.1)$$

La función de evaluación requiere la incorporación del parámetro α , el cual se utiliza para determinar el peso relativo de los objetivos, es decir, la importancia de un objetivo por sobre el otro. El parámetro α puede tomar valores entre 0 y 1, inclusive. Cabe destacar que si α toma el valor 0, se desprecia la distancia en la función de evaluación, ya que cualquier valor de distancia multiplicado por α , valdría 0. Así, la búsqueda en este caso se concentra solo en minimizar los riesgos asociados a las rutas. Por otro lado, para el caso en que α es 1, se desprecia el valor de los riesgos, debido a que cualquier valor del riesgo multiplicado por $(1-\alpha)$ es igual a cero. En este caso la búsqueda se concentra solo en encontrar soluciones que minimicen la distancia total recorrida.

Para el cálculo de la distancia y el riesgo entre un par de clientes, se tiene una matriz

completa de nodos de la ciudad de Santiago. Dichos nodos corresponde a cruces de calles en la ciudad de Santiago de Chile completa. En dicha matriz se tienen todos los pares de nodos por los que es posible transitar. Para cada par de nodos se tiene una distancia asociada y un riesgo dependiendo de los cinco diferentes tipos de materiales transportado en la ruta. Por lo tanto un camino entre un par de clientes es una secuencia de visitas de varios nodos. La ruta de mínima distancia/riesgo es calculada mediante el algoritmo Dijkstra. Para esto, el algoritmo considera la ponderación de la importancia relativa de los objetivos ponderados.

3.3. Estructura general

El algoritmo propuesto funciona como un algoritmo de búsqueda tabú. El algoritmo comienza su proceso de búsqueda generando una solución inicial a través de un algoritmo Greedy. Una vez obtenida la solución inicial, se escoge un operador al azar para generar el vecindario de la solución. La propuesta implementa dos operadores de búsqueda, insert y 2opt. Posteriormente se actualiza el archivo de soluciones no dominadas para incluir el óptimo local encontrado durante la fase actual.

Algoritmo 1 Main()

```

1: Initialize  $s_{start} \leftarrow Greedy()$ 
2: Initialize  $solsDom \leftarrow []$ 
3: Initialize  $TabuList \leftarrow []$ 
4: Initialize  $s_{current} \leftarrow s_{start}$ 
5:  $solsDom.push(s_{start})$ 
6: for  $alpha$  in  $alphas$  do
7:   while Stop criterion is not met do
8:      $s_{current} \leftarrow getSolMov(s_{current})$ 
9:     if  $TabuListFunction(mov, TabuList)$  then
10:       $solsDom \leftarrow paretoFront(s_{current}, solsDom)$ 
11:    end if
12:  end while
13: end for

```

En el algoritmo 1 se presenta la estructura general del código utilizado para resolver el problema del presente trabajo de memoria. En las primeras tres líneas se inicializan tres de las variables más importantes del problema. Estas variables corresponden a la solución inicial, una lista que contiene las soluciones no dominadas encontradas durante el proceso y la lista tabú. La solución inicial es construida utilizando el algoritmo Greedy que será explicado en la próxima sección. La lista que contiene las soluciones almacena todas las soluciones que no son dominadas entre sí y que representan la aproximación del frente de Pareto que se construye en el algoritmo. La lista tabú, almacena los últimos movimientos realizados de modo de impedir que el algoritmo genere ciclos durante el proceso de búsqueda.

Luego, el algoritmo debe iterar por cada valor de α determinado como parámetro de entrada. Para cada valor se deben realizar tantos movimientos como permita la condición de termino que se muestra en la línea 7. Cada vez que se encuentra una solución de mejor calidad esta actualiza la solución actual del proceso. En cada caso, para ambos movimientos, se selecciona como nueva solución aquella generada por el movimiento que permite obtener el mejor vecino de la solución actual, aunque no necesariamente sea mejor que la solución actual. La aplicación de cada movimiento está restringido por la información almacenada en la lista tabú. Además, en cada iteración se revisa si la nueva solución debe ingresar al conjunto de soluciones no dominadas. Finalmente, lo que se considera como resultado del algoritmo es el conjunto de de soluciones no dominadas.

A continuación se explica en detalle cada uno de los procedimientos implementados.

3.3.1. Algoritmo Greedy

El algoritmo Greedy es utilizado para construir una solución inicial factible. En este proceso se construye una solución a partir de una solución inicialmente vacía y, paso a paso, se van incorporando componentes a la solución hasta conseguir una solución completa.

Así, el algoritmo comienza escogiendo un camión al azar, y calculando todos los costos posibles entre la posición actual del camión, inicialmente en el depósito, hacia todos los

clientes. De esta manera es posible obtener el costo mínimo entre la posición actual del camión y algún cliente que no haya sido visitado aún. El cliente seleccionado, solo se añade a la ruta del camión si es que cumple con las restricciones del problema. Principalmente se debe cumplir que el material que se suma a la carga transportada por el camión, debe ser compatible con todos los materiales del camión. Y además no se debe superar la capacidad total de camión al sumar carga. Este proceso se repite hasta que no queden clientes por asignar a rutas de camiones. Debido a que este algoritmo va construyendo la solución paso a paso, es posible que algún cliente no esté asignado a ninguna ruta por problemas de compatibilidad o de capacidad. En ese caso, se reinicia el algoritmo para comenzar con una nueva construcción de solución inicial. Se realizan tantos re-inicios como sean necesarios hasta encontrar una solución factible que visite a todos los clientes.

Cabe destacar que para este algoritmo, solo se consideró el objetivo de minimización de distancias para elegir la mejor opción. Esto se hizo para facilitar la búsqueda de una solución inicial factible de calidad, dado además que el efecto del riesgo no es posible medirlo hasta conocer la última visita de cada ruta.

El algoritmo 2 muestra la estructura del algoritmo Greedy implementado. El proceso itera mientras queden clientes sin visitar (línea 9). En cada paso se elige aleatoriamente un camión (línea 5) y se calculan los costos mínimos entre par de clientes, es decir, desde el cliente actual en el que se encuentra el camión y hacia los cuales se dirigirá. En una primera instancia, el algoritmo debe inicializar la variable solución del algoritmo, llamada *solGreedy* esta tiene la misma estructura definida en la figura 3.2. Además se inicializa la variable *clients*, que corresponde a una lista y representa a todos los clientes que quedan por visitar. Luego el algoritmo entra en un ciclo *while*, en donde itera siempre que se cumpla la condición de que el tamaño de la lista *clients* sea mayor a cero, lo que se traduce en que aún hay clientes por visitar y además que aún no se alcance el criterio de término.

En las siguientes tres líneas, se escoge un camión al azar (línea 4), también el cliente más cercano en términos de costo para la ruta del camión escogido (línea 5) y en la variable *matClient* se obtiene el material del cliente escogido (línea 6). En la línea 7, se establece la condición de que el material del cliente que se requiere visitar, es compatible con todos los materiales de la ruta del camión. Si esta condición se cumple, se agrega el cliente en la ruta

correspondiente, tal como se muestra en la línea 8. Además, a la variable *clients* se le quita el cliente que ya fue visitado por un camión. De esta manera se evita visitar a un cliente dos o más veces.

En la línea 12, se establece la condición de que si aún existen clientes por visitar, se debe reiniciar el algoritmo y empezar de nuevo. Es posible que se cumpla dicha condición, debido a que se puede haber alcanzado el criterio de término o por problemas de compatibilidad de los materiales en la ruta de un camión o también por la capacidad del camión que se pudo haber visto superada. Por situaciones como las mencionadas anteriormente, es posible que aún queden clientes sin visitar. En caso contrario el algoritmo retorna la solución, que corresponde a la solución inicial del algoritmo de búsqueda.

Algoritmo 2 Greedy()

```

1: Initialize solGreedy  $\leftarrow$  [[]]
2: Initialize clients
3: while size(clients) > 0 and Stop criterion is not met do
4:   truck  $\leftarrow$  getRandomTruck()
5:   client  $\leftarrow$  getMinClient(truck)
6:   matClient  $\leftarrow$  clientMaterial
7:   if matClient isCompatible with materials(Truck) then
8:     solGreedy(truck).push(client)
9:     clients.delete(client)
10:  end if
11: end while
12: if size(clients) > 0 then
13:   Greedy()
14: else
15:   return solGreedy
16: end if

```

3.3.2. Operadores de búsqueda

En el algoritmo 3 se presenta la estructura del proceso de selección de los movimientos para aplicarlos en una solución del problema. Dicha función se utiliza en la línea 5 del algoritmo 1. El algoritmo 3 debe recibir como parámetro la solución del problema, para poder aplicarle algún movimiento. Y debe inicializar la variable *prob*, que corresponde a un número aleatorio entre 0 y 1. La variable *probInsert* corresponde a la probabilidad de escoger dicho movimiento, por lo tanto tal como se establece la condición en la línea 5, se escoge el movimiento Insert, si es que se cumple que *prob* es menor a *probInsert*. En caso contrario se escoge el movimiento 2-OPT. En el movimiento insert se establece que en las líneas 3 y 4, se escogen los camiones sobre los cuales se aplicará el movimiento y se debe cumplir la condición de que estos camiones sean distintos. En la línea 5, se escoge la posición en la cual se insertará el cliente en el camión 1 ($Truck_1$). En la línea 7, se llama a la función *Insert* con los parámetros necesarios. En el movimiento 2-OPT, en la línea 10, se escoge el camión sobre el cual se aplicará el movimiento y en la línea 11 se llama a la función *2-OPT* con los parámetros necesarios.

Algoritmo 3 *getSolMov(Sol)*

```
1: Initialize prob  $\leftarrow$  random(0, 1)
2: if prob < probInsert then
3:    $Truck_1 \leftarrow$  getRandomTruck(Sol)
4:    $Truck_2 \leftarrow$  getRandomTruck(Sol)
5:    $pos1 \leftarrow$  getRandomPos( $Truck_1$ )
6:   if  $Truck_1 \neq Truck_2$  then
7:      $solMov \leftarrow$  Insert(Sol,  $Truck_1$ ,  $Truck_2$ ,  $pos1$ )
8:   end if
9: else
10:   $Truck_1 \leftarrow$  getRandomTruck(Sol)
11:   $solMov \leftarrow$  2-OPT(Sol,  $Truck_1$ )
12: end if
13: return solMov
```

A continuación se explica en detalle la implementación de cada operador de búsqueda.

3.3.3. Insert

La idea del movimiento *insert* es cambiar el conjunto de visitas de cada ruta. Para esto el movimiento quita un cliente de una ruta, es decir, de los clientes que visita un cierto camión e inserta este cliente en otra ruta distinta. Este movimiento está muy acotado a la restricción de compatibilidad, debido a que en la ruta que es insertado un cliente, debe cumplir con la restricción de compatibilidad, en caso contrario no es posible realizar el movimiento. Para este operador se necesitan dos rutas, que son definidas por dos camiones escogidos y además dos posiciones. Una es la posición en donde se insertará dicho cliente en una ruta y la otra denota la posición del cliente en la ruta de donde se extrae. Este último indicador es variable durante la ejecución del movimiento, debido a que se van iterando todos los posibles indicadores para la ruta en la que se extrae un cliente. De esta forma se diversifica aún más el espacio de soluciones. Dentro de todas las soluciones visitadas, se escoge la mejor.

En el algoritmo 4 está detallado el funcionamiento del movimiento Insert. La función debe recibir cuatro parámetros, que son la solución en la que se va a aplicar el movimiento, dos camiones involucrados para aplicar el movimiento y la posición (*pos1*) en que se insertará el cliente escogido en la ruta del camión. En las primeras tres líneas de código, se inicializan tres variables. La primera de ellas llamada *min*, es utilizada con la intención de almacenar la calidad de la mejor solución encontrada. *Route1* y *Route2*, son las rutas de los camiones *Truck1* y *Truck2* respectivamente. En la línea 4, se recorren todos los clientes de la ruta 2 y dentro del ciclo for, se realiza la parte más importante de este algoritmo. Como primer paso, se elimina el cliente escogido de la ruta 2 y se inserta en la ruta 1, específicamente en *pos1*. En *solMov* se almacena la nueva solución. En la línea 8, se revisa si es que la calidad de la nueva solución es mejor que la almacenada hasta el momento. En caso de que se cumpla dicha condición, se actualiza la variable *min* y además en *solMov* se almacena la solución con la mejor calidad para ser retornada en la línea 13.

Un ejemplo del uso de este movimiento es el presentado en la figura 3.4, en donde se observan dos situaciones. Por el lado izquierdo, se observan dos rutas de camiones, la primera contiene tres clientes y la segunda cuatro clientes. La flecha roja indica que el cliente con identificador 26701 de la ruta del camión 2, será eliminado de la ruta en la que se encuentra

Algoritmo 4 Insert($Sol, Truck_1, Truck_2, pos_1$)

```
1: Initialize  $min \leftarrow infinity$ 
2: Initialize  $Route_1 \leftarrow Sol[Truck_1]$ 
3: Initialize  $Route_2 \leftarrow Sol[Truck_2]$ 
4: for  $Client_{T_2}$  in  $Route_2$  do
5:    $Route_2.delete(Client_{T_2})$ 
6:    $Route_1.insert(Client_{T_2}, pos_1)$ 
7:    $solMov \leftarrow update(Sol)$ 
8:   if  $eval(solMov) < min$  then
9:      $min \leftarrow eval(Sol)$ 
10:     $solMov = Sol$ 
11:   end if
12: end for
13: return  $solMov$ 
```

actualmente y se insertará en la segunda posición de la ruta del camión 1, tal como se observa en el lado derecho de la figura. Como resultado se obtiene una nueva solución en que el camión 1 visita ahora a cuatro clientes y el camión 2 visita solo tres clientes.

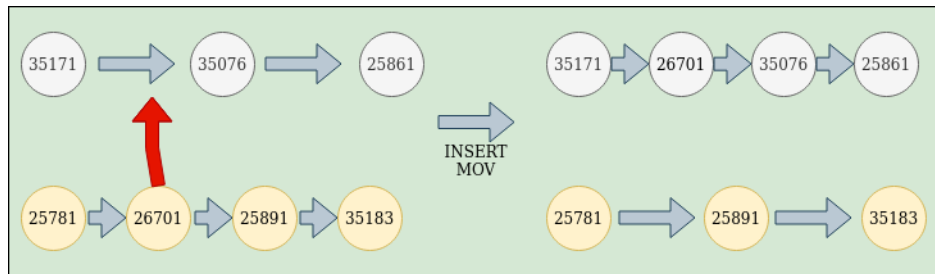


Figura 3.4: Ejemplo uso del movimiento insert.

Fuente: Elaboración Propia

3.3.4. 2-opt

El movimiento 2-opt tiene por objetivo mejorar las secuencias de visitas de las rutas. En el algoritmo propuesto es utilizado con la intención de buscar soluciones de mejor calidad en el vecindario de una solución. Dada una solución y una ruta, definida de acuerdo al camión escogido, se construyen diferentes rutas alternativas a través de movimientos 2-opt. Luego de obtener una ruta sobre la cual va a operar, y dados los puntos **i** y **k** de la ruta, el operador divide la ruta en tres partes de la siguiente forma.

- Tomar una subruta desde el primer cliente hasta el cliente $i - 1$
- Tomar una subruta desde el cliente i hasta el cliente k , e invertir esta subruta.
- Tomar una subruta desde el cliente $k + 1$ hasta el final.

Una vez obtenidas las tres subrutas, solo basta con unir las.

Así, para generar un vecindario usando este operador, es necesario solo cambiar los valores de **i** y **k**. Al igual que en el movimiento insert, en 2-opt los valores **i** y **k** son variables en la ejecución del algoritmo. Estos valores van tomando todas las posiciones posibles de la ruta del camión a la que se le aplicará el movimiento. Además se debe cumplir que las posiciones sean válidas, por ejemplo que el valor de **k** siempre debe ser mayor al de **i** y, a su vez, que los valores no superen el largo total de la ruta. A partir del conjunto de rutas generadas el movimiento retorna aquella de mejor calidad.

En el algoritmo 5, se detalla el funcionamiento del algoritmo 2-OPT implementado.

El procedimiento debe recibir como parámetros la solución a la que se le aplicará el movimiento y el camión involucrado en el operador. En las primeras dos líneas se inicializan las variables *min*, que es utilizada para almacenar la calidad de la mejor solución encontrada. Y la variable *Route*₁, que corresponde a la ruta del camión involucrado. En las siguientes líneas se encuentra un doble ciclo for. El primer ciclo debe recorrer la ruta completa y el segundo ciclo recorre la ruta desde el *Client*₁ hasta el final de la ruta. Se inicializan las variables **i** y **k** para construir las subrutas de *Route*₁. Las subrutas corresponden a *subRoute*₁, *subRoute*₂

y $subRoute_3$. La primera va desde la primera posición hasta la $i - 1$ de $Route_1$. La segunda va desde las posiciones i hasta k y se le aplica un reverse, tal como se indica en la línea 12. La tercera subruta va desde las posiciones $k + 1$ hasta el final de la ruta. En la variable $solMov$ se almacena la nueva solución con el movimiento aplicado. Luego se establece la misma condición que en el algoritmo 4, si es que la calidad de la nueva solución es mejor que la almacenada hasta el momento. En caso de que se cumpla se actualiza la variable min y además se inicializa la variable $solsMov$, que corresponde a la solución del algoritmo que será retornada en la línea 19.

Algoritmo 5 2-OPT($Sol, Truck_1$)

```

1: Initialize  $min \leftarrow infinity$ 
2: Initialize  $Route_1 \leftarrow Sol[Truck_1]$ 
3: for  $Client_1$  in  $Route_1$  do
4:   for  $Client_2$  in  $Route_1$  do
5:      $i \leftarrow getPos(Client_1, Route_1)$ 
6:      $k \leftarrow getPos(Client_2, Route_1)$ 
7:      $subRoute_1 \leftarrow Route_1[0, i - 1]$ 
8:      $subRoute_2 \leftarrow Route_1[i, k]$ 
9:      $subRoute_3 \leftarrow Route_1[k + 1, len(Route_1)]$ 
10:     $subRoute_2 \leftarrow subRoute_2.reverse()$ 
11:     $Route_1 \leftarrow subRoute_1 + subRoute_2 + subRoute_3$ 
12:     $solMov \leftarrow update(Sol)$ 
13:    if  $eval(solMov) < min$  then
14:       $min \leftarrow eval(Sol)$ 
15:       $solMov = Sol$ 
16:    end if
17:  end for
18: end for
19: return  $solMov$ 

```

Un ejemplo de la aplicación de este movimiento se presenta en la figura 3.5. En esta figura se observan dos situaciones. Por la parte superior, se observa la ruta de un camión que contiene cinco clientes. El valor de i en este caso es 1 y el valor de k en este caso es 4.

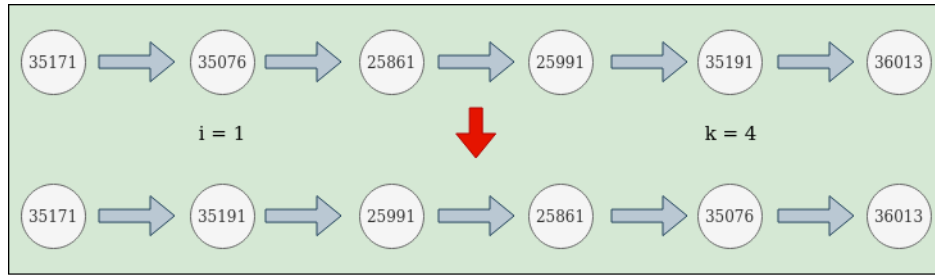


Figura 3.5: Ejemplo uso movimiento 2-OPT.

Fuente: Elaboración Propia

Considerando que la primera posición se numera como 0, las subrutas quedan definidas de la siguiente forma:

- **Subruta 1:** Desde 0 a $i - 1$, lo que corresponde a un único cliente.

[35171]

- **Subruta 2:** Desde i hasta k y se le aplica un reverse, lo que corresponde a:

[35191, 25991, 25861, 35076]

- **Subruta 3:** Desde $k + 1$ hasta el final de la ruta, lo que corresponde a un único cliente.

[36013]

Así, en la parte inferior de la figura 3.5 se puede observar la nueva ruta del camión luego de aplicar el movimiento 2-OPT.

3.3.5. Lista Tabú

El uso de la lista tabú, se justifica para impedir que movimientos ya realizados se vuelvan a repetir. Para esto se utiliza una estructura de datos con límite finito y tamaño *tabuSize*, en donde se almacenan los últimos movimientos realizados. El tamaño *tabuSize* de la lista, es

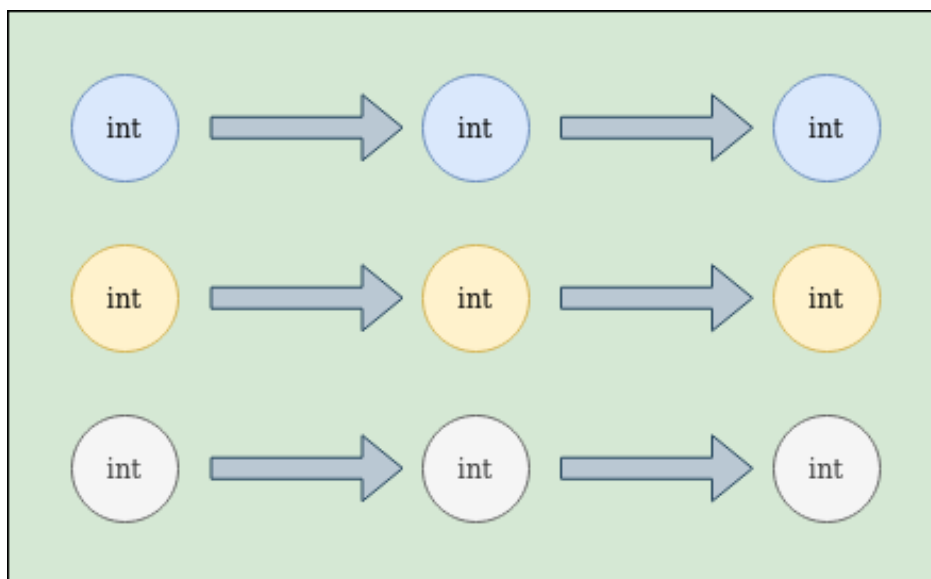


Figura 3.6: Representación del almacenamiento de la lista tabú.

Fuente: Elaboración Propia

definido en base a un porcentaje de la cantidad total de clientes, por ejemplo si existen 18 clientes y se define el porcentaje como 30 %, el tamaño de la lista tabú (*tabuSize*) es 5.4, pero como debe ser un valor entero, se tomará la parte entera, lo que corresponde al valor 5.

La representación utilizada de la lista tabú es una lista de listas, tal como se indica en la figura 3.6. Cada una de las filas representa un movimiento realizado y, para cada uno de ellos, se almacenan las variables implicadas en la aplicación del movimiento. Para el caso de insert, se almacenan los dos camiones involucrados y la posición en que se insertará el cliente en el primer camión. Para el caso del movimiento 2-OPT, se necesita almacenar solo una variable que es el camión involucrado en el movimiento, por lo tanto las otras 2 variables se registran con el valor -1, para evitar confusiones. No es necesario guardar el tipo de movimiento, ya que es posible diferenciarlo al observar si es que éste tiene variables con el valor -1.

En el algoritmo 6, se detalla el cómo se utiliza la lista tabú. En una primera instancia se inicializa la variable *flag*, que es la variable a retornar del algoritmo, ésta representa si es que se pudo insertar el movimiento candidato en la lista tabú. Se establece la condición de verificar si es que el movimiento no está en la lista tabú, para no repetirlo en las últimas

Algoritmo 6 TabuListFunction(*mov*, *listTabu*)

```
1: Initialize flag  $\leftarrow 0$ 
2: if mov not in listTabu then
3:   listTabu.push(mov)
4:   flag  $\leftarrow 1$ 
5:   if size(listTabu) == tabuSize then
6:     listTabu.delete(listaTabu[0])
7:   end if
8: end if
9: return flag
```

iteraciones. En caso de que se cumpla la condición, se agrega el movimiento a la lista tabú y el valor de *flag* cambia a 1, lo que se traduce en que si se puede realizar el movimiento. En caso de que la lista tabú alcance el mismo tamaño que *tabuSize*, tal como se indica en la condición de la línea 5, se debe eliminar el movimiento más antiguo de la lista, lo que corresponde al primero de la lista. Finalmente se retorna el valor de *flag*, tal como se observa en la línea 9.

3.3.6. Actualización archivo de soluciones no dominadas

El objetivo de un algoritmo de búsqueda heurística multi-objetivo es encontrar un conjunto de soluciones no dominadas que en el espacio objetivo se acerquen lo más posible a lo que se considera el frente de Pareto real del problema. Para esto, en esta propuesta se utiliza un *archivo* que permite almacenar la aproximación actual al frente de Pareto que se ha construido en el proceso de búsqueda. Así, cada vez que se aplican los movimientos a una solución para construir un vecindario y de éste obtener la mejor solución, ésta pasará a ser parte del archivo de soluciones no dominadas si es que cumple con las condiciones necesarias para ser añadida. Las condiciones necesarias son que la solución candidata debe ser no dominada con respecto a todas las soluciones que ya se encuentren en el archivo. Otra condición para que la solución candidata pueda ser añadida al archivo de soluciones no dominadas es que domine a alguna otra solución que esté en ese momento almacenada en el archivo. Si se cumple lo

anterior, la solución candidata es agregada y la solución dominada es eliminada del archivo de soluciones y se debe continuar con la revisión de la solución candidata frente a las otras que podrían también ser eliminadas del archivo.

Algoritmo 7 *paretoFront(sol_current, solsDom)*

```

1: flag  $\leftarrow$  1
2: for sol_dom in solsDom do
3:   if sol_current dominate sol_dom then
4:     solsDom.delete(sol_dom)
5:     solsDom.push(sol_current)
6:   else if sol_dom dominate sol_current then
7:     stopFor
8:     flag  $\leftarrow$  0
9:   end if
10: end for
11: if flag == 1 then
12:   solsDom.push(sol_current)
13: end if
14: return solsDom

```

En el algoritmo 7, se detalla el cómo funciona el almacenamiento de las soluciones no dominadas. El algoritmo recibe como parámetro la solución (*sol_current*) y *solsDom*, que corresponde a la lista que almacena el conjunto de soluciones no dominadas. En ella se verificará si es que la solución candidata puede entrar a la lista *solsDom*. El algoritmo recorre la lista usando la variable *sol_dom*. En caso de que *sol_current* domine a *sol_dom*, se debe eliminar ésta última de la lista de las soluciones no dominadas. Y se agrega la solución *sol_current* a la lista. En caso de que *sol_current* sea dominada por *sol_dom*, se debe terminar el ciclo, ya que es imposible que *sol_dom* entre a la lista debido a que es una solución dominada por otra. Y como último caso, si *sol_current* no domina, ni es dominada por ninguna solución de *solsDom*, el algoritmo entrará a la condición de la línea 11 y se añadirá *sol_current* al conjunto de soluciones no dominadas.

3.4. Resumen

Para resolver el problema multi-objetivo de distribución de materiales peligrosos en este capítulo se presenta un acercamiento heurístico que combina utilizar el algoritmo Greedy + Tabu Search. La representación de la solución utilizada es una lista de listas de valores enteros, en donde cada lista interna representa a las rutas de los camiones y dentro de estas listas, están representados los identificadores de los clientes visitados por el camión. Cabe recordar que en la representación de la solución, no se almacena el nodo depósito, pero el camión siempre debe comenzar y terminar su ruta en el nodo depósito. La función que mide la calidad de la solución, relaciona los dos objetivos del problema, que son minimizar los riesgos y minimizar los costos de las rutas. Esta relación se hace a través del parámetro α , el cual determina que tan importante es un objetivo por sobre otro. El algoritmo Greedy es utilizado para construir la solución inicial del problema. El algoritmo Greedy utiliza una función miope que busca obtener el mejor camino entre un par de clientes en términos de distancia. Para esto utiliza el algoritmo Dijkstra, el cual permite calcular el camino con menor costo entre cada par de clientes en una red de nodos que describe una ciudad completa. Una vez obtenida la solución inicial utiliza el algoritmo Tabu Search para la búsqueda local. Este algoritmo es utilizado junto a dos movimientos 2-OPT e Insert. Estos operadores son utilizados para recorrer el espacio de búsqueda de manera de encontrar soluciones de buena calidad para cada combinación de ponderaciones de los objetivos del problema. A estos operadores se le suma el uso de una lista tabú, que permite evitar visitar soluciones ya revisadas. Finalmente por cada solución obtenida, después de realizar los movimientos y también incluyendo la solución inicial construida por Greedy, éstas deben verificarse como candidatas a formar parte del conjunto de soluciones no dominadas.

Capítulo 4

Experimentos y resultados

En este capítulo se presentan los experimentos y resultados realizados para validar el algoritmo de búsqueda heurística propuesto. Para esto se considera un conjunto de siete instancias de pruebas construidas a partir de instancias reales de transporte de materiales peligrosos en Santiago, Chile.

El capítulo está organizado comenzando con los objetivos del proceso de validación. A continuación se presentan las instancias de prueba que se utilizan en los experimentos. Luego se explican los indicadores de calidad para evaluar las soluciones obtenidas. Se presenta el proceso de sintonización de parámetros realizado. Por último se presentan los resultados obtenidos en cada una de las instancias de prueba consideradas.

4.1. Objetivos

En este trabajo de título se consideran los siguientes objetivos de experimentación:

- Determinar si el algoritmo propuesto es capaz de obtener soluciones de calidad para las instancias del problema estudiadas.
- Comparar la calidad de los resultados y tiempos de cómputo del algoritmo propuesto

con los resultados obtenidos al resolver el problema con un algoritmo de resolución completo.

4.1.1. Instancias de Prueba

Para la validación del algoritmo propuesto se utilizó el conjunto de instancias de prueba propuesto en [9]. Dichas instancias de pruebas fueron construidas basadas en un caso real de recolección de materiales peligrosos de 167 fábricas ubicadas en la ciudad de Santiago, Chile.

En su estudio los autores dividen las 167 fábricas en 7 zonas geográficas diferentes. Cada una de las zonas genera su propia instancia de prueba. Así, es posible evaluar el algoritmo en casos de estudio más pequeños. Esta división geográfica considerada es frecuentemente usada por los distribuidores de productos y servicios en la ciudad de Santiago. Esto, debido a que los vecindarios agrupados en dichas zonas se rigen por regulaciones municipales similares.

La tabla 4.1 lista las características de las instancias de prueba consideradas en el proceso de evaluación. Para cada una de las zonas consideradas $\#Nsize$ representa la cantidad de clientes considerados en cada caso en particular. Las columnas A , B , C , D , E muestran, respectivamente, la cantidad de clientes que producen cada tipo de desecho. Cada zona tiene dos camiones con suficiente capacidad para transportar los desechos de cada subdivisión de compatibilidades. En la tabla $\#Camiones$ muestra la cantidad de camiones asignados a cada instancia, cada uno con 40,000 de capacidad de transporte.

La tabla 4.2 muestra las principales características de las instancias de prueba consideradas. Para cada una de las zonas se muestra la cantidad de material producido en total por todos los clientes de cada zona. En esta tabla las columnas A , B , C , D y E muestran la cantidad de cada desecho tipo producido en cada una de las zonas.

Como es posible observar, la cantidad de clientes varía entre 11 para la zona 7 y 36 para la zona 2. Todos los casos de prueba considerados requieren la recolección de los cinco tipos de materiales peligrosos definidos. La zona 2 es la que más material peligroso produce con

Instancia	#Nsize	A	B	C	D	E	#Camiones
zona 1	32	7	7	6	6	6	2
zona 2	36	7	7	8	7	7	2
zona 3	15	3	3	3	3	3	2
zona 4	30	6	6	6	6	6	2
zona 5	21	4	4	4	5	4	2
zona 6	22	5	4	4	4	5	2
zona 7	11	2	3	3	1	2	2

Tabla 4.1: Características de instancias de pruebas obtenidas de [9]

Instancia	Total	A	B	C	D	E
zona 1	27,340	7,640	6,680	3,040	4,260	5,720
zona 2	63,030	5,440	10,890	14,420	20,540	11,740
zona 3	20,470	8,040	2,170	4,170	4,120	1,970
zona 4	27,170	12,440	1,840	3,560	5,940	3,390
zona 5	13,320	1,560	4,370	1,820	2,760	2,810
zona 6	21,010	5,600	3,270	6,060	2,310	3,770
zona 7	12,220	3,840	2,490	2,170	1,490	2,230

Tabla 4.2: Material producido por los clientes en las instancias de prueba obtenidas de [9]

un total de 63,030 mientras que la zona 5 es la que menos material produce con un total de 13,320. Debido a esto, se podría intuir que la instancia más compleja es la correspondiente a la zona 2, seguida por las instancias de las zonas 1 y 4. Ambas con una cantidad similar de clientes, pero menor cantidad de material producido comparadas con la zona 2. Las zonas 3, 6, 7 y 5 probablemente tengan una complejidad considerablemente menor dada la cantidad de nodos que poseen.

A partir de la tabla de compatibilidades del Capítulo 1 (sección 1.1), es posible observar que en todas las instancias se requiere dividir las cargas que se asignan a los camiones. Los clientes que producen material peligroso de tipo A deben entregarse en conjunto con alguna

combinación de materiales compatibles con A (ya sean C y/o D). En los otros camiones deberán agruparse las visitas a clientes que producen materiales tipo B y E combinados con materiales compatibles con B y E (pueden ser C y/o D).

4.1.2. Indicadores de calidad

La evaluación de calidad de los resultados de un algoritmo multi-objetivo requiere evaluar siempre un conjunto de soluciones al problema. Para esto es necesario utilizar indicadores que permitan medir, para el conjunto de soluciones tanto su convergencia al frente de Pareto real del problema como su distribución sobre éste.

Como indicador de calidad del conjunto de soluciones obtenido, en este trabajo de título se utilizó el indicador de hipervolumen. El indicador fue propuesto por [28]. El indicador de hipervolumen permite medir (hiper-) espacio dominado por el conjunto de soluciones obtenido respecto a un punto de referencia en el espacio. A mayor valor de hipervolumen, mejor aproximación del respectivo frente de Pareto.

La figura 4.1 muestra un ejemplo de un problema donde se minimizan dos objetivos y se obtienen tres soluciones que forman la aproximación del frente de Pareto. En la esquina superior derecha de la figura se muestra el punto de referencia (**R**) para el cálculo del hipervolumen. Es importante indicar que el punto de referencia elegido debe sobrepasar levemente los mayores valores de objetivos obtenidos en cada caso. Esto, de modo que todas las soluciones que se puedan obtener queden siempre cubiertos por dicho punto y puedan ser considerados en el cálculo del hipervolumen.

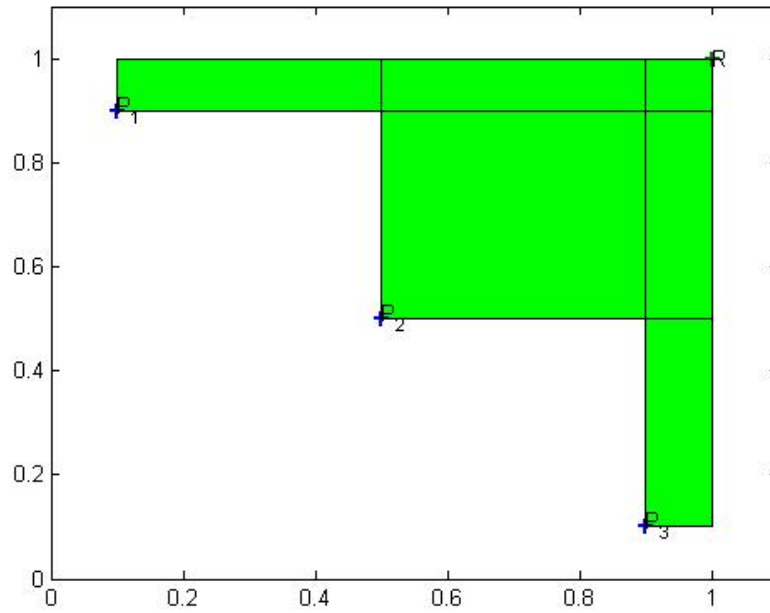


Figura 4.1: Hipervolumen para un problema de minimización de dos objetivos.

Para medir el hipervolumen en cada uno de los casos de pruebas, se consideraron los resultados obtenidos al resolver el problema con un acercamiento de resolución completo para determinar los puntos de referencia correspondientes. Para cada uno de los valores máximos encontrados por objetivo, se consideró una holgura de 10 % de modo que los puntos extremos aporten valor al indicador. Así los puntos de referencia utilizados se muestran en la tabla 4.3.

Instancia	Puntos de referencia	
	Riesgo	costo
zona 1	1,653,399	327,571
zona 2	1,544,180	204,217
zona 3	864,635	116,471
zona 4	1,140,887	147,575
zona 5	1,117,523	164,566
zona 6	707,923	114,149
zona 7	283,314	107,185

Tabla 4.3: Puntos de referencia para la evaluación del hipervolumen en cada zona.

4.1.3. Sintonización de parámetros

De manera de obtener los mejores valores de parámetros del algoritmo propuesto para cada una de las instancias a resolver y, para el conjunto completo, se realizó un proceso de sintonización de parámetros. En este trabajo de memoria se utilizó el algoritmo de sintonización de parámetros ParamILS [14].

ParamILS, corresponde a un algoritmo de búsqueda local iterada. Este proceso permite encontrar una configuración de parámetros, tal que produzca el mejor nivel de hipervolumen promedio. Para realizar este proceso, se parte con una configuración de parámetros inicial y se van generando vecindarios a partir de modificaciones a los parámetros, uno a la vez. Estas modificaciones se aceptan si se obtiene un resultado de mejor rendimiento. Las modificaciones se dejan de realizar, hasta que no se vean mejoras en el rendimiento al cambiar cualquier parámetro.

La tabla 4.4 muestra los parámetros a sintonizar, la discretización de su rango de valores posibles y el valor inicial considerado para cada uno. Cada proceso de sintonización consideró como criterio de término una cantidad máxima de ejecuciones del algoritmo propuesto. Este valor fue fijado en 1,000 ejecuciones para cada proceso de sintonización.

Los parámetros a sintonizar son, *tabu_size*, n_α y *p_insert*. El primero de ellos es utilizado para definir el tamaño de la Lista Tabú, la cual se encarga de almacenar los últimos movimientos realizados durante la búsqueda y evitar ciclos. Los valores que puede tomar *tabu_size* son 0, 0.1, 0.2 y 0.3. Estos valores se traducen en porcentajes (0 %, 10 %, 20 % y 30 %) respecto a la cantidad de clientes de la instancia a resolver. Por lo tanto, si *tabu_size* se define como 0.1 (que corresponde al valor inicial) y la cantidad total de clientes es 22, esto significa que *tabu_size* tomará el valor 2.2, pero como debe ser un valor entero, corresponde asignarle el valor 2.

El segundo parámetro a sintonizar es n_α , que corresponde a la cantidad de valores que puede el ponderador de importancia relativa de los objetivos α . Cabe recordar que este parámetro puede tomar valores entre 0 y 1, los cuales, dependiendo de la cantidad especificada, se tomaran equidistantes entre sí y consideraran siempre los pesos extremos. Por ejemplo si n_α toma el valor 11 (que corresponde al valor inicial), los posibles valores que puede tomar α son: 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 y 1.0.

El tercer parámetro a sintonizar es *p_insert*, que corresponde a la probabilidad de que se pueda escoger el movimiento insert, es decir si *p_insert* toma el valor 0.1 (que corresponde al valor inicial), significa que existe un 10 % de probabilidad de escoger el movimiento insert y por lo tanto existe un 90 % de probabilidad de escoger el movimiento 2-OPT en cada iteración del proceso de búsqueda de Tabu Search.

Nombre	Descripción	Valores posibles	Valor inicial
<i>tabu_size</i>	largo de la lista tabu	0, 0.1, 0.2, 0.3	[0.1]
n_α	valores posibles de α	{5, 11, 21}	[11]
<i>p_insert</i>	probabilidad de inserción	{0.0, 0.1, 0.25, 0.5, 0.75, 1.0}	[0.1]

Tabla 4.4: Parámetros para el proceso de sintonización

La tabla 4.5 muestra los valores de parámetros determinados por el proceso de sintonización para cada uno de los casos de prueba y para la sintonización que considera todos los casos juntos.

Instancia	<i>tabu_size</i>	<i>n_α</i>	<i>p_insert</i>
zona 1	0.2	21	0
zona 2	0.2	5	0
zona 3	0.2	11	0.5
zona 4	0.3	5	0
zona 5	0	21	0
zona 6	0.2	5	0
zona 7	0.2	5	0.1
ALL	0	5	0.25

Tabla 4.5: Configuración de parámetros sintonizada por instancia y en conjunto

4.2. Resultados

En esta sección se exponen los resultados obtenidos en cada uno de los casos de prueba previamente presentados. Utilizando los parámetros obtenidos para cada instancia y los parámetros globales obtenidos al sintonizar los parámetros usando el conjunto de instancias completo. En ambos casos se realizaron 20 ejecuciones utilizando semillas aleatorias diferentes. Para cada ejecución se calculó el hipervolumen correspondiente a partir de los putos de referencia indicados. Además, se conformó un archivo con todas las soluciones obtenidas para cada caso y se calculó el hipervolumen agregado de las 20 ejecuciones en conjunto.

Además, para cada instancia de prueba se comparan los resultados obtenidos por un acercamiento exacto y los resultados obtenidos por el algoritmo de búsqueda local propuesto. Para cada caso se muestran los frentes obtenidos, el frente de Pareto agregado conseguido por el algoritmo propuesto y el frente de Pareto conseguido por el método de resolución exacta.

Por último, se incorpora la información de los tiempos totales de ejecución de ambos acercamientos de resolución.

4.2.1. Zona 1

La figura 4.2 muestra los frentes de Pareto agregados obtenidos por el acercamiento heurístico propuesto para la instancia correspondiente a la zona 1. En el gráfico se muestran en azul las soluciones obtenidas con el algoritmo usando la sintonización de parámetros obtenida para la instancia de la zona 1 en particular y en color rojo se presentan los resultados obtenidos usando la sintonización de parámetros que consideró todas las instancias para el proceso de entrenamiento. El eje x muestra el riesgo total de las soluciones y el eje y la distancia total. El frente obtenido con la sintonización zona 1 está conformado por un total de 42 soluciones mientras que el frente obtenido por la configuración ALL posee 84 soluciones. Esto indica que la configuración ALL para la zona 1 permite diversificar mejor el espacio de soluciones en este caso, esto a pesar de que ambos conjuntos agregados poseen soluciones dominadas. Es muy curioso que la sintonización ALL haya logrado una mayor diversificación, ya que, comparando esta configuración de parámetro con la sintonización zona 1, tiene mucho más en contra. Para el caso de *tabu_size*, para zona 1 el valor es 0.2 y para ALL no usa lista tabu, por lo que con la última sintonización es mucho más probable que el algoritmo quede atrapado en ciclos y esto impida diversificar. Lo mismo ocurre con n_{α} , el valor de zona1 es aproximadamente cuatro veces mayor que el utilizado en la configuración ALL, en este caso la sintonización con zona 1 también se ve favorecida. Y para *p_insert*, también se ve favorecida la configuración zona 1, ya que si este parámetro es 0, no hay posibilidad de escoger el movimiento *p_insert* y solo se utiliza el 2-OPT. Como el movimiento insert está muy acotado a la condición de compatibilidad por la cantidad de camiones (que es 2), es muy difícil que se logre diversificar utilizando este movimiento. La única explicación que puede existir, es que la cantidad de clientes es muy grande y esto permite diversificar aún más utilizando el movimiento 2-OPT. Por lo que la configuración de parámetros no tuvo un impacto en esta instancia.

Es interesante destacar la distribución de soluciones obtenidas que presentan un agrupamiento hacia soluciones que minimizan los riesgos, muy pocas en la zona central y algunas otras soluciones que muestran valores pequeños de distancia total.

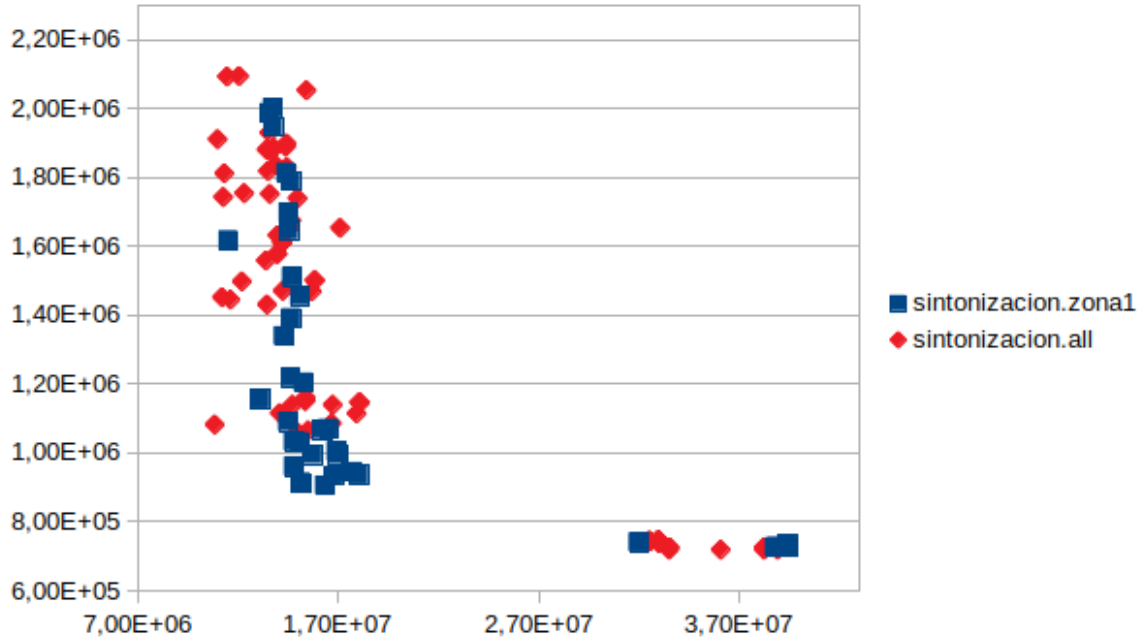


Figura 4.2: Frente de Pareto heurístico para ambas configuraciones: zona1 (en azul) y ALL (en rojo) para la zona 1.

La tabla 4.6 muestra el resumen de las hipervolumenes conseguidos en las diferentes ejecuciones del algoritmo con cada configuración de parámetros. Se muestra en cada caso el hipervolumen mínimo, máximo y el hipervolumen promedio de las ejecuciones. A partir de esto es posible observar que los hipervolumenes mínimo y promedio utilizando la configuración ALL, son mayores que los que se obtienen usando la configuración zona1. Y para el caso de hipervolumen máximo, utilizando la configuración zona1 se obtiene un mejor valor que usando la configuración ALL.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona 1	2.06E+13	3.23E+13	2.69E+13
ALL	2.28E+13	3.18E+13	2.74E+13

Tabla 4.6: Valores de hipervolumen de ambas configuraciones para la zona 1.

Se decidió utilizar la medida hipervolumen promedio para escoger entre una de las dos configuraciones. Por lo tanto, se utilizarán las soluciones obtenidas por la configuración ALL para compararlas con los resultados obtenidos en la misma instancia de prueba usando el

solver CPLEX.

La figura 4.3 muestra los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento heurístico utilizando la configuración ALL (en color rojo). A partir de este gráfico es evidente que el algoritmo heurístico no es capaz de acercarse al frente de Pareto óptimo obtenido por el solver. El tiempo de ejecución total del algoritmo completo corresponde a la suma del tiempo que toma resolver el problema 11 veces. En este caso ese tiempo corresponde a 100,070 segundos. La heurística toma un tiempo de alrededor de 650 segundos por ejecución. En este caso no es posible comparar los hipervolumenes dado que no se puede usar el mismo punto de referencia en los dos frentes.

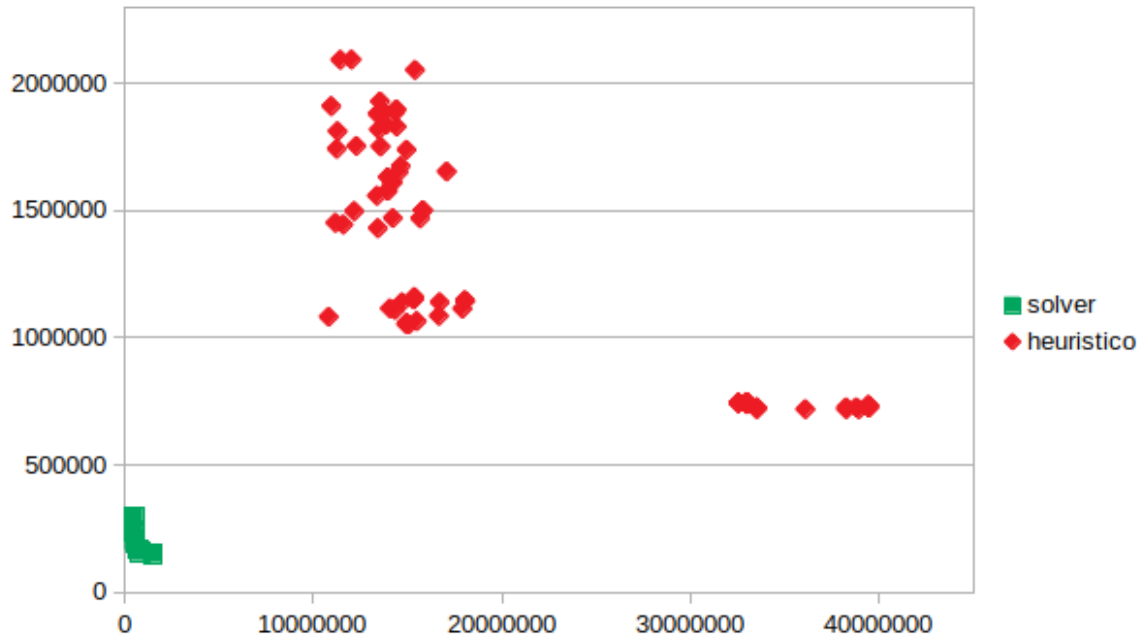


Figura 4.3: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en rojo) para la zona 1.

4.2.2. Zona 2

La figura 4.4 muestra los frentes de Pareto agregados que se obtuvieron usando el acercamiento heurístico propuesto en la instancia correspondiente a la zona 2. Al igual que en

la figura 4.2, se muestran en azul las soluciones obtenidas con el algoritmo usando la sintonización de parámetros zona 2 y en color rojo se presentan los resultados obtenidos con la sintonización ALL que se obtuvo en el proceso de sintonización que usa todas las instancias. El frente obtenido con la sintonización zona 2 tiene un total de 13 soluciones mientras que el frente obtenido por la configuración ALL contiene 28 soluciones. Por lo tanto, utilizando la configuración ALL, el algoritmo fue capaz de diversificar mucho más que usando la otra configuración. De todas maneras, los conjuntos de soluciones muestran, nuevamente, una tendencia muy similar, agrupándose en su mayoría en soluciones de bajo riesgo total y algunas otras pocas soluciones de muy baja distancia total en la esquina inferior derecha. Para la cantidad de soluciones de esta instancia, ocurre algo muy similar a la instancia zona 1. Con la configuración ALL se obtuvo una diversificación mucho mayor. Si bien con la sintonización zona 2, se debió haber visto favorecida por las mismas razones que el caso anterior (exceptuando el valor de n_{α} , que para ambas configuraciones de la zona 2, se tiene el mismo valor). Pero cabe destacar que las zonas 1 y 2 son las más grandes, con 32 y 36 clientes respectivamente. Por lo que se puede apoyar la idea de que si la cantidad de clientes de la instancia es muy grande, no se ve impactada por la configuración de los parámetros.

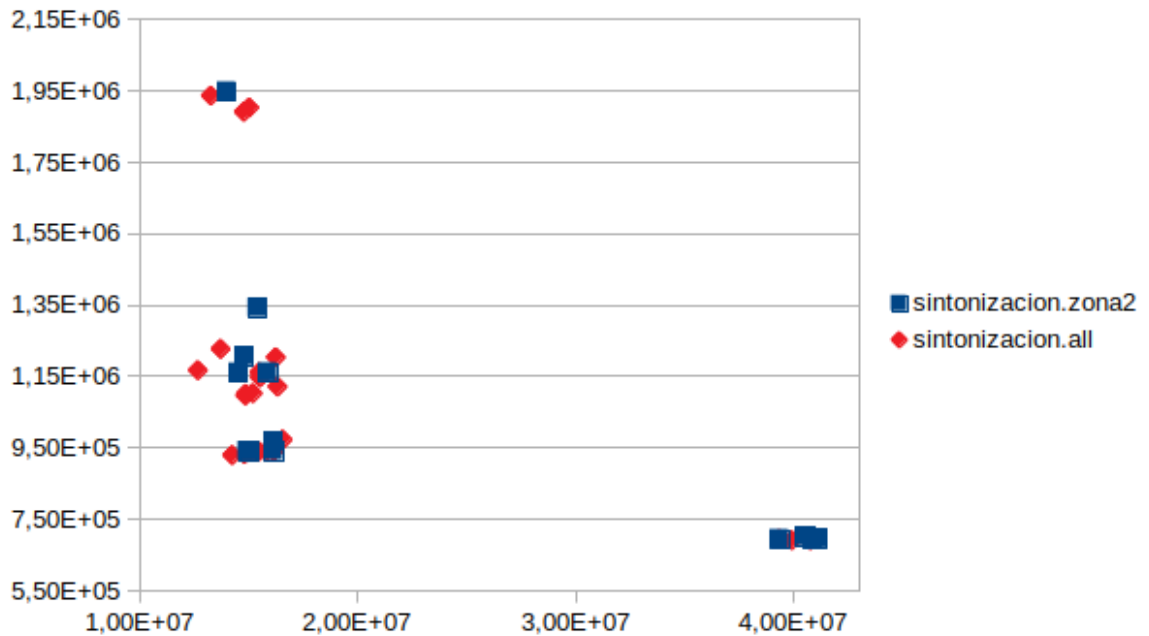


Figura 4.4: Frente de Pareto heurístico para ambas configuraciones, zona2 (en azul) y ALL (en rojo) para la zona 2.

La tabla 4.7 muestra el resumen de los hipervolumenes de las diferentes ejecuciones del algoritmo utilizando las configuraciones zona 2 y ALL. Para ambos casos, se muestran hipervolumen mínimo, máximo e hipervolumen promedio de 20 ejecuciones. Es posible observar que el hipervolumen promedio de la configuración zona 2 es mayor el de la configuración ALL. Esto a pesar de que posee menos soluciones. Así, en este caso se compararán los resultados de la técnica de resolución exacta con los obtenidos en el algoritmo del presente trabajo de memoria, utilizando la configuración zona 2.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona 2	2.57E+13	3.49E+13	2.97E+13
ALL	1.07E+13	3.57E+13	2.58E+13

Tabla 4.7: Valores de hipervolumen de ambas configuraciones para la zona 2.

En la figura 4.5 se muestran los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento heurístico utilizando la configuración zona 2 (en color azul). Al igual que en el caso anterior, el algoritmo heurístico no es capaz de acercarse al frente de Pareto óptimo del solver. El tiempo de ejecución del acercamiento exacto fue de 100,077.44 segundos para todas las ejecuciones. Para el caso del acercamiento heurístico fue mucho menor cercano a 10,000 segundos. De todas maneras, no es posible comparar los hipervolumenes, debido a que no es posible utilizar el mismo punto de referencia en ambos frentes propuestos.

4.2.3. Zona 3

En la figura 4.6 se muestran los frentes de Pareto agregados por el algoritmo de búsqueda local. Se observan de color azul las soluciones correspondientes a la sintonización de la instancia (zona3) y en rojo los resultados correspondientes a la sintonización ALL. El frente que corresponde a la sintonización zona3 está conformado por 73 soluciones, mientras que el frente correspondiente a la configuración ALL, tiene solo 54 soluciones. Esto indica que la configuración específica para la zona 3 permite diversificar mejor el espacio de soluciones en este caso, esto a pesar de que ambos conjuntos agregados poseen soluciones dominadas. Para

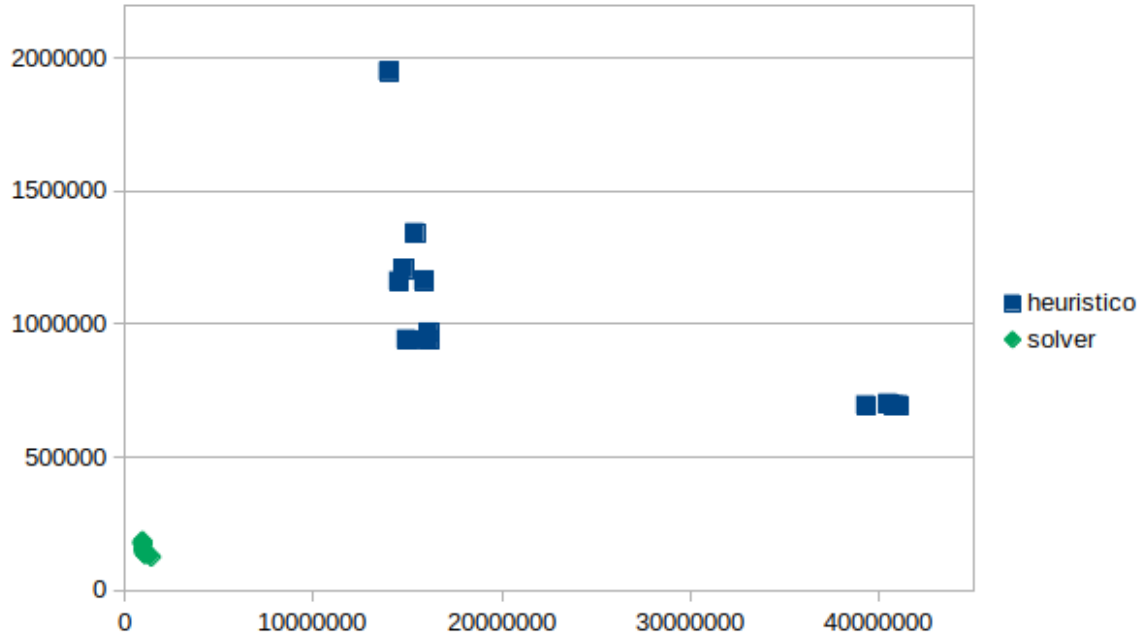


Figura 4.5: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 2.

esta instancia, la cantidad de clientes es mucho menor (15), comparadas con las zonas 1 y 2 (32 y 36 respectivamente), por lo que es muy probable que la configuración de parámetros si impacte en la diversificación. Como n_{α} en ambas configuraciones es igual, se ignorará este parámetro. Para $tabu_size$, el valor de la sintonización zona3 es 0.2 y para la configuración ALL es 0, por lo que en la última configuración es muy probable que el algoritmo se quede atrapados en ciclos, ya que para ese caso no existe lista tabú que lo pueda impedir. Comparando los valores p_insert , para la configuración zona3 es de 0.5 y para ALL es 0.25, por lo que la sintonización ALL se ve favorecida por sobre la zona 3. Esto puede explicar porque la diferencia entre la cantidad de soluciones no es tan grande. Puede parecer interesante la idea de que con un menor valor de p_insert para la sintonización zona 3 la diferencia entre la cantidad de soluciones sea mayor. Por otro lado, en la misma figura es posible observar que las soluciones obtenidas usando la configuración de parámetros zona 3 muestran mejor convergencia en la mayoría de las áreas del frente cubiertas.

La tabla 4.8 muestra los hipervolumenes mínimo, máximo y promedio correspondientes de cada configuración del algoritmo propuesto en este trabajo de memoria. Se observa que

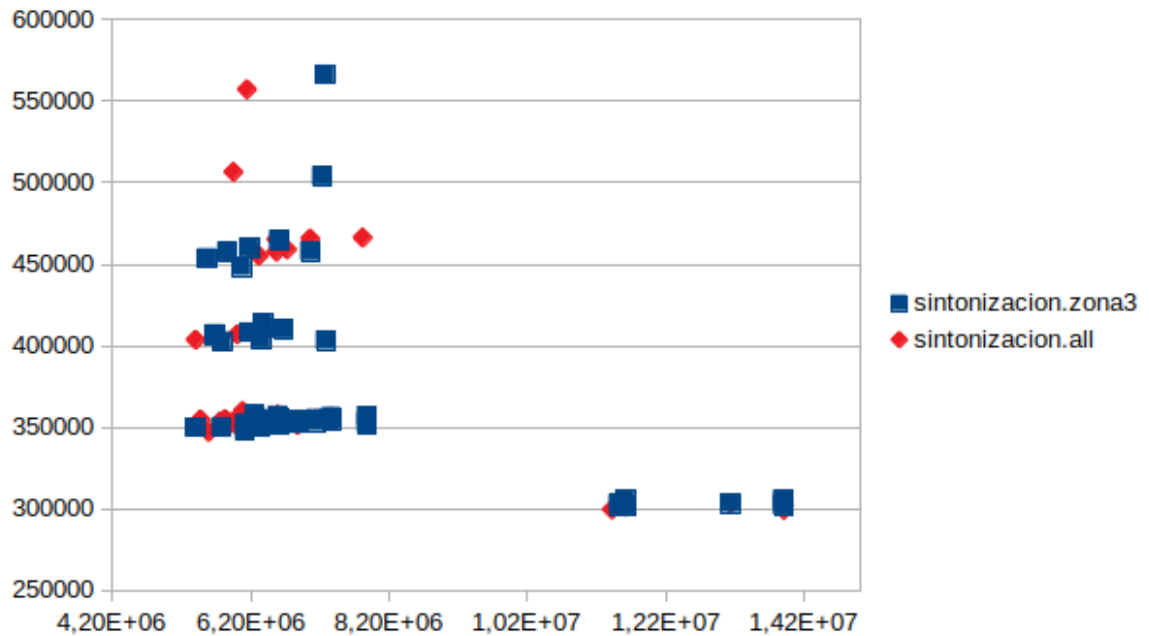


Figura 4.6: Frente de Pareto heurístico para ambas configuraciones, zona3 (en azul) y ALL (en rojo) para la zona 3.

los frentes obtenidos utilizando la configuración zona3 son siempre mayores que los que se obtienen usando la configuración ALL. Esto permite validar la necesidad de configuraciones distintas en diferentes instancias de pruebas. En este caso, para una instancia de pequeño tamaño (16 clientes) resulta mejor utilizar la variable una lista *tabu_size* de tamaño 0.2 versus no usarla. Utilizar una mayor cantidad de valores α (11 por sobre 5) que permite repetir la búsqueda para un mayor conjunto de ponderaciones y utilizar una mayor cantidad de veces el operador de inserción *p_insert* (0.5 por sobre 0.25).

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona 3	1.65E+12	4.14E+12	2.89E+12
ALL	1.26E+12	3.8E+12	2.73E+12

Tabla 4.8: Valores de hipervolumen de ambas configuraciones para la zona 3.

Así entonces, se utilizarán las soluciones obtenidas por la configuración zona 3 para compararlas con los resultados obtenidos usando acercamiento exacto. La figura 4.7 muestra los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento

heurístico utilizando la configuración zona 3 (en color azul).

Al igual que en los casos anteriores, el algoritmo heurístico no es capaz de acercarse a los resultados obtenidos por el solver. A pesar de la cantidad de soluciones obtenidas por el acercamiento en un amplio rango del espacio objetivo, estas soluciones presentan importantes problemas de convergencia respecto a lo que se considera el conjunto óptimo de Pareto. El tiempo de ejecución para el acercamiento exacto corresponde a 458.54 segundos. La heurística toma un tiempo de alrededor de 400.2 segundos por ejecución. Al igual que los casos anteriores, no es posible comparar los hipervolumenes dado que no se puede usar el mismo punto de referencia en los dos frentes.

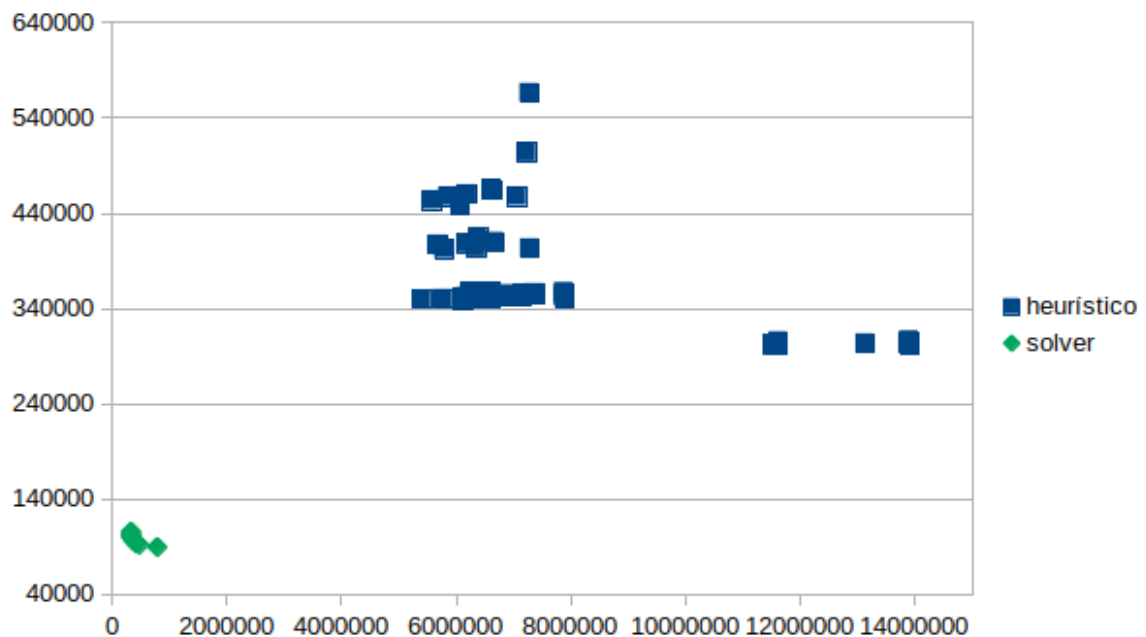


Figura 4.7: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 3.

4.2.4. Zona 4

En la figura 4.8, se observan los frentes de Pareto agregados por ambas configuraciones del acercamiento heurístico. En color azul se muestran las soluciones usando la sintonización zona 4 y en color rojo la sintonización ALL. El frente correspondiente a la sintonización zona

4, en este caso está compuesto por 62 soluciones, mientras que el otro frente contiene solo 51 soluciones. Al igual que en el caso anterior, la sintonización de la instancia permitió una mayor diversificación. Esto se puede deber a que en ambas sintonizaciones se utiliza una lista tabu mientras que la configuración ALL no la utiliza. Por lo que es probable que con aquellas sintonizaciones, en ambos casos, se evita la generación de ciclos que puede ser más frecuente en problemas de menor tamaño.

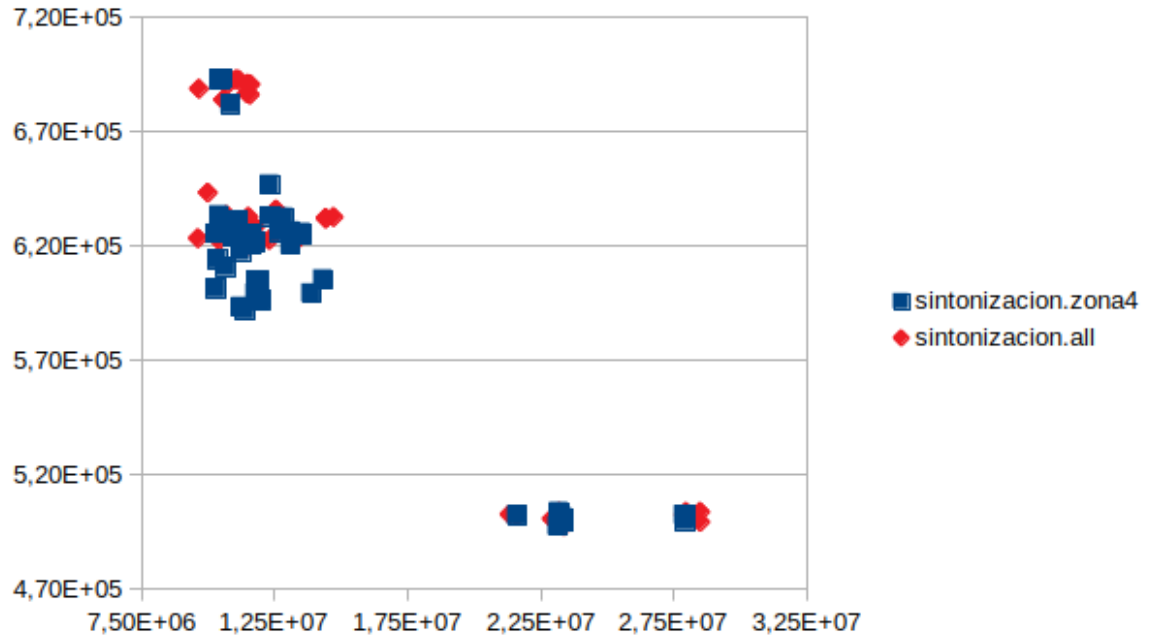


Figura 4.8: Frente de Pareto heurístico para ambas configuraciones, zona4 (en azul) y ALL (en rojo) para la zona 4.

La tabla 4.9 muestra el hipervolumen mínimo, máximo y promedio obtenidos por ambos frentes agregados. Acá es posible observar que si se evalúa el hipervolumen promedio, los frentes obtenidos utilizando la configuración zona 4 son mayores que los que se obtienen usando la configuración ALL.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona4	9.03E+12	1.13E+13	9.81E+12
ALL	4.95E+12	1.15E+13	9.00E+12

Tabla 4.9: Valores de hipervolumen de ambas configuraciones para la zona 4.

Se utilizarán entonces estas soluciones para compararlas con los resultados obtenidos por el solver. La figura 4.9 muestra los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento heurístico utilizando la configuración zona 4 (en color azul). Al igual que en los casos anteriores, se visualiza que el algoritmo heurístico no es capaz de acercarse al frente de Pareto óptimo obtenido por el solver. También existen problemas de convergencia a lo que se considera el conjunto óptimo de Pareto.

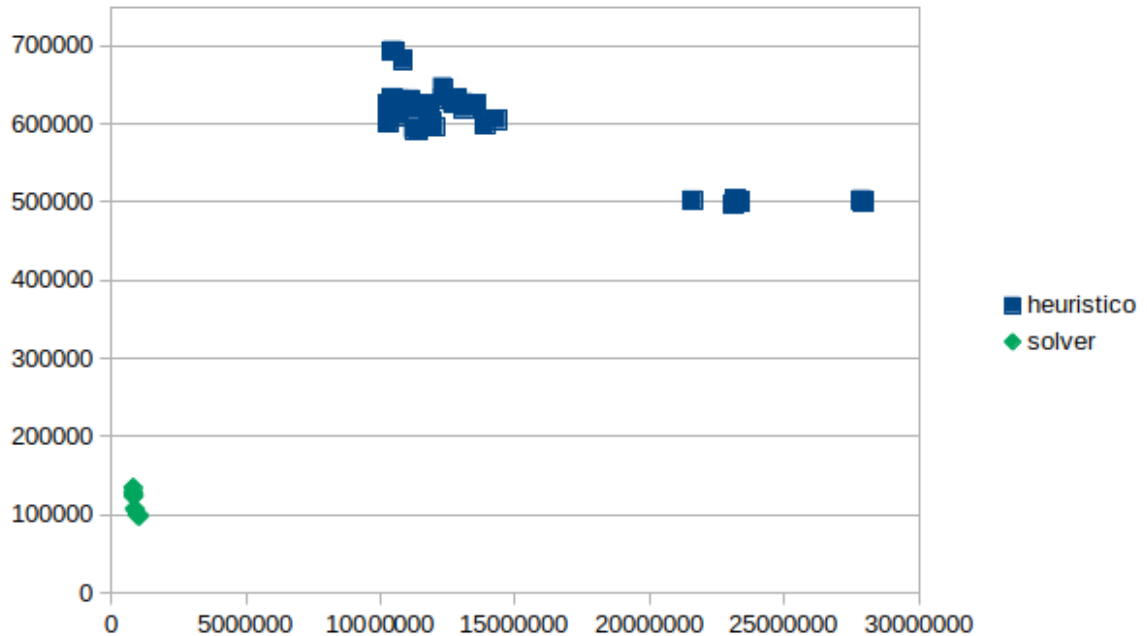


Figura 4.9: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 4.

El tiempo de ejecución total que le tomó al solver encontrar todos los puntos del frente fue de 6,105.67 segundos y para el acercamiento heurístico fue de 6,200 segundos. Tampoco es posible comparar los hipervolumenes, debido a que no se puede usar el mismo punto de referencia en los dos frentes.

4.2.5. Zona 5

En la figura 4.10, se muestran los frentes de Pareto agregados para el acercamiento heurístico con ambas configuraciones. Representados por el color azul están los resultados de la sintonización zona 5 y en color rojo las soluciones de la sintonización ALL. El frente obtenido por la sintonización zona 5, está conformado por 125 soluciones, mientras que el otro frente contiene solo 71 soluciones. A simple vista, se puede observar que las soluciones obtenidas usando la configuración de parámetros zona 5 muestra una mejor convergencia a los que es el frente de Pareto. A pesar de que en ambos casos (sintonización zona5 y ALL) *tabu_size* es 0, la mayor cantidad de soluciones obtenidas por la configuración zona5, se puede deber a que el parámetro n_{α} es mayor que el de la sintonización ALL (21 por sobre 5). Lo que permite una mayor búsqueda de soluciones. También cabe destacar que p_{insert} es 0 con la sintonización zona 5, por lo tanto no se usa este movimiento que está muy condicionado si es que la cantidad de camiones del caso es dos. Así que debido al no uso del movimiento insert, el algoritmo pudo haber intensificado más.

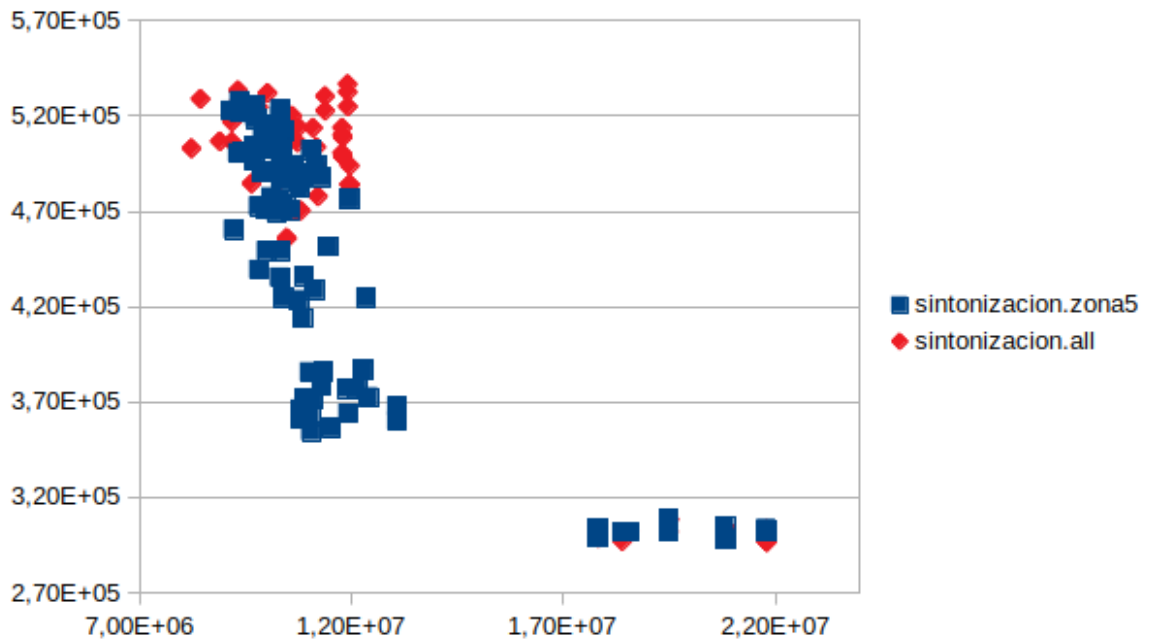


Figura 4.10: Frente de Pareto heurístico para ambas configuraciones, zona5 (en azul) y ALL (en rojo) para la zona 5.

En la tabla 4.10 se muestran los hipervolumen mínimo, máximo y promedio. Se observa que el hipervolumen promedio utilizando la configuración zona5 es mayor que utilizando la configuración ALL. Por lo tanto se utilizarán los resultados obtenidos por la configuración zona 5 para compararlas con los resultados obtenidos por el solver.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona5	3.82E+12	5.05E+12	4.64E+12
ALL	3.41E+12	5.40E+12	4.58E+12

Tabla 4.10: Valores de hipervolumen de ambas configuraciones para la zona 5.

La figura 4.11 muestra los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento heurístico utilizando la configuración zona 5 (en color azul). Al igual que en los casos anteriores, es evidente que el algoritmo heurístico no es capaz de acercarse al frente de Pareto óptimo obtenido por el solver. El tiempo de ejecución total del algoritmo completo corresponde a 17,022.82 segundos, mientras que para el acercamiento heurístico fue de solo 3,500 segundos. En este caso no es posible comparar los hipervolumenes dado que no se puede usar el mismo punto de referencia en los dos frentes.

4.2.6. Zona 6

En la figura 4.12 se muestran los frentes de Pareto agregados obtenidos por el algoritmo propuesto en el presente trabajo de memoria. En color azul se muestran las soluciones obtenidas utilizando la configuración zona 6 y en color rojo usando la sintonización ALL. El frente de la sintonización zona 6, contiene 86 soluciones y la otra configuración tiene 87 soluciones. La cantidad de soluciones es muy similar y se puede deber a que en ambas configuraciones, el parámetro n_α tiene el mismo valor (5), lo que puede inducir en una misma cantidad de iteraciones del algoritmo para ambas configuraciones. En ambos casos nuevamente se observa un elevado número de soluciones en el sector superior izquierdo (aquellas que minimizan el riesgo total) y algunas otras pocas soluciones en el sector inferior derecho (aquellas que minimizan la distancia total). El algoritmo no es capaz de encontrar soluciones en el área central del frente en este caso.

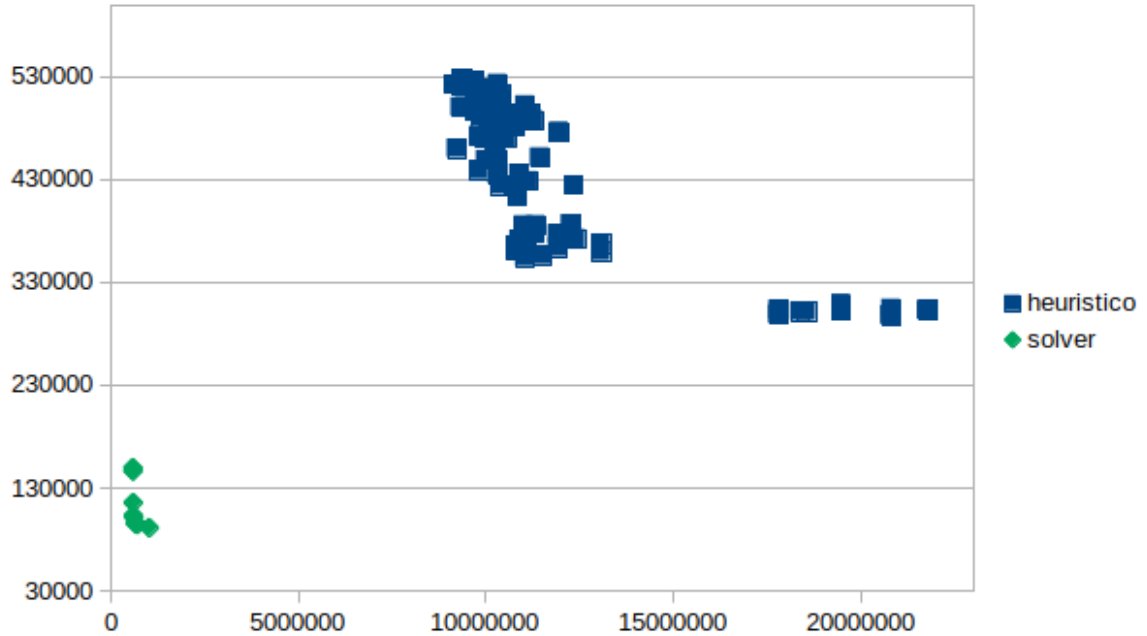


Figura 4.11: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 5.

En la tabla 4.11 se presentan los valores de hipervolumen para ambas configuraciones. A partir de esto es posible observar que los frentes obtenidos utilizando la configuración zona 6 son levemente mayores (comparando el promedio) que los obtenidos por la sintonización ALL.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona6	1.19E+12	1.49E+12	1.29E+12
ALL	9.35E+11	1.51E+12	1.28E+12

Tabla 4.11: Valores de hipervolumen de ambas configuraciones para la zona 6.

Las soluciones de la zona 6 se comparan con los resultados obtenidos en la misma instancia de prueba usando el solver CPLEX en la figura 4.13. Los resultados obtenidos por el acercamiento exacto se muestran en color verde y los obtenidos por el acercamiento heurístico se muestran en color azul. Al igual que en los cinco casos anteriores, el algoritmo heurístico no es capaz de acercarse al frente de Pareto óptimo obtenido por el solver. Y también se observa un problema de convergencia al conjunto óptimo de Pareto. El tiempo total de ejecución del

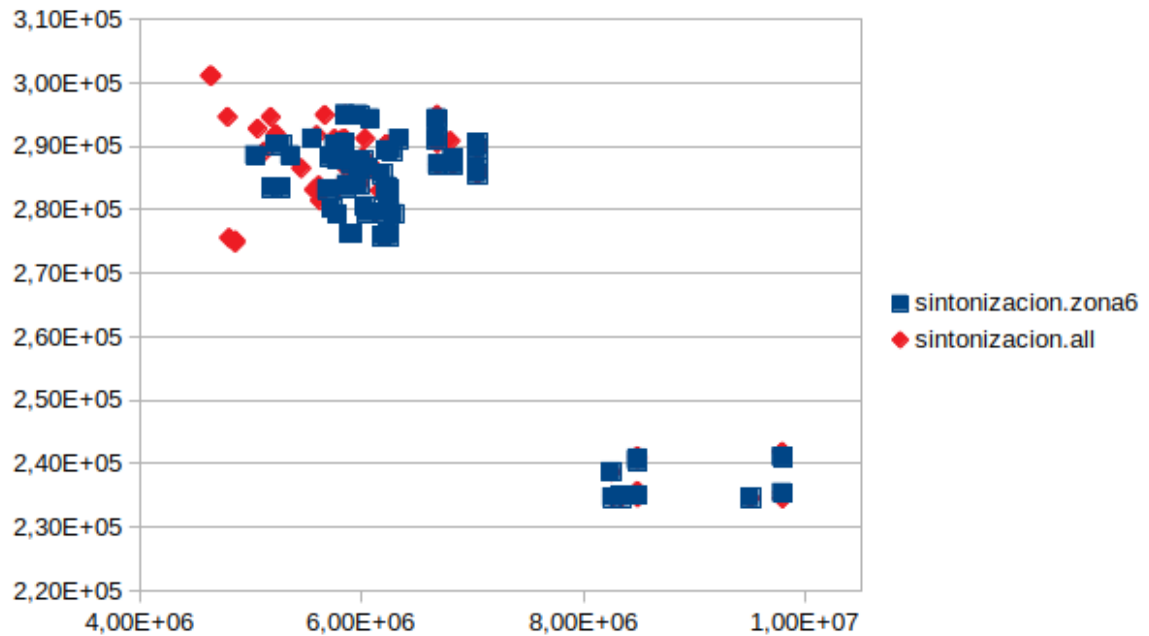


Figura 4.12: Frente de Pareto heurístico para ambas configuraciones, zona6 (en azul) y ALL (en rojo) para la zona 6.

algoritmo completo fue de 525.19 segundos y para el acercamiento heurístico fue de 3,080 segundos. Tampoco es posible comparar los hipervolumenes dado que no se puede usar el mismo punto de referencia en los dos frentes.

4.2.7. Zona 7

En la figura 4.14, se muestran los frentes de Pareto agregados obtenidos utilizando ambas configuraciones. En color azul están los resultados obtenidos por la sintonización zona 7 y en color rojo usando la configuración ALL. La cantidad de soluciones obtenidas por la sintonización zona 7 fue de 64 soluciones y para la configuración ALL 46 soluciones. Al igual que en algunos casos anteriores, la diferencia está en los valores seleccionados para los parámetros *tabu_size* y *p_insert*. Como *tabu_size* en ALL es 0, es muy probable que caiga en ciclos y esto impide diversificar el espacio de soluciones. Mientras que por las restricciones del problema, utilizar el movimiento *p_insert* no es muy favorable, ya que en una gran cantidad de casos, el movimiento no se podrá aplicar por compatibilidades entre

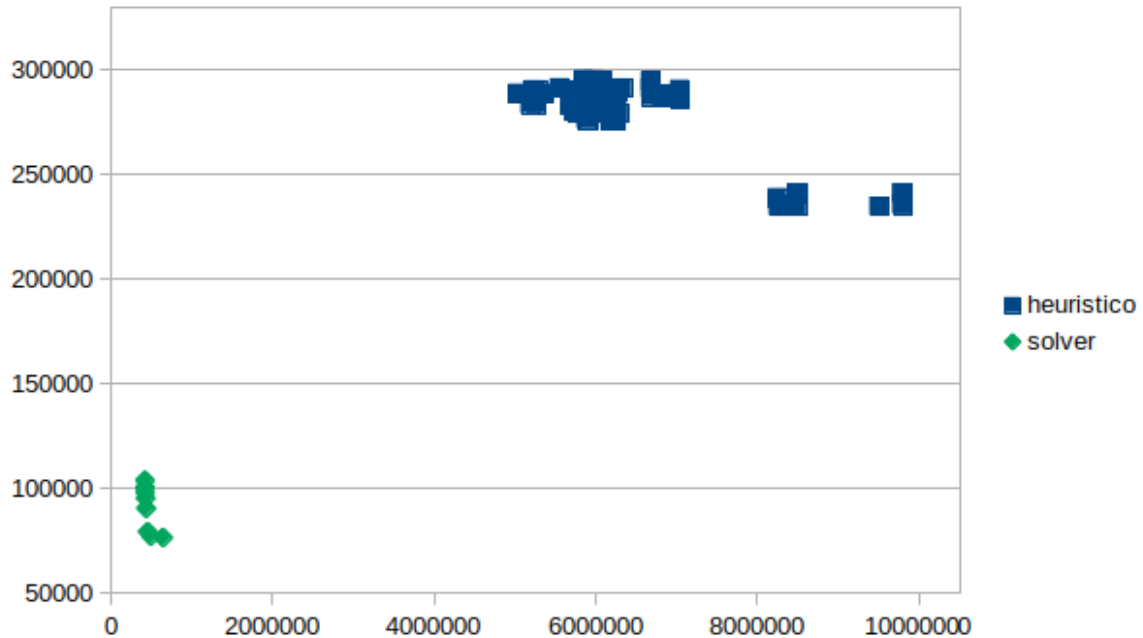


Figura 4.13: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 6.

los materiales, esto principalmente se debe a que existen solo dos camiones para utilizar en todas las zonas.

En la tabla 4.12 se muestran los resultados de hipervolumen mínimo, máximo y promedio para ambas configuraciones de parámetros. Es posible observar que los frentes obtenidos utilizando la sintonización zona 7 son mayores que los que se obtienen usando la configuración ALL. Es por esto que las soluciones obtenidas con la sintonización de parámetros zona 7, se compararán con las soluciones obtenidas por el solver. Cabe destacar que en 6 de las 7 zonas, se escogió comparar los resultados de la configuración de la instancia por sobre la configuración de todas las instancias. Esto valida la necesidad de configuraciones diferentes en distintas instancias de pruebas.

La figura 4.15 muestra los frentes de Pareto obtenidos por el acercamiento exacto (en color verde) y por el acercamiento heurístico utilizando la configuración zona 7 (en color azul). A partir de este gráfico es evidente que el algoritmo heurístico no es capaz de acercarse al

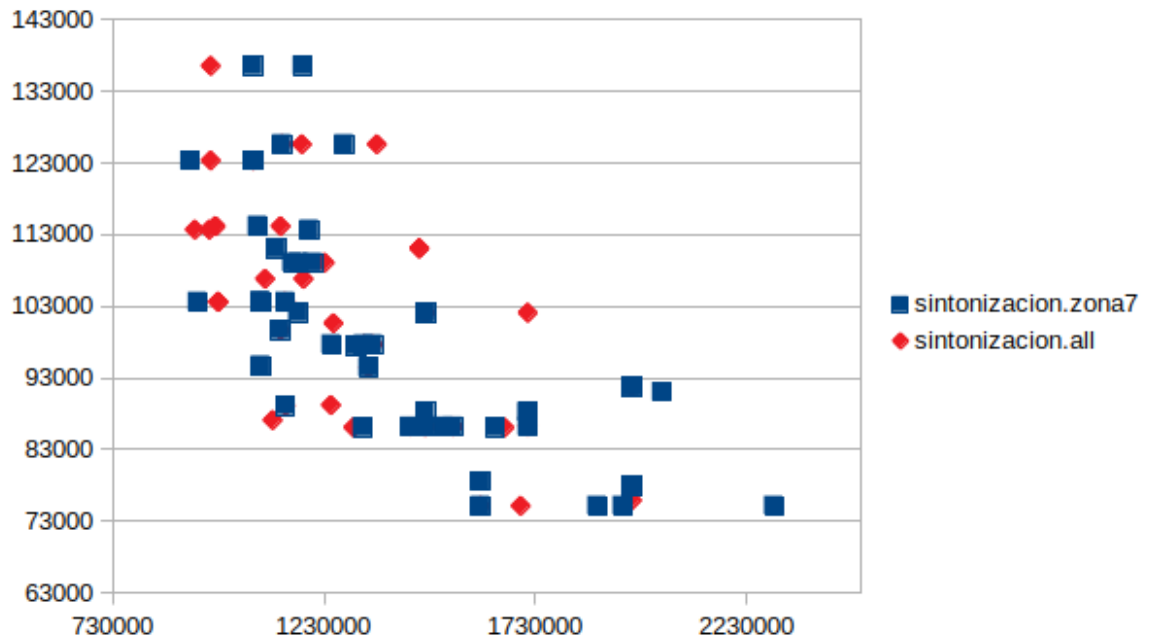


Figura 4.14: Frente de Pareto heurístico para ambas configuraciones, zona6 (en azul) y ALL (en rojo) para la zona 7.

Caso	Hipervolumen mín	Hipervolumen máx	Hipervolumen promedio
zona7	2.10E+10	6.44E+10	3.65E+10
ALL	6.04E+09	5.23E+10	2.49E+10

Tabla 4.12: Valores de hipervolumen de ambas configuraciones para la zona 7.

frente de Pareto óptimo obtenido por el solver. A pesar de la cantidad de soluciones obtenidas por el acercamiento en un amplio rango del espacio objetivo, estas soluciones presentan importantes problemas de convergencia respecto a lo que se considera el conjunto óptimo de Pareto. El tiempo de ejecución total del algoritmo completo corresponde 17.18 segundos y para la heurística toma un tiempo de alrededor de 770 segundos. Si bien los resultados obtenidos por el acercamiento heurístico no se acercan lo suficiente al conjunto óptimo de Pareto. Es posible visualizar que para el eje Y (que corresponde a las distancias) los resultados son bastante buenos y comparables con los obtenidos por el solver.

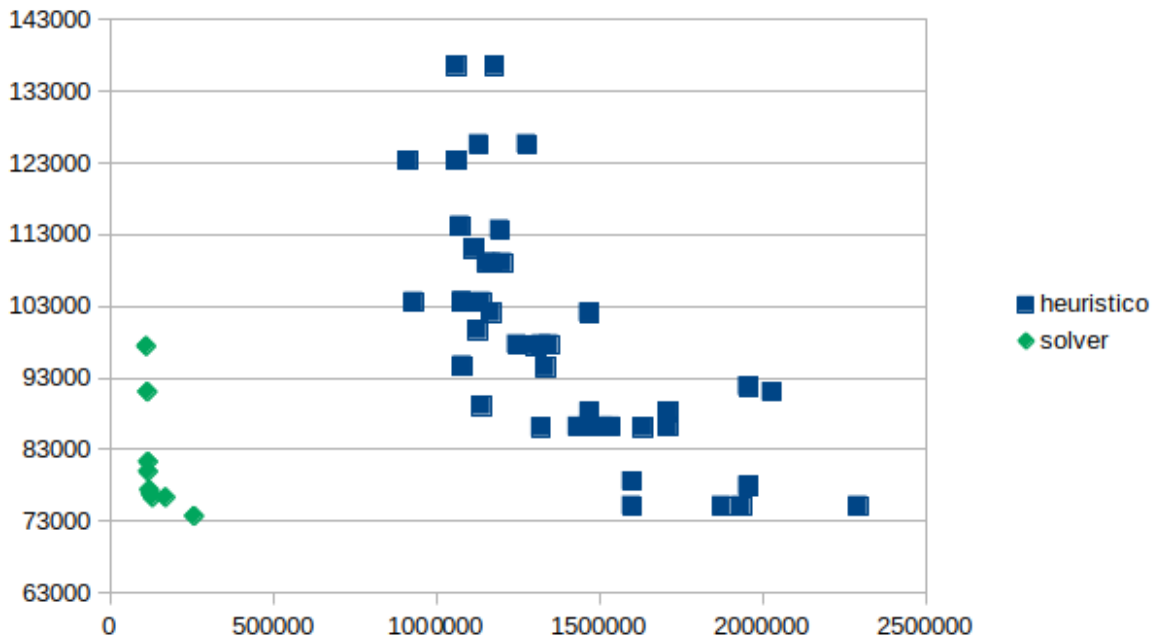


Figura 4.15: Frente de Pareto exacto (en verde) versus Frente de Pareto heurístico (en azul) para la zona 7.

4.3. Resumen

Para el acercamiento heurístico, las principales diferencias de los resultados de ambas configuraciones están en que tanto pueden diversificar. Para las instancias más grandes (zona 1, zona 2) se observó que la configuración ALL, que tenía puntos en contra respecto a las sintonizaciones respectivas de cada instancia, obtuvo una mejor diversificación. Una de las razones se puede deber a que la cantidad de clientes es tan grande, que el algoritmo es capaz de diversificar sólo utilizando los movimientos (en su mayoría 2-OPT, por las condiciones del problema).

Para las instancias más pequeñas (zona 3 y zona 7), es menos probable que el algoritmo pueda diversificar sólo utilizando los movimientos, ya que hay menos posibilidades de aplicaciones de movimientos. Por lo que las configuraciones si tienen impacto y se ven favorecidas principalmente por el parámetro *tabu_size*, el cual establece el largo de la lista tabú y con ésta es posible impedir que el algoritmo se quede atrapado en ciclos.

Para la zona 4, la configuración de la instancia se pudo haber visto favorecida en diversificación, por el valor de *tabu_size*, que si bien no puede ser muy alto o sino el algoritmo no visitaría otras soluciones, pero de todas maneras se observa que tuvo un impacto en impedir ciclos y así aumentar la diversificación. El otro parámetro que favorece a la sintonización de la instancia es *p_insert*, por la misma razón de que el movimiento insert es muy poco utilizado por las condiciones del problema, entonces si este valor es alto, gran parte de las iteraciones del algoritmo que puedan permitir visitar otras soluciones con 2-OPT, puedan ser impedidas. A pesar de que estos parámetros favorecen a la configuración de la zona, la diferencia entre la cantidad de soluciones de ambas sintonizaciones no es muy grande, ya que la cantidad de clientes de esta instancia también es alta (30 clientes).

Para las dos zonas restantes (5 y 6) que tienen una cantidad de clientes casi igual (21 y 22 respectivamente), no se observó una relación clara entre la cantidad de clientes y si la sintonización impactó en la diversificación de las instancias. Pero si cabe destacar que con la zona 5 la de cantidad de soluciones utilizando la configuración de la instancia es mucho mayor que con la sintonización ALL, principalmente se puede deber al valor n_α , al tener un mayor cantidad de α , el algoritmo realiza aún más iteraciones. Y también por el valor *p_insert*, que corresponde al valor 0 y por las mismas razones ya mencionadas, esto beneficia la diversificación.

Se observó que los resultados de las siete instancias (con una pequeña excepción) obtenidos por el acercamiento heurístico distan bastante de la solución óptima obtenida por el SOLVER, en valores absolutos. La pequeña excepción está en la figura 4.15, específicamente en el eje Y, que corresponde a las distancias. En este caso los valores del frente de Pareto óptimo no distan de los obtenidos por el acercamiento heurístico.

Cabe destacar que no se tenían las rutas obtenidas por el solver, por lo que fue imposible comprobar los resultados, como para validar en detalle y así poder sacar mejores conclusiones de las comparaciones de ambos algoritmos.

Capítulo 5

Conclusiones

En el presente trabajo de memoria se estudió el problema multi-objetivo de *distribución de materiales peligrosos con mezcla*, su símil en la literatura corresponde al VRP (problema de enrutamiento de vehículos). Lo que transforma este problema en un problema multi-objetivo la necesidad de minimizar dos objetivos a la vez. Por una parte, es importante minimizar los costos de transporte asociados a las distancias recorridas por las rutas de distribución. Y por otro lado, es fundamental minimizar la población expuesta a posibles reacciones de los materiales peligrosos transportados. Además, en el proceso de distribución se permite transportar en un mismo camión diferentes tipos de materiales, lo que se entiende como mezcla. Esto, siempre y cuando, los materiales transportados en un mismo camión sean compatibles entre ellos.

Las instancias de prueba del problema estudiado corresponden a un caso real de la ciudad de Santiago de Chile. En las instancias de prueba consideradas, la ciudad es dividida en siete zonas geográficas distintas, que se considera, funcionan de forma independiente. Así entonces, cada instancia considera sus propios clientes y camiones de reparto. Lo único que tienen en común las instancias de prueba asociadas a las siete zonas de distribución es el depósito desde donde salen los camiones a realizar el reparto y donde vuelven al final de día. Las instancias de Santiago de Chile consideran cinco tipos de materiales peligrosos dada la legislación vigente en el país. El detalle de compatibilidad de los materiales se presenta en la tabla 1.1 del capítulo 1.

Los datos asociados a la red vial de Santiago de Chile, corresponden a información de distancias y riesgos medidos entre intersecciones de calles. Para obtener distancias y riesgos entre el depósito y los clientes es necesario calcular caminos de costos mínimo sobre la red completa minimizando de forma simultánea ambos objetivos. Para esto se utilizó el algoritmo Dijkstra considerando pesos relativos de los objetivos. Los datos de cada instancia (zonas de la ciudad de Santiago), también fueron pre-procesados y almacenados. Estos datos principalmente corresponden a cantidad de clientes, camiones, carga y tipo de materiales requeridos por cada cliente y compatibilidad entre ellos.

Así, el problema requiere diseñar rutas que sean capaces de visitar a todos los clientes y cumplir con las restricciones de compatibilidades entre materiales y capacidad de rutas minimizando ambos objetivos del problema.

En este trabajo de título se propone un algoritmo heurístico para resolver el problema planteado. La solución inicial del problema es construida utilizando un algoritmo Greedy. El algoritmo greedy trata de obtener la mejor ruta eligiendo las visitas paso a paso. Para este caso la mejor ruta es definida como aquella que minimiza los costos. Para la construcción inicial no se considera el objetivo de riesgos, ya que es necesario conocer la ruta completa para el cálculo de este objetivo.

Para mejorar la calidad de las soluciones iniciales se implementa un algoritmo de búsqueda local basado en Tabu Search. Dado que se busca resolver un problema multi-objetivo, iterativamente se cambian las ponderaciones de los objetivos para considerar una función de evaluación basada en pesos. El algoritmo Tabu Search implementa dos operadores de búsqueda: 2-OPT e Insert. El operador 2-OPT es reconocido la literatura por su capacidad de recorrer espacio de búsqueda asociados a recorridos. El operador 2-OPT busca cambiar las secuencias de visitas de un conjunto dado de clientes. El operador Insert modifica las rutas, quitándole visitas y agregándoselas a otras rutas. Además, se considera el uso de una lista tabú, para impedir ciclos durante el proceso de búsqueda y así evitar volver a visitar soluciones o espacios de búsqueda ya analizados. Además, el algoritmo implementa un proceso de archivado que permite llevar registro de las soluciones no dominadas que se han encontrado a lo largo de cada proceso de búsqueda.

El algoritmo utilizado para la resolución del problema, se evaluó utilizando las instancias de prueba construidas en las siete zonas de la ciudad de Santiago de Chile. Las instancias consideran entre 11 y 36 clientes, todas de ellas tienen dos camiones disponibles para utilizar, los camiones tienen capacidad suficiente para visitar a todos los clientes de cada zona y, además, todas las instancias consideran clientes que requieren los cinco tipos de materiales peligrosos. Por último, todas las instancias tienen las mismas reglas de compatibilidad entre materiales, y en caso de que se carguen en el mismo camión, se considera que el tramo de la ruta correspondiente posee el riesgo del material más peligroso para la población, este material se denota como material dominante.

Para el seteo de los parámetros del algoritmo se llevó a cabo un proceso de sintonización utilizando en bien conocido sintonizador ParamILS. Se definieron ocho sintonizaciones distintas, donde siete de ellas correspondían a un escenario particular para cada instancia. Y la octava configuración de parámetros fue conseguida usando como instancias de entrenamiento del proceso las siete instancias del problema. La calidad de los resultados se compara utilizando el indicador de hipervolumen para problemas multi-objetivo. Las sintonizaciones tuvieron un impacto para aquellas instancias donde la cantidad de clientes era muy pequeña (como las zonas 3 y 7), el algoritmo fue capaz de diversificar (obteniendo una mayor cantidad de soluciones) aún más. Para las instancias más grandes (zona 1 y 2), las configuraciones no tuvieron un impacto en la diversificación. Esto se puede deber a que una gran cantidad de clientes, permite diversificar por sí sola, más aún si se piensa que la cantidad de camiones es baja y se aprovecha bastante el movimiento 2-OPT. Para la zona 4, donde la cantidad de clientes es alta, pero no mayor a las de la zona 1 y 2, el algoritmo fue impactado positivamente por las sintonizaciones de la instancia, la diferencia no fue muy grande, pero se obtuvieron una mayor cantidad de soluciones. Para el resto (zonas 4 y 5), que tenían una cantidad de clientes media, no se observó una clara relación entre esta variable y la configuración de los parámetros.

Además los parámetros de las configuraciones impactan en los resultados. Por ejemplo para el caso de *tabu_size* puede haber pasado que para la mayoría de los resultados utilizando la configuración ALL, el algoritmo pudo haber quedado atrapado en ciclos, ya que el valor del parámetro es 0. El valor de α , impacta principalmente en la diversificación, ya que si este

valor es más alto, el algoritmo realiza mucha más iteraciones, a pesar de esto hay un punto en que si se aumenta aún más el valor de este parámetro, puede que no aumente lo suficiente la cantidad de soluciones. Y el último parámetro *p_insert*, impacta favorablemente al algoritmo heurístico si el valor es menor, ya que el movimiento insert es muy poco utilizado por las condiciones del problema, por lo tanto resulta más provechoso utilizar las iteraciones del algoritmo en el movimiento 2-OPT. Es por esto que es una buena opción dejar el valor del parámetro como cero.

A partir de estos resultados se observa que el acercamiento heurístico parece buscar bien en los extremos de los frentes (en las esquinas de arriba a la izquierda y en la de abajo a la derecha de los frentes de Pareto), pero le cuesta encontrar soluciones en los puntos medios en la mayoría de los casos. Una de las posibles razones puede ser que estos son espacios de búsqueda más complejos que no pudieron ser visitados por los movimientos utilizados en el algoritmo propuesto. Por lo que si se utilizan operadores más complejos que puedan diversificar aún mas las soluciones, puede ser posible mezclar subrutinas que no fueron alcanzadas por los movimientos propuestos para el acercamiento.

Al comparar los resultados con los obtenidos en la literatura, no se lograron obtener resultados similares. Al observarlos, se ve claramente que los resultados de la literatura son mejores. Aunque se debe tener en cuenta el tiempo de ejecución del algoritmo en instancias grandes y pequeñas. Por ejemplo para las instancias, zona 1 y zona 2 que son las instancias con mayor cantidad de clientes, el tiempo de ejecución del algoritmo utilizado en la literatura fue de aproximadamente 100,000 segundos. Mientras que el algoritmo utilizado para el presente trabajo de memoria, fue de casi 10 veces menos. Si bien la instancia que sigue (zona 4), también tiene una cantidad de clientes bastante alta, los tiempos de ejecución fueron bastante similares. Para las instancias más pequeñas, el tiempo de ejecución fue muy similar para la zona 3. Pero para la zona 7 el tiempo de cómputo del algoritmo de la literatura fue casi 46 veces menos. Para las instancias zona 5 y zona 6 no se ve una clara relación entre los tiempos de ejecución y calidad de la solución.

Como trabajo futuro se considera utilizar el algoritmo propuesto en ese trabajo, en instancias más grandes y con mayor cantidad de camiones así se puede aprovechar de mejor forma el movimiento Insert. Ya que debido a las condiciones de las instancias, es muy difícil

que haya sido utilizado apropiadamente. Además es posible sumar variaciones del algoritmo genético, con el fin de diversificar aún más el espacio de soluciones. Y también es posible agregar nuevos movimientos, uno de estos puede ser el movimiento SWAP, que es fácil de implementar y además no existirían muchos problemas en tiempos de ejecución para aplicarlo.

Bibliografía

- [1] Alan Holliday Anthony Wren. Computer scheduling of vehicles from one or more depots to a number of delivery points. *The OR Society*, pages 2–4, 1972.
- [2] Carlos Andres Zabala Arias. Implementación del sistema de colonia de hormigas con búsqueda local al problema de ruteo de vehículos con capacidad limitada y ventanas de tiempo (cvrptw). *UniAndes*, pages 14–19, 2005.
- [3] Gustavo Alfredo Bula. Vehicle routing for hazardous material transportation. *Universidad de Tecnología de Troyes*, page 118, 2016.
- [4] Puglisi Alfio David Barilla, Giovanni Leonardi. Risk assessment for hazardous materials transportation. *ResearchGate*, pages 5–12, 2009.
- [5] Instituto Nacional de Normalización INN. Decreto Supremo N 298. Sustancias peligrosas - terminología y clasificación general. *Norma Chilena Oficial*, 2004.
- [6] Amin Faridnia Mehdi Soltani Gerhard-Wilhelm Weber Erfan Babae Tirkolaee, Alireza Goli. Multi-objective optimization for the reliable pollution-routing problem with cross-dock selection using pareto-based algorithms. *ElSevier*, pages 4–21, 2019.
- [7] Armann Ingolfsson Erhan Erkut1. Transport risk models for hazardous materials: revisited. *ElSevier*, pages 3–7, 2003.
- [8] J. H. Ramser G. B. Dantzig. The truck dispatching problem. *PubsOnLine*, pages 1–3, 1959.
- [9] Vladimir Marianov Guillermo Latorre-Núñez Germán Paredes-Belmar, Andrés Bronfman. Hazardous materials collection with multiple-product loading. *ScienceDirect*, pages 3–7, 2017.
- [10] Ioannis Giannikos. A multiobjective programming model for locating treatment sites and routing hazardous wastes. *ElSevier*, pages 2–7, 1998.
- [11] Fabio A. González-Caroline Prodhon Nubia Velasco Gustavo A. Bula, H. Murat Afsar. Bi-objective vehicle routing problem for hazardous materials transportation. *ElSevier*, pages 3–13, 2018.

- [12] Yuanyuan Zhang Changjing Shang Sicheng Zhang Hao Hu, Xiang Li. Multi-objective location-routing model for hazardous material logistics with traffic restriction constraint in inter-city roads. *ElSevier*, pages 4–7, 2018.
- [13] Nikolai Holeczek. Analysis of different risk models for the hazardous materials vehicle routingproblem in urban areas. *ElSevier*, pages 2–10, 2020.
- [14] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. Paramils: An automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36(1):267–306, September 2009.
- [15] M Barkaoui J Berger. A new hybrid genetic algorithm for the capacitatedvehicle routing problem. *The OR Society*, pages 2–5, 2003.
- [16] Steve Foote K. David Pijawka and Andy Soesilo. Improving transportation of hazardous materials through risk assessment and routing. *Transportation Research Board*, pages 1–4, 1980.
- [17] Saadedeen Samara Kostas G. Zografos. Combined location-routing model for hazardous waste transportation and disposal. *Transportation Research Board*, pages 2–4, 1989.
- [18] Giovanni Agazzi Luca Maria Gambardella, Eric Taillard. Macs-vrptw: A multiple ant colony system for vehicle routing problems with time windows. *ResearchGate*, pages 3–7, 1999.
- [19] Hamed Farrokhi-Asl Navid Rahimi Masoud Rabbania, Razieh Heidari. Using metaheuristic algorithms to solve a multi-objective industrial hazardous waste location-routing problem considering incompatible waste types. *ElSevier*, pages 3–12, 2016.
- [20] Gilbert Laporte Michel Gendreau, Alain Hertz. A tabu search heuristic for the vehicle routing problem. *JSTOR*, pages 3–7, 1994.
- [21] Ministerio de Salud. Decreto Supremo N 148 MINSAL. Aprueba reglamento sanitario sobre manejo de residuos peligrosos. *Diario Oficial*, 2004.
- [22] Ministerio de Salud. Decreto Supremo N 298 MINSAL. Reglamenta transporte de cargas peligrosas por calles y caminos. *Diario Oficial*, page Artículo 87, 2004.
- [23] Consolación Gilb Antonio Fernández Francisco de Toro Raúl Bañosa, Julio Ortega. A simulated annealing-based parallel multi-objective approach to vehicle routing problems with time windows. *ElSevier*, pages 4–7, 2013.
- [24] Tadashi Yamada Rojee Pradhananga, Eiichi Taniguchi. Ant colony system based routing and scheduling for hazardous material transportation. *ElSevier*, pages 2–10, 2017.

- [25] Fatemeh Zandiyeh Seyed Farid Ghannadpour. An adapted multi-objective genetic algorithm for solving the cash in transit vehicle routing problem with vulnerability estimation for risk quantification. *ElSevier*, pages 5–20, 2019.
- [26] Marius M. Solomon. Algorithms for the vehicle routing and scheduling problems with time window constraints. *PubsOnlinep*, pages 7–11, 1987.
- [27] Gary L. Urbanek and Edward J. Barber. Development of criteria to designate routes for transporting hazardous materials. *Washington*, pages 91–94, 1980.
- [28] Eckart Zitzler and Lothar Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.