

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ**

**ALGORITMO GENÉTICO MULTIOBJETIVO PARA LA
OPTIMIZACIÓN DE LA DISTRIBUCIÓN DE AYUDA
HUMANITARIA EN CASO DE DESASTRES NATURALES EN EL
PERÚ**

Tesis para optar por el Título de Ingeniero Informático que presenta el bachiller:

Robert Alonso Aduviri Choque

Asesor: Mg. Rony Cueva Moscoso

Lima, Junio de 2018

Resumen

El Perú es un país susceptible a desastres naturales, hecho evidenciado con el terremoto de Pisco en 2007 y el fenómeno de El Niño Costero en 2017. Ante la ocurrencia de un desastre surge una cantidad considerable y variada de ayuda proveniente de distintos lugares: INDECI, empresas privadas, ciudadanía y ayuda internacional. Es posible identificar una demanda de bienes de ayuda diferenciada entre las zonas afectadas; sin embargo, el plan logístico del INDECI no indica un procedimiento de distribución de ayuda humanitaria con sustento científico. Ante esto, en el presente proyecto se propone un modelo de programación no lineal entero mixto que servirá como base para generar un plan de distribución de ayuda humanitaria a gran escala de manera automática, abarcando los niveles nacional y regional luego de un desastre. Asimismo, dado que en escenarios reales el tiempo de ejecución de métodos para encontrar soluciones para este modelo es limitado, se propone el diseño e implementación de un algoritmo genético, el cual es un algoritmo de optimización metaheurístico basado en principios evolutivos que se caracteriza por obtener soluciones aproximadas con un tiempo de ejecución menor al de métodos exactos, y permiten generar poblaciones de posibles soluciones para evaluar múltiples objetivos de optimización como la minimización de la inequidad en la distribución y el costo de transporte. Para probar el método exacto y metaheurístico se generará información a partir de datos disponibles del INDECI y la simulación de datos de prueba a partir de distribuciones de probabilidad. Asimismo, se realizará una comparación entre estos métodos, buscando obtener con el algoritmo genético una efectividad aproximada al del método exacto en un tiempo de ejecución mucho menor, y con el objetivo de conocer a partir de qué tamaño de problema es preferible el uso del algoritmo genético. Finalmente, como cada plan de distribución dependerá de las características del desastre a analizar, se propone la implementación de una interfaz de software para la entrada y visualización de información.

Tabla de Contenido

Resumen	2
Tabla de Contenido	3
Índice de Figuras	9
Capítulo 1. Generalidades.....	11
1.1 Problemática	11
1.2 Objetivos	14
1.2.1 Objetivo general.....	14
1.2.2 Objetivos específicos	14
1.2.3 Resultados esperados	15
1.2.4 Mapeo de objetivos, resultados y verificación	16
1.3 Herramientas y Métodos	19
1.3.1 Python.....	19
1.3.2 Jupyter Lab	19
1.3.3 Visual Studio Code	20
1.3.4 AMPL	20
1.3.5 AMPL IDE	20
1.3.6 Couenne	20
1.3.7 Kanban.....	21
1.3.8 Prueba de Shapiro-Wilk	21
1.4 Viabilidad.....	21
1.4.1 Viabilidad Técnica.....	21
1.4.2 Viabilidad Temporal	22
1.4.3 Viabilidad Económica.....	22
1.4.4 Conclusión	22
1.5 Alcance, Limitaciones y Riesgos.....	23
1.5.1 Alcance	23

1.5.2	Limitaciones	23
1.5.3	Riesgos	24
Capítulo 2.	Marco Conceptual.....	25
2.1	Introducción.....	25
2.2	Peligros y desastres.....	25
2.3	Gestión del desastre y logística humanitaria	25
2.4	Logística humanitaria en el Perú.....	27
2.5	Optimización Combinatoria	27
2.6	Optimización de múltiples objetivos	28
2.7	Problema de transporte.....	28
2.8	Problema de transbordo.....	29
2.9	Métodos para la solución de problemas de optimización	30
2.9.1	Métodos exactos.....	30
2.9.2	Métodos heurísticos.....	30
2.9.3	Métodos metaheurísticos.....	31
2.9.3.1	Algoritmos Genéticos	32
Capítulo 3.	Estado del Arte	33
3.1	Introducción.....	33
3.2	Investigaciones Académicas.....	33
3.2.1	Transporte en operaciones de respuesta a desastres (<i>Transportation in disaster response operations</i>).....	33
3.2.2	Enrutamiento para esfuerzos de ayuda humanitaria (<i>Routing for relief efforts</i>)	33
3.2.3	Modelos para enrutamiento de ayuda humanitaria: Equidad, eficiencia y eficacia (<i>Models for relief routing: Equity, efficiency and efficacy</i>).....	34
3.2.4	Óptima reubicación de stock bajo incertidumbre en operaciones humanitarias post-desastre (<i>Optimal stock relocation under uncertainty in post-disaster humanitarian operations</i>).....	35

3.2.5	Planificación óptima multi-objetivo para el diseño de sistemas de distribución de ayuda humanitaria (<i>Multi-objective optimal planning for designing relief delivery systems</i>).....	35
3.2.6	Metaheurística bi-objetivo para la planificación de operaciones de ayuda humanitaria en desastres (<i>Bi-objective metaheuristic for disaster relief operation planning</i>).....	36
3.2.7	Modelo de optimización multi-criterio para la distribución de ayuda humanitaria (<i>A multi-criteria optimization model for humanitarian aid distribution</i>).....	36
3.3	Productos comerciales.....	37
3.3.1	SimpliRoute.....	37
3.3.2	SAP Transportation Management VSR Optimization	38
3.3.3	Google Optimization Tools.....	39
3.4	Revisión y discusión.....	39
3.5	Conclusiones.....	40
Capítulo 4.	Definición de la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria	41
4.1	Introducción.....	41
4.2	Resultado	41
4.2.1	Parámetros del problema de optimización.....	41
4.2.1.1	Red de distribución.....	41
4.2.1.2	Costo de transporte.....	41
4.2.1.3	Oferta y demanda en los nodos de la red de distribución	41
4.2.1.4	Capacidad de almacenes.....	42
4.2.2	Variables de decisión.....	42
4.2.2.1	Cantidad de bienes que se transportan	42
4.2.3	Función objetivo.....	42
4.2.3.1	Minimizar costo de transporte	42
4.2.3.2	Minimizar inequidad en la distribución	43

4.2.4	Restricciones	44
4.2.4.1	Balance en los puntos de abastecimiento.....	44
4.2.4.2	Balance en los puntos de demanda	45
4.2.4.3	Balance en los puntos de transbordo.....	45
4.2.4.4	Evitar sobrepasar la capacidad de los almacenes	45
4.2.5	Rango de existencia	46
4.3	Relación con el objetivo	46
Capítulo 5. Definición de las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria.....		47
5.1	Introducción.....	47
5.2	Resultado	47
5.2.1	Estructuras de datos para los parámetros iniciales y variables auxiliares del problema.....	47
5.2.2	Variable de decisión y cromosoma	48
5.2.3	Cromosoma	48
5.3	Relación con el objetivo	54
Capítulo 6. Diseño del algoritmo genético para el problema de optimización de distribución de ayuda humanitaria.....		55
6.1	Introducción.....	55
6.2	Esquema general del algoritmo genético.....	55
6.3	Construcción de la población inicial	55
6.3.1	Generación de cromosomas.....	56
6.4	Evaluación de la aptitud	56
6.4.1	Frente de Pareto	57
6.5	Selección.....	58
6.5.1	Selección por torneo	59
6.6	Cruce.....	60
6.7	Mutación.....	62

6.8	Criterio de parada	63
Capítulo 7. Diseño del modelo matemático de programación no lineal para el problema de optimización de distribución de ayuda humanitaria..... 64		
7.1	Introducción.....	64
7.2	Formulación del modelo matemático	64
7.2.1	Conjuntos.....	64
7.2.2	Parámetros	64
7.2.3	Variables de decisión.....	64
7.2.4	Funciones objetivo	65
7.2.5	Restricciones	65
7.2.6	Rango de existencia	66
7.3	Algoritmo de optimización exacto (<i>solver</i>)	66
Capítulo 8. Experimentación Numérica..... 69		
8.1	Introducción.....	69
8.2	Configuración de la experimentación.....	69
8.2.1	Generación de conjuntos de datos aleatorios.....	69
8.2.2	Recolección de datos.....	70
8.3	Resultados de la experimentación	70
8.3.1	Tiempo de ejecución.....	71
8.3.2	Valor objetivo	76
8.4	Discusión y conclusiones	80
Capítulo 9. Evaluación del algoritmo genético multiobjetivo basado en el análisis de datos de un escenario real		
9.1	Introducción.....	81
9.2	Contexto del escenario real	81
9.3	Recolección y construcción del conjunto de datos	82
9.4	Ejecución de los algoritmos de optimización	86
Capítulo 10. Interfaz de usuario para la ejecución del algoritmo genético y el método exacto		
	91	

10.1	Introducción.....	91
10.2	Ingreso de datos	91
10.3	Ejecución de los algoritmos	91
Capítulo 11.	Conclusiones y trabajos futuros	93
11.1	Conclusiones.....	93
11.2	Trabajos futuros	95
Referencias	96



Índice de Figuras

Ilustración 1. Fenomenología más frecuente en Perú (INDECI, 2008).	11
Ilustración 2. Comparación entre resultados de programación entera mixta y heurística (Campbell et al., 2008).	34
Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010).	35
Ilustración 4. Matriz de recompensa para cada objetivo de optimización (Vitoriano, Ortuño, Tirado, Montero, et al., 2011).	37
Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018).	38
Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018).	39
Ilustración 7. Región de soluciones dominadas por la solución A en un espacio de 2 objetivos (Luke, 2015)	57
Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015).	58
Ilustración 9. Dispersión de las soluciones A y B. B está más alejado de sus vecinos, por lo cual tendría un valor más alto de dispersión (Luke, 2015).	59
Ilustración 10: Lenguaje de modelado AMPL	67
Ilustración 11: Ingreso de datos en lenguaje AMPL	67
Ilustración 12: Ejecución y reporte de resultados de Couenne	68
Ilustración 13: Escenario de riesgos por heladas y friaje 2017 (CENEPRED, SGRD, MINSA, MVCS, MINEDU, MIMP y MINAGRI)	82
Ilustración 14: Vista previa de los datos del Plan Multisectorial ante Heladas y Friaje (PCM, 2017)	83
Ilustración 15: Estadísticas de los datos del Plan Multisectorial antes Heladas y Friaje 2017 (PCM)	84
Ilustración 16: Interfaz de usuario para la ejecución de los algoritmos de optimización	92

Índice de Tablas

Tabla 1. Cronograma del proyecto.....	22
Tabla 2. Matriz de riesgos	24
Tabla 3: Resultados del tiempo de ejecución de los algoritmos.....	72
Tabla 4: Resultados del valor objetivo de los algoritmos	77
Tabla 5: Diferencia relativa entre valores objetivo.....	79
Tabla 6: Diferencia relativa entre valores objetivo - línea base.....	79



Capítulo 1. Generalidades

1.1 Problemática

A lo largo de los años, la humanidad se ha visto continuamente afectada por la presencia de desastres naturales, como terremotos, tsunamis, erupciones volcánicas, tornados e inundaciones. Dependiendo de la ubicación geográfica, las diversas civilizaciones en el mundo han afrontado determinados tipos de desastres naturales con mayor o menor frecuencia. Mientras que EEUU es continuamente afectada por huracanes, China es afectada por inundaciones y Filipinas es afectada por ciclones. En el caso del Perú, debido a su ubicación en el borde occidental de Sudamérica, la cual es una de las regiones de mayor actividad tectónica en el mundo, conocida como el Cinturón de Fuego del Pacífico, es constantemente afectada por terremotos, deslizamientos de tierra, erupciones volcánicas e inundaciones (Young & León, 2009).



Fuente: Instituto Nacional de Defensa Civil (INDECI)

Ilustración 1. Fenomenología más frecuente en Perú (INDECI, 2008).

Dado que los desastres naturales se caracterizan por su inevitabilidad y por lo difícil de su predicción (Kumar & Havey, 2013), se requiere de una ardua preparación para enfrentarlos con la mínima cantidad de daños materiales y principalmente humanos. Dentro de los últimos terremotos acontecidos en Perú, el último terremoto de gran magnitud aconteció en Pisco el 15 de agosto del 2007, con una intensidad de 7.9 grados en la escala de Richter contando con 1294 heridos, 596 víctimas mortales y más de 90,000 casas destruidas (INDECI, 2008).

Dicho desastre permitió evidenciar la calidad de planificación y respuesta, dirigida por el INDECI (Instituto Nacional de Defensa Civil). Lamentablemente, esta se caracterizó por una desafortunada respuesta caótica y falta de coordinación e información, así como la falta de consideración de los distintos tipos de demanda de cada zona, tal como queda expresado en el documento de lecciones aprendidas del INDECI:

“Las necesidades diferenciadas entre los damnificados de la costa y de la sierra resultaron evidentes durante la fase de emergencia: mientras los primeros esperaban ayuda en alimentos, abrigo y techo, los segundos solicitaban principalmente herramientas manuales para reconstruir sus canales de regadío colapsado.”

Asimismo, resultan lamentables las consecuencias adicionales que se pueden generar debido a un mal manejo de la logística por parte del gobierno. Como ejemplo claro, cuando la población pasa tiempo sin conseguir recibir la ayuda que necesita, la tendencia a que se realicen actos vandálicos se ve incrementada, hechos que efectivamente fueron presenciados durante el citado terremoto de Pisco (La Jornada, 2007):

“La gente se queja de que no está llegando la ayuda, por eso no hay seguridad. Lo contado por el director de un hospital de Chíncha parece confirmar su dicho. Según el médico Jorge Parrera, el centro de salud que dirige fue atacado por varios civiles que pensaban que ahí se estaban guardando los alimentos que debían ser distribuidos.”

Considerando que se movilizaron más de 14,000 toneladas de ayuda humanitaria, es claro que un planeamiento logístico efectivo y rápido es crucial para el bienestar de la población peruana en casos de desastres. La pregunta es ¿Cómo distribuir la ayuda humanitaria de forma eficiente?

Se sabe que el planeamiento logístico involucra la solución a diversos tipos de problemas de optimización, incluyendo problemas de ubicación de almacenes, diseño de la cadena de suministro, distribución a gran escala, distribución de última milla, evacuación y planeamiento de inventario (Ortuño et al., 2013). En el caso de logística de ayuda humanitaria, se pueden distinguir como características principales el volumen de bienes a transportar ubicados en almacenes que normalmente se encuentran alejados de la zona de desastre, la urgencia de la distribución y la priorización del tipo de bienes que requiera cada zona afectada. Por tales motivos, uno de los problemas más críticos a resolver después de acontecido el desastre es la planificación rápida de distribución a gran escala.

La planificación de distribución presenta un escenario donde existen centros de distribución o almacenes, los cuales contienen bienes que serán movilizados por medios de transporte hacia puntos de demanda, considerando la existencia de puntos intermedios o de transbordo. El total de estas ubicaciones forman la red de distribución. Ahora, una consideración importante para la distribución es que la cantidad de bienes que pueden ser transportados están limitados por capacidades de los almacenes y medios de transporte. Es más, en escenarios de distribución a gran escala, la distribución suele realizarse en un horizonte de tiempo de días, realizándose por fases, lo cual añade una dimensión de complejidad al problema de crear un plan de distribución óptimo. Finalmente, los criterios para determinar el mejor plan de distribución pueden estar dados por diversos factores, incluyendo el costo de transporte, la equidad en la distribución y los costos de inventario sin utilizar.

Para poder abordar un problema de tal magnitud, durante las últimas décadas se ha visto un incremento en la utilización de métodos computacionales para la resolución de problemas de optimización. En particular, en Norteamérica y Europa, se ha mostrado que el uso de procedimientos automatizados mediante el uso de modelos de optimización para los procesos de planificación de distribución produce ahorros sustanciales (generalmente entre el 5% y 20%) en los costos totales de transporte (Toth & Vigo, 2002).

Sin embargo, la aplicación de dichos métodos ha sido aplicada principalmente en la industria, gracias a la gran competitividad presente que impulsa el crecimiento de áreas de investigación y desarrollo para mantener un factor diferencial, mientras que el dominio de las cadenas de suministro de ayuda humanitaria no presenta el mismo nivel de desarrollo. A pesar de ello, en los últimos años se ha podido ver un crecimiento en la investigación relacionada a modelos de optimización en logística de emergencia resaltando principalmente modelos de optimización de ubicación de almacenes y enrutamiento de vehículos (Caunhye, Nie, & Pokharel, 2012).

Al considerar el uso de métodos computacionales para resolver problemas de optimización, un factor importante para evaluar la viabilidad de su uso es el tiempo de ejecución que demande su procesamiento ante instancias que representen escenarios reales del problema. Esto dificulta el uso de métodos exactos, por lo cual se hace necesario utilizar técnicas que requieran un menor tiempo de procesamiento a costa de tener soluciones con valores aproximados al óptimo.

En particular, uno de los métodos aproximados que ha obtenido mejores resultados es el uso de metaheurísticas o métodos de optimización estocástica, los cuales forman parte del estado del arte en resolución de problemas de optimización dada su capacidad de evitar óptimos locales, lo cual sucede frecuentemente con el uso de heurísticas (Toth & Vigo, 2002). Dada la cantidad de variables en la formulación del problema, así como los múltiples objetivos de optimización, un tipo de algoritmos metaheurísticos que son ideales para este tipo de casos son los algoritmos evolutivos o genéticos. Debido a que los algoritmos genéticos mantienen una población de soluciones candidatas, se facilita obtener un conjunto diverso de soluciones que prioricen diferentes objetivos de optimización, de tal manera que al final se obtenga el conjunto de mejores soluciones para diferentes niveles de compromiso entre los objetivos.

De esta manera, el presente proyecto de fin de carrera busca desarrollar un algoritmo bioinspirado, como es el algoritmo genético, con el fin de servir como alternativa para abordar el problema de optimización de planificación logística en caso de desastres en el Perú y como propuesta para aliviar las carencias planteadas.

1.2 Objetivos

1.2.1 Objetivo general

Implementar un algoritmo genético con múltiples objetivos para optimizar la distribución de ayuda humanitaria en caso de desastres naturales en el Perú.

1.2.2 Objetivos específicos

- 1. Definir la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria.
- 2. Definir las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria.
- 3. Definir y diseñar la función de aptitud y operadores del algoritmo genético.
- 4. Implementar el algoritmo genético para la resolución del problema de distribución de ayuda humanitaria.
- 5. Diseñar un modelo matemático de programación no lineal entero mixto a partir de la formulación del problema de optimización para resolver el problema de forma exacta.
- 6. Diseñar y desarrollar la experimentación numérica para comparar el desempeño del algoritmo genético con el método exacto.

- O 7. Desarrollar la interfaz de usuario para la ejecución y visualización de los algoritmos de optimización.

1.2.3 Resultados esperados

- R 1. Función objetivo, restricciones, variables de decisión y parámetros definidos para el problema de distribución de ayuda humanitaria (O1).
- R 2. Estructuras de datos del algoritmo genético propuesto (O2).
- R 3. Pseudocódigo del algoritmo genético conteniendo la función evaluación de aptitud y los operadores (O3).
- R 4. Algoritmo genético implementado y contenido en un componente reutilizable (O4).
- R 5. Modelo matemático de programación no lineal entero mixto formulado e implementado en un lenguaje de modelado para su optimización mediante un algoritmo de optimización exacto (*solver*) (O5).
- R 6. Componente de software que contenga el modelo matemático de programación no lineal entero mixto implementado y ejecute el algoritmo de optimización exacto (O5)
- R 7. Desarrollo de un generador de datos de prueba para los algoritmos de optimización (O6).
- R 8. Análisis de datos de un caso real para generar un caso de prueba basado en un escenario real (O6).
- R 9. Instancia de prueba basada en un escenario real conteniendo información sobre la red distribución, así como la oferta, demanda y tipos de bienes a transportar (O6).
- R 10. Instancias de prueba de diferente tamaño generadas aleatoriamente (O6).
- R 11. Informe de evaluación y comparación de tiempos de ejecución y valores objetivo de los algoritmos de optimización mediante experimentación numérica (O6).
- R 12. Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el algoritmo de optimización exacto (*solver*) (O7).

1.2.4 Mapeo de objetivos, resultados y verificación

Objetivo (O1): Definir la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria		
Resultado (R1)	Meta física	Medio de verificación
Función objetivo, restricciones, variables de decisión y parámetros definidos	Documento	- Definición de función objetivo, restricciones, variables de decisión y parámetros en el documento
Objetivo (O2): Definir las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria		
Resultado (R2)	Meta física	Medio de verificación
Estructuras de datos definidas	Documento	- Definición de estructuras de datos en el documento
Objetivo (O3): Definir y diseñar la función de aptitud y operadores del algoritmo genético		
Resultado (R3)	Meta física	Medio de verificación
Pseudocódigo del algoritmo genético conteniendo la función evaluación de aptitud y los operadores	Documento	- Definición del pseudocódigo de la función de evaluación de aptitud y los operadores del algoritmo genético en el documento
Objetivo (O4): Implementar el algoritmo genético para la resolución del problema de distribución de ayuda humanitaria		
Resultado (R4)	Meta física	Medio de verificación

Algoritmo genético implementado y contenido en un componente reutilizable	Software	<ul style="list-style-type: none"> - Código del algoritmo genético - Interfaz en consola para ejecutar el algoritmo genético
<p>Objetivo (O5): Diseñar un modelo matemático de programación no lineal entero mixto a partir de la formulación del problema de optimización para resolver el problema de forma exacta</p>		
Resultado (R5)	Meta física	Medio de verificación
Modelo matemático de programación no lineal entero mixto formulado e implementado en un lenguaje de modelado para su optimización mediante un algoritmo de optimización exacto (<i>solver</i>)	Software	<ul style="list-style-type: none"> - Modelo matemático formulado e implementado en un lenguaje de modelado para su optimización mediante un algoritmo de optimización exacto
Resultado (R6)	Meta física	Medio de verificación
Componente de software que contenga el modelo matemático de programación no lineal entero mixto implementado y ejecute el algoritmo de optimización exacto	Software	<ul style="list-style-type: none"> - Código del componente - Interfaz en consola para ejecutar el algoritmo de optimización exacto (<i>solver</i>)
<p>Objetivo (O6): Diseñar y desarrollar la experimentación numérica para comparar el desempeño del algoritmo genético con el método exacto</p>		
Resultado (R7)	Meta física	Medio de verificación

Desarrollo de un generador de datos de prueba para los algoritmos de optimización	Software	<ul style="list-style-type: none"> - Código del generador de datos - Interfaz en consola para generar datos
Resultado (R8)	Meta física	Medio de verificación
Análisis de datos de un caso real para generar un caso de prueba basada en un escenario real	Documento	<ul style="list-style-type: none"> - Análisis de los datos utilizados de un caso real describiendo dimensiones del problema de un escenario real
Resultado (R9)	Meta física	Medio de verificación
Instancia de prueba basada en un escenario real conteniendo información sobre la red distribución, así como la oferta, demanda y tipos de bienes a transportar	Conjunto de datos	<ul style="list-style-type: none"> - Archivo con datos describiendo los datos de entrada de un escenario real del problema de optimización
Resultado (R10)	Meta física	Medio de verificación
Instancias de prueba de diferente tamaño generadas aleatoriamente	Conjunto de datos	<ul style="list-style-type: none"> - Archivo con datos describiendo datos de entrada de diferente tamaño para los algoritmos de optimización
Resultado (R11)	Meta física	Medio de verificación
Informe de evaluación y comparación de tiempos	Documento	<ul style="list-style-type: none"> - Resultados de evaluación y comparación de tiempo de

de ejecución y valores objetivo de los algoritmos de optimización mediante experimentación numérica		ejecución y valores objetivo de los algoritmos de optimización
Objetivo (O7): Desarrollar la interfaz de usuario para la ejecución y visualización de los algoritmos de optimización		
Resultado (R12)	Meta física	Medio de verificación
Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el algoritmo de optimización exacto (<i>solver</i>)	Software	<ul style="list-style-type: none"> - Código de la interfaz de usuario - Interfaz gráfica para ejecutar los algoritmos de optimización

1.3 Herramientas y Métodos

1.3.1 Python

Python (Python Software Foundation, 2018) es un lenguaje de programación interpretado, interactivo y orientado a objetos que incorpora módulos, excepciones, tipos de datos dinámicos de muy alto nivel y clases. Python combina poder con una sintaxis bastante clara, así como una gran variedad de librerías que permiten incrementar la eficiencia del programador al escribir programas con el mínimo código necesario, motivo por el cual se ha optado por su uso para implementar el algoritmo genético.

1.3.2 Jupyter Lab

Jupyter Notebook (Jupyter, 2018) es una aplicación web que permite crear y compartir documentos que contienen código, ecuaciones, visualizaciones y texto descriptivo, lo cual lo hace perfecto como entorno de desarrollo para desarrollar prototipos de programas de forma rápida e incluyendo todos los elementos antes mencionados, lo cual hace la presentación del programa más visual e interactiva. Por tanto, se ha optado

por su uso en el proyecto para probar diferentes operadores para el algoritmo genético de forma ágil.

1.3.3 Visual Studio Code

Visual Studio Code (Microsoft, 2018) es un entorno de desarrollo integrado (IDE) desarrollado por Microsoft que provee inspecciones de código, destaque de errores en tiempo real, sugerencias de correcciones y altas capacidades de navegación sobre todo el código de forma bastante ligera. Por tanto, se ha optado por su uso en el proyecto para implementar el algoritmo final así como la interfaz de usuario que lo contenga.

1.3.4 AMPL

AMPL (A Mathematical Programming Language) (AMPL, 2018) es un lenguaje algebraico de modelado usado para describir y resolver problemas de gran complejidad para cómputo matemático de gran escala. Fue desarrollado en Bell Laboratories y soporta docenas de algoritmos de optimización exactos (*solvers*), tanto de código abierto como comerciales, incluyendo CBC, CPLEX, FortMP, Gurobi, MINOS, IPOPT, SNOPT, KNITRO y LGO. Una de las principales ventajas de AMPL es la similitud de su sintaxis con la notación matemática de problemas de optimización. Esto permite una concisa y legible definición de problemas de optimización. Debido a ello, se ha decidido usarlo en el presente proyecto de fin de carrera para implementar la formulación del problema de optimización.

1.3.5 AMPL IDE

AMPL IDE (AMPL, 2018) es un entorno de desarrollo integrado para modelamiento de problemas de optimización, el cual cuenta con un editor de código y consola para la ejecución de diversos algoritmos exactos (*solvers*) para problemas de optimización. Por tanto, se ha optado por su uso en el proyecto para implementar la formulación del programa lineal para resolver el problema de optimización.

1.3.6 Couenne

Couenne (Convex Over and Under Envelopes for Nonlinear Estimation) (Couenne, 2006) es un algoritmo de optimización de código abierto orientado a resolver problemas de programación no lineal entero mixto (MINLP). Cuenta con un conjunto de rutinas implementadas en ANSI C y organizadas en la forma de una librería. Debido a que su uso no requiere una licencia, se utilizará en el proyecto de fin de carrera para realizar la optimización mediante métodos exactos.

1.3.7 Kanban

Kanban (Anderson, 2014) es un marco de trabajo (*framework*) bastante popular usado para el desarrollo de software de forma ágil. Los principios del *framework* se basan en aumentar la flexibilidad mediante el uso de una lista de tareas pendientes (*backlog*) sujeta a priorización a medida avanza el proyecto, así como la reducción de las tareas realizadas en simultáneo para incrementar la eficiencia de cada tarea por separado y reducir la duración de las iteraciones. Asimismo, uno de sus valores principales consiste en la plena visualización de todas las tareas que se vienen haciendo, lo cual es muy útil como mecanismo para tener claro el nivel de avance a cada momento. Por todas estas características, se usará esta metodología ágil para la gestión del proyecto de fin de carrera a lo largo de todas sus fases.

1.3.8 Prueba de Shapiro-Wilk

La prueba de Shapiro-Wilk (Shapiro & Wilk, 1965) es una prueba estadística para evaluar la normalidad de la distribución de una muestra de valores continuos, bajo la hipótesis nula de que las muestras vienen de una población con distribución normal. A través de simulaciones de Monte Carlo se ha mostrado que la prueba de Shapiro-Wilk tiene la mejor potencia de prueba, seguida de las pruebas de Anderson-Darling y Kolmogorov-Smirnov (Razali & Wah, 2011), por lo que será usada en la etapa de experimentación numérica.

1.4 Viabilidad

1.4.1 Viabilidad Técnica

Se cuenta con los conocimientos necesarios para alcanzar los objetivos del presente proyecto de fin de carrera. Se cuenta con experiencia previa en la implementación de algoritmos metaheurísticos, así como en desarrollo de software en general, gracias a los diferentes cursos a lo largo de la carrera de Ingeniería Informática, y gracias al aprendizaje autodidacta, donde se ha hecho un extensivo uso de nuevas plataformas y lenguajes de programación.

Asimismo, según lo visto en el estado del arte, existe investigación sobre optimización con múltiples objetivos con el uso de algoritmos metaheurísticos, lo cual servirá de valiosa referencia y guía al momento de desarrollar el proyecto de fin de carrera.

Adicionalmente, se harán uso de buenas prácticas en gestión de proyectos a través del uso de metodologías ágiles, que ya se han aplicado con proyectos anteriores.

Finalmente, es importante mencionar que las herramientas a utilizar son de libre uso o se encuentran instaladas en los equipos de la universidad, por lo cual no habrá inconvenientes debido a licencias de software que puedan obstaculizar el proyecto.

1.4.2 Viabilidad Temporal

El cronograma estimado para el proyecto es el siguiente:

Tabla 1. Cronograma del proyecto

Nombre de la tarea	Inicio	Fin
Elaboración del plan de proyecto	- Miércoles, 26/10/16	- Miércoles, 16/11/16
Exposición del plan de trabajo en Proyecto de Tesis 1	- Jueves, 17/11/16	- Viernes, 02/12/16
Definición de la formulación del problema	- Lunes, 05/03/18	- Lunes 19/03/18
Definición de las estructuras de datos	- Lunes 19/03/18	- Lunes 02/04/18
Diseño del algoritmo genético	- Lunes 02/04/18	- Lunes 16/04/18
Implementación del algoritmo genético	- Lunes 16/04/18	- Lunes 30/04/18
Formulación e implementación del método exacto	- Lunes 30/04/18	- Lunes 07/05/18
Desarrollo de las pruebas de experimentación numérica	- Lunes 07/05/18	- Lunes 14/05/18
Implementación de la interfaz de usuario	- Lunes 14/05/18	- Lunes 21/05/18
Exposición final en Proyecto de Tesis 2	- Lunes 25/06/18	- Lunes 25/06/18

1.4.3 Viabilidad Económica

Como se mencionó anteriormente, las herramientas usadas serán de libre uso o estarán disponibles en equipos de la universidad, por lo cual no se incurrirá en costos por licencias. Para la parte de experimentación numérica es posible que se incurra en gastos por usar equipos en la nube (cloud computing) para realizar pruebas que demanden extenso procesamiento. Para ello, en la medida de lo posible se utilizarán créditos gratuitos que suelen ofrecer las plataformas cloud a nuevos usuarios (AWS, Azure, Google Cloud) para minimizar esta inversión.

1.4.4 Conclusión

Se ha podido notar que se cuentan con los conocimientos y herramientas necesarias para poder iniciar el proyecto de fin de carrera. Asimismo, no se requiere realizar una inversión en licencias de software y el período de tiempo es suficiente para desarrollar lo propuesto según los plazos establecidos en el cronograma. De esta manera, se puede

apreciar la viabilidad del proyecto, que garantizará que el proyecto de fin de carrera se desarrolle de forma satisfactoria.

1.5 Alcance, Limitaciones y Riesgos

1.5.1 Alcance

Este proyecto de fin de carrera tiene como fin presentar una alternativa de solución al problema de optimización de la distribución de ayuda humanitaria en Perú. Esta solución debe presentar una propuesta de asignación del transporte de diferentes cantidades y tipos de recursos a diferentes puntos de transbordo y demanda. Esta propuesta estará alineada a múltiples objetivos, tales como la equidad de distribución y la reducción del costo del transporte.

A continuación, se procederá a implementar el algoritmo genético que, tomando las consideraciones previamente mencionadas, generará soluciones con un valor cercano al óptimo. Dado que se propone hacer uso del algoritmo después de acontecido un desastre, es importante mencionar que su tiempo de ejecución no debe escapar del orden de los minutos, pues es esencial obtener soluciones rápidas en momentos críticos.

Finalmente, luego de desarrollar el algoritmo, se procederá a rodearlo de una interfaz gráfica, con el fin de que pueda ser utilizado por un mayor número y una mayor diversidad de personas. Finalmente, se evaluará el desempeño del algoritmo frente a métodos exactos con el fin de demostrar la calidad aceptable de sus resultados, así como la rapidez de su ejecución.

El alcance del proyecto no comprende el desarrollo de un sistema de información que contenga el algoritmo implementado.

1.5.2 Limitaciones

La siguiente lista muestra las limitaciones de este proyecto de fin de carrera:

- La formulación y resultados del algoritmo dependerán de la información que pueda ser recolectada. Para este proyecto, se usará data del INDECI para obtener información sobre ubicaciones geográficas de almacenes y posibles puntos de demanda, así como los diferentes tipos de recursos que forman parte de la distribución de ayuda humanitaria, mientras que la información detallada

sobre las cantidades de recursos por cada ubicación se generará aleatoriamente según distribuciones de probabilidad.

- El tiempo de ejecución del algoritmo dependerá de las características del equipo en el cual se ejecuten. Por tanto, es importante realizar la ejecución del algoritmo y el método exacto en el mismo dispositivo. La comparación de tiempos estará basada en la hora del sistema del equipo.

1.5.3 Riesgos

En la siguiente tabla se muestran los riesgos identificados que podrían impactar este proyecto de fin de carrera:

Tabla 2. Matriz de riesgos

Riesgo Identificado	Impacto en el proyecto	Medidas correctivas para mitigar
Mala planificación del proyecto	- Entregables presentados fuera de tiempo - Entregables rechazados	Tener presentes y claros los plazos de entrega para organizarse de acuerdo a ello
Pérdida parcial o total de la información y avances del proyecto	- Gran retraso en la presentación de entregables - Posible desaprobación del curso	Tener un repositorio en línea del proyecto y realizar actualizaciones y respaldos de manera periódica
Enfermedades	- Entregables presentados fuera de tiempo - Entregables rechazados	Tener siempre un margen de tiempo entre la finalización del entregable y la fecha límite para anticipar imprevistos
Mala comunicación con el asesor	- Entregables sin corregir - Entregables rechazados - Falta de asesoría de un experto	Mantener una comunicación constante con el asesor y manifestar inquietudes de forma oportuna
Falta de acceso a la información dependiente de externos	- Falta de información para la ejecución del algoritmo	Gestionar los accesos de forma oportuna, y en caso no se consiga, obtener información de libre disponibilidad o generada personalmente

Capítulo 2. Marco Conceptual

2.1 Introducción

En el presente apartado se definirán conceptos que serán de ayuda para entender mejor el problema que se desea resolver, así como la solución propuesta. Se abordarán conceptos de logística humanitaria que permitirán entender el contexto y consideraciones que se tengan con el problema, y posteriormente se abarcarán conceptos de optimización combinatoria, complejidad computacional y los problemas relevantes que forman la base del problema de distribución a resolver, así como los métodos exactos y aproximados que existen para afrontarlos, que comprenden métodos heurísticos y metaheurísticos.

2.2 Peligros y desastres

Un peligro es un evento amenazante o la probabilidad de ocurrencia de un fenómeno potencialmente dañino dentro de un determinado período y área (Ortuño et al., 2013). Puede ser causado tanto por la naturaleza como por el hombre. Una emergencia es una situación que presenta un riesgo inmediato a la salud, vida, propiedad o entorno. Un desastre es la disrupción del normal funcionamiento de un sistema o comunidad, que causa un fuerte impacto en las personas, estructuras y entorno, y supera la capacidad de respuesta. Algunas veces, el declarar o no una emergencia como desastre es una decisión política, pues tiene consecuencias para el involucramiento de terceras personas en la intervención

2.3 Gestión del desastre y logística humanitaria

La respuesta ante un desastre es un proceso complejo que involucra extrema presión sobre el tiempo, gran incertidumbre y muchos agentes involucrados (Ortuño et al., 2013). También involucra la participación de un número de entidades autónomas para la mitigación, preparación, respuesta y recuperación ante diversos peligros para la sociedad. Los agentes involucrados difieren dependiendo del tipo de desastre, las consecuencias del desastre y el lugar donde acontece, debido a la vulnerabilidad, y se pueden clasificar en tres niveles que están directamente relacionados a las consecuencias del desastre:

- a. Nivel local: es el primer nivel de respuesta, usualmente a cargo de agencias locales, organizaciones de sociedad civil y defensa civil. Típicamente, este nivel de emergencia no es declarado como desastre.

- b. Nivel nacional: el ejército y defensa civil a nivel nacional, organizaciones gubernamentales y ONGs están usualmente involucradas cuando una emergencia está definida como desastre. Algunas veces, organizaciones internacionales también participan a este nivel.
- c. Nivel internacional: comprende gobiernos extranjeros y organizaciones intergubernamentales, ONGs internacionales y agencias de las Naciones Unidas. Este nivel es alcanzado cuando la capacidad nacional de respuesta no es suficiente, debido a la magnitud del desastre o a la vulnerabilidad del país, y el gobierno nacional autoriza una operación humanitaria internacional.

Los procesos de toma de decisiones en la gestión de desastres son por tanto extremadamente difíciles, debido a los múltiples actores involucrados y la complejidad de las tareas a realizar. Dentro de dichas tareas, todos los procesos de planificación, implementación y control de almacenamiento y flujo de recursos y materiales, así como información relevante, desde el punto de origen al punto de consumo para el propósito de satisfacer los requerimientos de bienestar y alivio del sufrimiento de personas vulnerables se conoce como logística humanitaria.

Las principales características que diferencian las cadenas de suministro humanitaria en el contexto de gestión de desastres con las cadenas de suministro en negocios son las siguientes:

- Demanda impredecible en términos de tiempo, ubicación geográfica, tipo y cantidad de recursos de demanda.
- Corto tiempo de respuesta y demanda súbita de grandes cantidades de una amplia variedad de productos y servicios.
- Falta de recursos iniciales en términos de suministros, recursos humanos, tecnología, capacidad y financiamiento.
- Presencia de múltiples actores en la toma de decisiones que pueden resultar difíciles de identificar.

Asimismo, se pueden identificar cuatro etapas básicas en el ciclo de la gestión de ayuda humanitaria antes una emergencia:

- a. Etapa de mitigación, la cual tiene lugar antes del desastre, y tiene como objetivo desarrollar acciones preventivas que disminuyan los efectos de un posible desastre en el futuro.

- b. Etapa de preparación, donde se aborda la planificación con el objetivo de anticiparse a un posible desastre.
- c. Etapa de respuesta, la cual tiene lugar durante el desastre y los períodos inmediatamente posteriores, e incluyen la valoración inicial del impacto de la crisis y la organización de las acciones de ayuda y atención de damnificados.
- d. Etapa de rehabilitación, la cual acontece después del desastre, y se caracteriza por el alto nivel de interacción con los proyectos de cooperación para el desarrollo con el fin de mejorar la condición de vida de las comunidades afectadas.

2.4 Logística humanitaria en el Perú

Según el Centro Nacional de Estimación, Prevención y Reducción de Riesgo de Desastres (CENEPRED) y el Ministerio de Transporte y Comunicaciones, durante una emergencia se colocan los recursos de personal y materiales a disposición, coordinando con el Instituto Nacional de Defensa Civil (INDECI) las operaciones de ayuda necesaria y aplicando las medidas previstas en un plan preestablecido. Asimismo, dichas organizaciones tienen como responsabilidad evaluar los daños que se pudieran haber ocasionado en las redes viales y habilitar caminos de emergencia en caso se encuentren bloqueados o con estructuras colapsadas (INDECI, 2008).

Actualmente, la cadena de abastecimiento humanitaria en el Perú es gestionada por el Sistema Nacional de Gestión de Riesgo de Desastres (SINAGERD), que es el responsable técnico de coordinar, facilitar y supervisar la formulación e implementación del Plan Nacional de Gestión de Riesgo de Desastres, que comprende los procesos de estimación, prevención, reducción del riesgo y reconstrucción. Asimismo, se encarga de asesorar, elaborar y establecer los lineamientos técnicos y mecanismos para el desarrollo adecuado de los procesos previamente mencionados por los distintos entes públicos y privados que integran el SINAGERD.

2.5 Optimización Combinatoria

La Optimización Combinatoria es una rama de la optimización (Papadimitriou & Steiglitz, 1982) donde el conjunto de posibles soluciones es discreto o se puede reducir a un conjunto discreto. Para este tipo de problemas, a medida que la complejidad del espacio de búsqueda aumenta, el tiempo de ejecución de dichos algoritmos puede aumentar de forma exponencial, haciendo que la solución del problema sea prácticamente inviable.

2.6 Optimización de múltiples objetivos

La optimización de múltiples objetivos aborda un tipo especial de problemas de optimización que involucra más de una función a ser optimizada simultáneamente. Debido a ello, comúnmente se tiene que tomar cuenta de los sacrificios que se tengan que realizar entre objetivos potencialmente conflictivos, pues para problemas no triviales, no existe solución que alcance el valor óptimo para absolutamente todos los objetivos. Por tanto, usualmente existe un conjunto de soluciones que no puede ser comparado fácilmente comparadas con otras soluciones. Este tipo de soluciones se denominan soluciones no dominadas o soluciones óptimas de Pareto, para las cuales no es posible realizar una mejora en ninguna función objetivo sin sacrificar por lo menos una de las otras funciones objetivo (Konak, Coit, & Smith, 2006).

Mientras que un problema de optimización de un solo objetivo usualmente presenta la siguiente forma:

$$\begin{aligned} \max z &= f(\mathbf{x}) \\ \text{s. t. } g_i(\mathbf{x}) &\leq 0, \quad i = 1, 2, \dots, m \\ \mathbf{x} &\geq 0 \end{aligned}$$

Donde x es el conjunto de variables de decisión, $f(x)$ es la función objetivo y $g(x)$ son funciones que representan las restricciones del problema (s.t. significa “*subject to*” o “sujeto a”, e indica las restricciones en un modelo de optimización), sin pérdida de generalidad, un problema de optimización puede ser formulado como:

$$\begin{aligned} \max \{z_1 = f_1(\mathbf{x}), z_2 = f_2(\mathbf{x}), \dots, z_q = f_q(\mathbf{x})\} \\ \text{s. t. } g_i(\mathbf{x}) &\leq 0, \quad i = 1, 2, \dots, m \\ \mathbf{x} &\geq 0 \end{aligned}$$

2.7 Problema de transporte

El problema de transporte es un problema de optimización cuyo objetivo es distribuir productos de un grupo de puntos de abastecimiento, llamados fuentes, a un grupo de puntos de demanda, llamados destinos, de tal manera que se minimice el costo total de distribución (Hillier & Lieberman, 2015). El presente proyecto de fin de carrera está basado en este problema, el cual presenta la siguiente formulación:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

$$s. t. \sum_{j=1}^n x_{ij} = s_i, \text{ para } i = 1, 2, \dots, m$$

$$\sum_{i=1}^m x_{ij} = d_j, \text{ para } j = 1, 2, \dots, n$$

$$x_{ij} \geq 0, \text{ para todo } i \text{ y } j$$

Donde:

- La función objetivo es el costo total de distribución
- x_{ij} es el número de unidades a ser transportadas desde el origen i al destino j
- c_{ij} es el costo de transportar unidades desde el origen i al destino j
- s_i es el nivel de oferta en el nodo i
- d_i es el nivel de demanda en el nodo i

2.8 Problema de transbordo

El problema de transbordo es una generalización del problema de transporte que permite la inclusión de puntos que no son ni fuentes ni destino, los cuales son denominados puntos intermedios o de transbordo (Hillier & Lieberman, 2015). Como en un escenario real es usual contar con puntos intermedios en una red de distribución, se tomará como base para el presente proyecto de fin de carrera. Este problema presenta la siguiente formulación:

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

$$s. t. \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = b_i, \text{ para cada nodo } i$$

$$0 \leq x_{ij} \leq u_{ij}$$

Donde:

- La función objetivo es el costo total de distribución

- x_{ij} es el número de unidades a ser transportadas desde el origen i al destino j
- c_{ij} es el costo de transportar unidades desde el origen i al destino j
- u_{ij} es la capacidad máxima de unidades que se pueden transportar desde i a j
- b_i es el flujo neto generado en el nodo i
 - o $b_i > 0$, si el nodo i es un nodo de oferta
 - o $b_i < 0$, si el nodo i es un nodo de demanda
 - o $b_i = 0$, si el nodo i es un nodo de transbordo

2.9 Métodos para la solución de problemas de optimización

2.9.1 Métodos exactos

Los métodos exactos para abordar problemas de optimización combinatoria comúnmente se basan en resolver la formulación de dichos problemas como programas lineales. Es importante mencionar que, dado que los modelos cuentan con variables enteras, se tienen que considerar métodos de resolución de programación lineal entera mixta, los cuales son los siguientes (Toth & Vigo, 2002):

- Algoritmos de ramificación y poda: consiste en recorrer todo el espacio de búsqueda para encontrar la solución, con la mejora adicional de evitar soluciones que se están alejando del valor óptimo
- Algoritmos de ramificación y corte: consiste en realizar una relajación del programa lineal desconsiderando las restricciones de integralidad para generar una solución con métodos para programas lineales comunes, como el método simplex. Posteriormente, se aplica un algoritmo de planos de corte para volver a acotar las restricciones de integralidad
- Algoritmos basados en recubrimiento de conjuntos: consiste en modificar la formulación del programa lineal para adaptarlo en uno basado en el clásico problema de recubrimiento de conjuntos, para resolverlo con métodos aplicables a dicho problema

2.9.2 Métodos heurísticos

Los métodos heurísticos consisten en explotar las características de cada problema en particular para generar soluciones mediante métodos voraces con una calidad aceptable aproximándose al valor óptimo. Una gran ventaja de estos métodos es su simplicidad, lo cual los hace bastante rápidos, permitiéndoles resolver problemas intratables mediante métodos exactos, pero su principal desventaja es el peligro de hallar un óptimo local alejado del global que impida que el algoritmo intente buscar una solución mejor

(Schrijver, 2003). Como se mencionó anteriormente, dichos métodos dependen del problema a resolver, y en el caso del problema de enrutamiento, programación de tareas o mochila, heurísticas de construcción y heurísticas de mejora son dos de los tipos de heurísticas más usados.

2.9.3 Métodos metaheurísticos

La principal característica y motivación de los métodos metaheurísticos es su capacidad de evitar óptimos locales, debido a que evitan depender totalmente del problema, presentándose como métodos de uso general. Debido a ello, comúnmente obtienen soluciones de mucha mejor calidad que los métodos heurísticos. Sin embargo, la mayor complejidad de estos métodos los hace más lentos, por lo que su requerimiento computacional es mayor (Luke, 2015). Dentro de los métodos más conocidos se encuentran:

- Algoritmos Genéticos
- Búsqueda Tabú
- Recocido Simulado
- GRASP (Greedy Randomized Adaptive Search Procedure)

Para conseguir evitar óptimos locales, los métodos metaheurísticos realizan un balance entre explotación, o mejorar iterativamente la mejor solución actual (lo que realiza una heurística), y exploración, para buscar en otras ubicaciones del espacio de búsqueda (Luke, 2015). Esta optimización puede realizarse por un individuo a la vez (Búsqueda Tabú, GRASP o Recocido Simulado) o por población (Algoritmos Genéticos, Colonias de Hormigas, Enjambre de Partículas). Por ejemplo, el funcionamiento de la metaheurística de recocido simulado es la siguiente:

1. Inicializar temperatura con un valor alto
2. Crear una solución inicial
2. Repetir
 - a. Realizar una modificación a la solución actual
 - b. Si la nueva solución es mejor que la original o si un número aleatorio supera la temperatura actual, reemplazar la solución actual con la nueva solución
 - c. Disminuir la temperatura
3. Hasta algún criterio de convergencia o máximo número de generaciones

Como se puede ver, en el recocido simulado existe una tendencia a exploración al comienzo del algoritmo, permitiendo explorar soluciones alejadas de la mejor solución actual de forma aleatoria, cuando la temperatura es alta aún, y a medida que la temperatura disminuye, solo se reemplazará la solución actual si la nueva solución es estrictamente mejor que la actual.

2.9.3.1 Algoritmos Genéticos

Un algoritmo genético es una metaheurística y técnica de búsqueda global que resuelve problemas imitando procesos observados durante la evolución natural. Este paradigma para resolver problemas fue propuesto inicialmente por John Holland en los 70. Un algoritmo genético puro es un método de resolución de problemas que usa poca información sobre el dominio del problema. Por tanto, puede ser aplicado a un amplio rango de problemas holgadamente definidos que no permiten el uso de métodos especializados (Yang, 2014). En esencia, un algoritmo genético evoluciona una población de cadenas de bits o cromosomas generada aleatoriamente, donde cada cromosoma codifica una solución para una instancia particular. Esta evolución toma lugar a través de la aplicación de operadores que imitan los fenómenos naturales observados en la naturaleza como la reproducción y mutación, los cuales son aplicados a lo largo de un número determinado de iteraciones o generaciones. El esquema del algoritmo es el siguiente:

1. Generar la población inicial
2. Repetir
 - a. Seleccionar dos cromosomas de la población actual
 - b. Generar un nuevo cromosoma a partir de los dos anteriores
 - c. Aplicar una mutación aleatoria al nuevo cromosoma probabilísticamente
 - d. Reemplazar algún cromosoma de la población actual con el nuevo cromosoma generado
3. Hasta algún criterio de convergencia o máximo número de generaciones

Capítulo 3. Estado del Arte

3.1 Introducción

Para realizar la revisión del estado de arte, se utilizaron cadenas de búsqueda considerando los siguientes criterios:

- Investigación académica y productos comerciales
- Optimización con un objetivo y multiobjetivo
- Optimización con métodos exactos, heurísticos y metaheurísticos
- Optimización en el contexto de ayuda humanitaria

Las bases utilizadas fueron Wiley Online Library, ACM Digital Library, SpringerLink, ScienceDirect y Google Scholar.

3.2 Investigaciones Académicas

3.2.1 Transporte en operaciones de respuesta a desastres (*Transportation in disaster response operations*)

Djamel Berkoune, Jacques Renaud, Monia Rekik y Angel Ruiz (2012) de diferentes universidades e institutos de Canadá desarrollan dos propuestas para resolver el problema de enrutamiento con múltiples productos y múltiples productos que apunta a minimizar la duración total de los viajes: branch-and-bound clásico usando el conocido solver comercial CPLEX con un criterio heurístico de parada, y una construcción heurística rápida para generar soluciones factibles acompañada de un algoritmo genético que usa algunas de las soluciones generadas por el algoritmo heurístico. Los algoritmos fueron probados sobre conjuntos de datos de tamaño entre 20 y 60. Para las instancias más grandes, el nivel de logro del algoritmo genético sobre el valor objetivo es de 99.28% para una población de 300 individuos durante 300 generaciones en un tiempo de 63 segundos, mientras que el método branch-and-bound alcanzó el valor óptimo en 600 segundos.

3.2.2 Enrutamiento para esfuerzos de ayuda humanitaria (*Routing for relief efforts*)

Ann Campbell, Dieter Vandenbussche y William Hermann (2008) de diferentes universidades en Estados Unidos introducen y analizan dos funciones objetivo para el problema del viajante y el problema de enrutamiento, considerando el tiempo de llegada al destino, en orden de calzar con naturaleza humanitaria de la distribución. La primera

función objetivo minimiza el máximo tiempo de llegada y la segunda función objetivo minimiza el tiempo promedio de llegada. Los problemas son resueltos mediante heurísticas de inserción y búsqueda local y fueron probados sobre los conjuntos de datos de acceso en línea Augerat-A, Augerat-B y Golden, los cuales contienen diversas instancias con un tamaño entre 30 y 70 puntos de demanda. Para comparar los resultados, se usó un modelo de programación entera mixta (MIP) que fue ejecutado por 6 horas. Los resultados están disponibles gráficamente, los cuales se muestran a continuación:

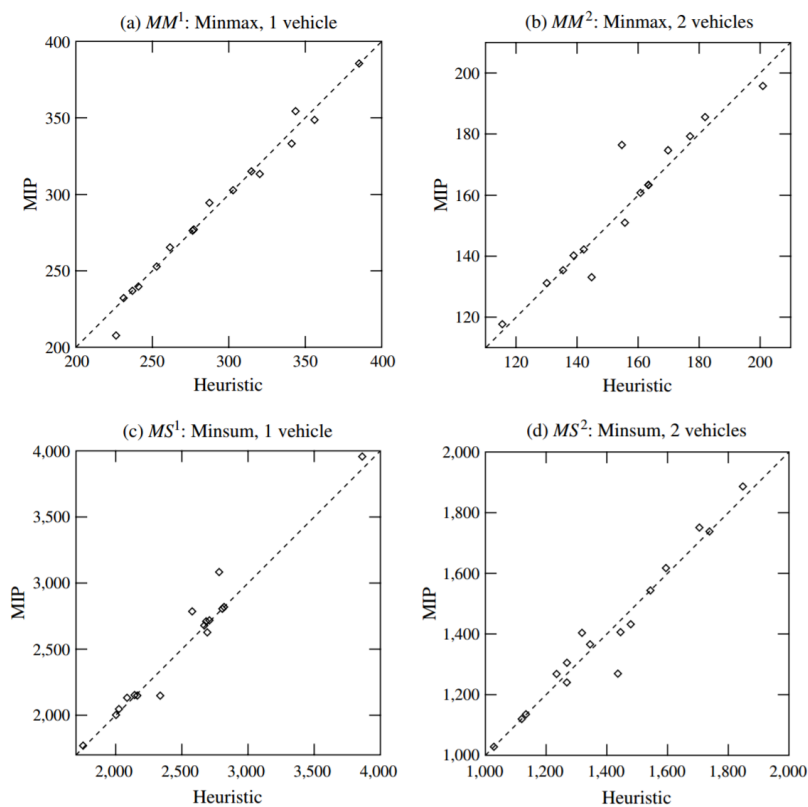


Ilustración 2. Comparación entre resultados de programación entera mixta y heurística (Campbell et al., 2008).

3.2.3 Modelos para enrutamiento de ayuda humanitaria: Equidad, eficiencia y eficacia (*Models for relief routing: Equity, efficiency and efficacy*)

Michael Huang, Karen Smilowitz y Burcu Balcik (2011) de universidades en Estados Unidos y Turquía proponen tres funciones objetivos para un problema de enrutamiento considerando el costo, velocidad y equidad de distribución. Los tres problemas resultantes son resueltos con diferentes metaheurísticas basadas en GRASP. Los

resultados son probados en instancias de tamaño 8 a 10 y se consigue un valor de logro sobre el objetivo principal de 0.87.

3.2.4 Óptima reubicación de stock bajo incertidumbre en operaciones humanitarias post-desastre (*Optimal stock relocation under uncertainty in post-disaster humanitarian operations*)

Alexander Blecken, Beate Rottkemper y Bernd Hellingrath (2010) de diferentes universidades de Alemania aplican métodos exactos y heurísticos para afrontar el problema de aprovisionamiento de almacén y flujo de ayuda humanitaria minimizando el costo total. Como método exacto se resolvió el programa lineal con un método exacto (*solver*), y como método aproximado se usó una heurística basada en un árbol de decisión acompañada de una función de utilidad. Los métodos se aplicaron sobre un conjunto de datos compuesto de 6 regiones de Myanmar, con una distribución sobre 13 períodos de tiempo. Los resultados se pueden apreciar en el siguiente gráfico:

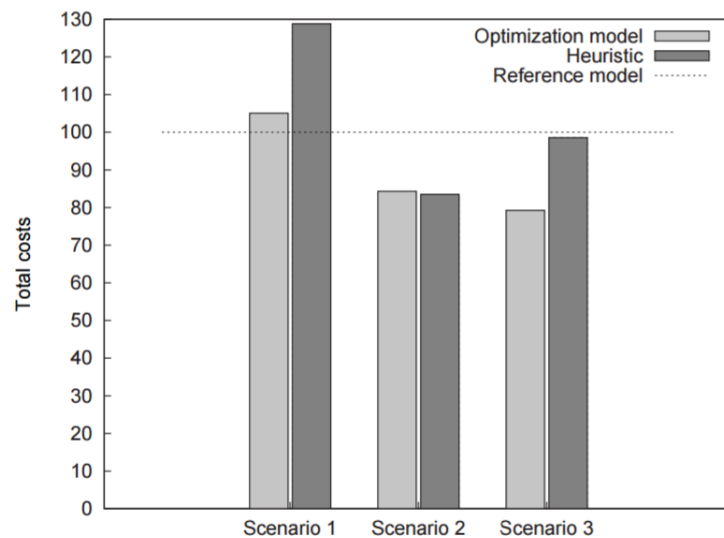


Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010).

3.2.5 Planificación óptima multi-objetivo para el diseño de sistemas de distribución de ayuda humanitaria (*Multi-objective optimal planning for designing relief delivery systems*)

Gwo Tzeng, Hsin Cheng y Tsung Huang (2007) de diferentes universidades e institutos en Taiwan usan una técnica para la resolución de programas lineales multi-objetivo conocida como programación difusa o fuzzy programming para abordar el VRP considerando una optimización sobre múltiples períodos a la vez en su función objetivo,

minimizando el costo total en primer lugar, el tiempo de viaje total en segundo lugar, y maximizando la satisfacción mínima durante el período de planeamiento en tercer lugar. El algoritmo fue probado sobre 8 puntos de demanda, 5 puntos de distribución y 4 puntos de transbordo que representan zonas de Taiwan y el ranking de nivel de logro estos tres valores objetivo fue de 0.93, 0.82 y 0.65.

3.2.6 Metaheurística bi-objetivo para la planificación de operaciones de ayuda humanitaria en desastres (*Bi-objective metaheuristic for disaster relief operation planning*)

Pamela Nolz, Karl Doerner, Walter Gutjahr y Richard Hartl (2010) de la Universidad de Viena presentan una solución híbrida basada en algoritmos genéticos, búsqueda de vecindario variable y enlace de caminos para abordar un problema multicriterio de distribución de agua afectada por una catástrofe. Los dos criterios contemplados son: la duración del viaje, y el último tiempo de llegada de un vehículo a un punto de demanda. El algoritmo es probado en data real de la provincia de Manabí en Ecuador contando entre 20 y 40 puntos de parada y el nivel de logro estos tres valores objetivo fue de 0.75 y 0.52.

3.2.7 Modelo de optimización multi-criterio para la distribución de ayuda humanitaria (*A multi-criteria optimization model for humanitarian aid distribution*)

Vitoriano Begoña, Teresa Ortuño, Gregorio Tirado y Javier Montero (2011) de la Universidad Complutense de Madrid proponen un modelo de doble flujo multi-criterio para abordar el problema de distribución de suministros a la población afectada por un desastre. Criterios como el costo, tiempo de respuesta, equidad de distribución, seguridad y confiabilidad son considerados en conjunto usando un enfoque en programación por objetivos. El modelo fue aplicado durante el terremoto de Haití acontecido en 2010. La red de transporte consistió en 24 nodos y 42 enlaces entre ubicaciones. El resultado se presenta mediante una matriz de recompensa que presenta los niveles de logro de los valores objetivos considerando una prioridad diferente para cada objetivo en cada fila.

Attribute	COST(\$)	TX(min.)	DX	PRI ₁₃	RMN	RG(-ln)	PX	PG(-ln)
COST	35,835.0	141.75	1.0	0.00	0.1	5.83	0.85	11.76
TX	79,204.5	83.85	1.0	0.13	0.1	7.28	0.92	19.32
DX	42,453.0	146.25	0.4	0.60	0.1	7.76	0.92	15.37
PRI ₁₃	39,498.0	141.75	1.0	1.00	0.1	4.32	0.88	12.32
RMN	79,931.5	159.75	1.0	0.90	0.75	2.07	0.92	19.60
RG	69,692.0	124.50	1.0	0.00	0.75	1.43	0.85	11.87
PX	80,000.0	117.00	1.0	0.00	0.1	4.73	0.19	9.08
PG	79,966.0	156.00	1.0	0.00	0.1	4.22	0.66	7.80

Ilustración 4. Matriz de recompensa para cada objetivo de optimización (Vitoriano, Ortuño, Tirado, Montero, et al., 2011).

3.3 Productos comerciales

3.3.1 SimpliRoute

SimpliRoute es un ejemplo de productos comerciales modernos, pues se trata de un emprendimiento tecnológico (*startup*) que ofrece una herramienta de planificación de rutas mediante software como servicio basado en interfaces de programación de aplicaciones, o API-based SaaS (Software as a Service) (SimpliRoute, 2018). SimpliRoute comenzó en el 2015 y su modelo de optimización considera 4 restricciones en consideración:

- Capacidades de los vehículos
- Diferentes puntos de inicio para los vehículos
- Cambios de conductor
- Ventanas de tiempo para los clientes

Asimismo, cuenta con integración con los servicios Waze y Google Maps, lo cual permite realizar un monitoreo o tracking en tiempo real.

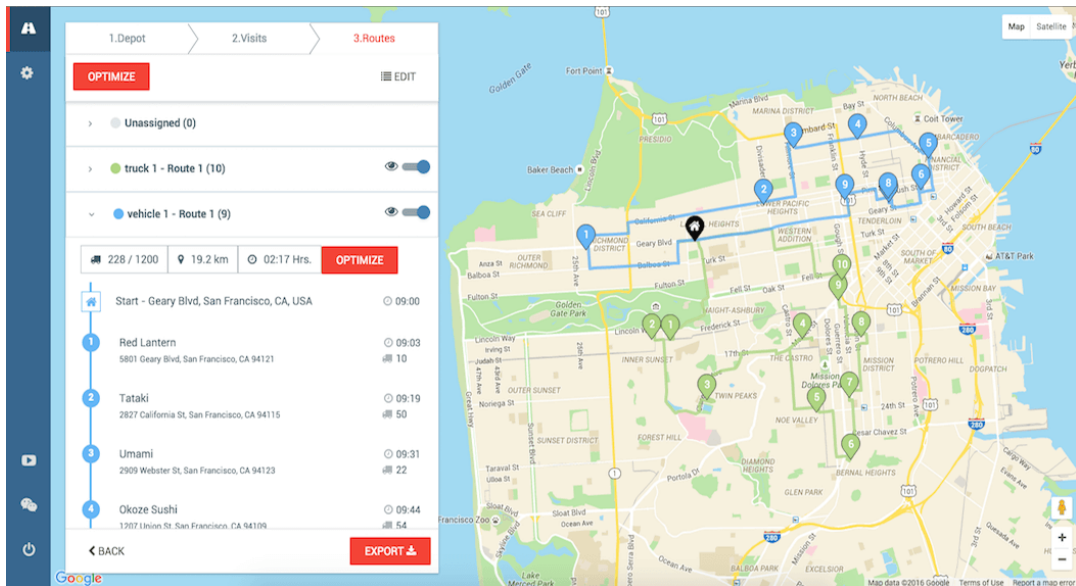


Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018).

3.3.2 SAP Transportation Management VSR Optimization

VSR Optimization es un servicio del módulo de Transportation Management del popular sistema SAP, el cual permite asignar unidades de carga a capacidades, que puedan pertenecer a vehículos, por ejemplo, de un modo eficiente, considerando las siguientes restricciones:

- Capacidades de los vehículos
- Ventanas de tiempo para los vehículos
- Ventanas de tiempo para los clientes
- Tiempo de expiración de productos
- Duración del viaje
- Distancia recorrida
- Cantidad de paradas intermedias

Dado que es parte del mismo ecosistema SAP, la integración con módulos de ventas y de gestión de la cadena de suministro no presenta ningún problema (SAP, 2018).

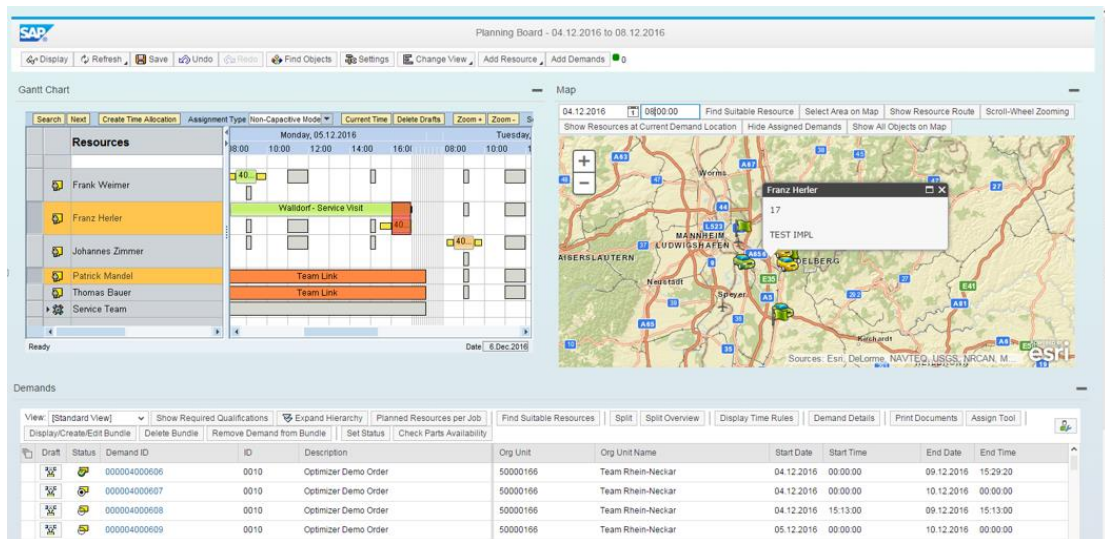


Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018).

3.3.3 Google Optimization Tools

Google Optimization Tools es una suite gratuita para problemas de optimización combinatoria, la cual consiste en una interfaz unificada para variados problemas de programación lineal y entera, incluyendo problemas de asignación, empaquetamiento y enrutamiento (Google, 2018). El modelo de VRP contempla las siguientes restricciones:

- Capacidad de ubicaciones y vehículos
- Máximo número de ubicaciones a visitar por vehículo
- Restricciones de tiempo y distancia
- Ventanas de tiempo
- Relaciones de precedencia entre pares de ubicaciones

Asimismo, la suite presenta herramientas para la resolución del problema de la mochila o knapsack, que considera como criterios el peso de los productos y el valor de los productos, así como el problema de programación de tareas o scheduling, que considera tareas secuenciales e indivisibles.

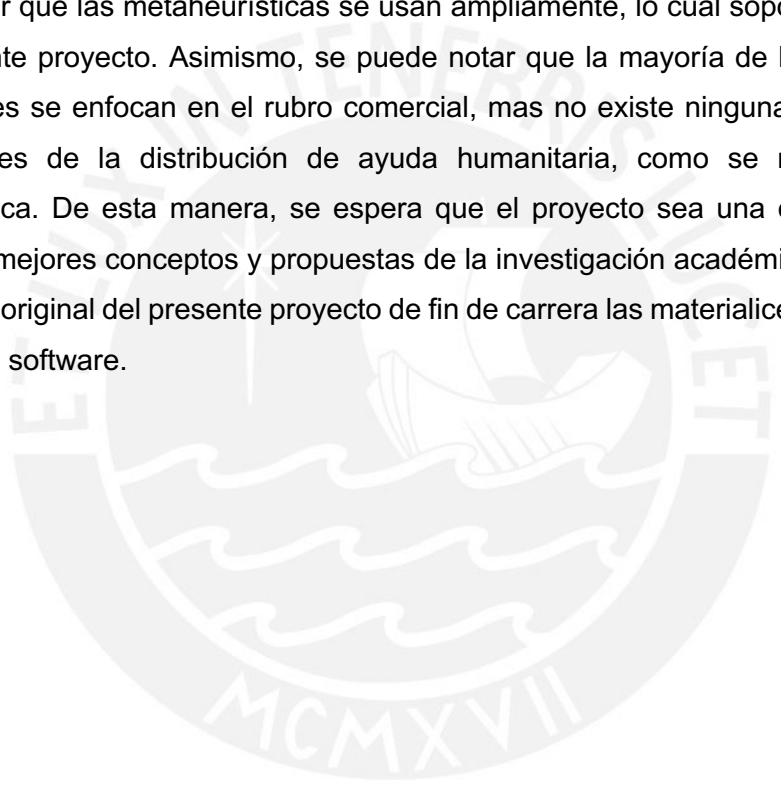
3.4 Revisión y discusión

Se consigue apreciar que los trabajos de investigación sobre optimización de logística humanitaria son variados, y se distinguen por el tipo de optimización que realizan, considerando tanto optimización sobre un objetivo como múltiples objetivos. Asimismo, se ha identificado que los métodos más utilizados son las metaheurísticas, métodos exactos de programación entera mixta y los métodos basados en lógica difusa o fuzzy.

Finalmente, se ha logrado identificar un conjunto de productos y herramientas comerciales muy conocidos y centrados en la resolución del problema de enrutamiento de vehículos principalmente, así como el problema de la mochila o knapsack.

3.5 Conclusiones

A partir de la revisión de literatura, se ha obtenido un panorama más claro de la investigación y herramientas existentes actualmente, con el fin de tener una referencia al momento de desarrollar el presente proyecto de fin de carrera. En particular, se ha logrado ver que las metaheurísticas se usan ampliamente, lo cual soporta la propuesta del presente proyecto. Asimismo, se puede notar que la mayoría de las herramientas comerciales se enfocan en el rubro comercial, mas no existe ninguna adaptada a las necesidades de la distribución de ayuda humanitaria, como se mencionó en la problemática. De esta manera, se espera que el proyecto sea una contribución que reúna los mejores conceptos y propuestas de la investigación académica y junto con la propuesta original del presente proyecto de fin de carrera las materialice a través de una interfaz de software.



Capítulo 4. Definición de la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria

4.1 Introducción

Para poder desarrollar algoritmos para optimizar la distribución de ayuda humanitaria primero se debe plantear el problema para conocer qué se desea optimizar, qué parámetros describen una instancia del problema, cuáles son las variables que describen una posible solución y cuáles son las restricciones que actúan sobre estas variables

4.2 Resultado

4.2.1 Parámetros del problema de optimización

4.2.1.1 Red de distribución

Uno de los elementos principales del problema es la red de distribución, la cual involucra representar un grafo mediante el número de nodos (K) y aristas entre los nodos, la cual consiste en un conjunto de pares ordenados (j, k) que indican que el nodo j está conectado con el nodo k .

4.2.1.2 Costo de transporte

El costo de transporte está vinculado a las características de la red de distribución, debido a que el costo es directamente proporcional a la distancia entre cada par de puntos. De esta manera, a través de este parámetro se pueden representar caminos de distribución que sean más eficientes que otros. Para ello se establece el parámetro **CostoTransp_{j,k}**, el cual indica el costo unitario de transporte para transportar un bien desde el nodo j al nodo k .

4.2.1.3 Oferta y demanda en los nodos de la red de distribución

Las condiciones iniciales del problema deben indicar cuánta oferta y cuánta demanda de los bienes de distribución existe en cada nodo. En el contexto de distribución de ayuda humanitaria, la oferta se entiende como la ayuda humanitaria disponible en almacenes y centros de distribución y demanda se entiende como la necesidad existente en cada ubicación en la zona de desastre. La oferta de bienes se define por el parámetro **Inventario_k**, que representa la cantidad de bienes que existe en el nodo k , mientras que

la demanda de bienes se define por el parámetro ***Demanda_k***, que representa la cantidad de bienes que se requieren en el nodo k.

Para considerar bienes de diferentes tipos, estos parámetros se pueden expandir a ***Inventario_{ki}*** y ***Demanda_{ki}***, que indican la oferta y demanda de bienes de tipo i en el nodo k.

4.2.1.4 Capacidad de almacenes

La capacidad máxima de bienes que se pueden transportar está limitada por la capacidad de los puntos intermedios entre los almacenes generales y los puntos finales de demanda. Para ello se establece el parámetro ***Capacidad_k***, que indica la capacidad máxima del nodo k, espacio que será compartido por todos los tipos de bienes.

4.2.2 Variables de decisión

4.2.2.1 Cantidad de bienes que se transportan

Para representar una posible solución del problema, se requiere una variable de decisión que establezca cuántos bienes serán transportados entre un determinado par de nodos, lo cual queda origina la variable de decisión que indique cuántos bienes serán transportados en los vehículos desde el nodo j al nodo k (***Transp_{jk}***). Para considerar bienes de diferentes tipos, esta variable se puede expandir a ***Transp_{jki}***, que indica la cantidad de bienes de tipo i que serán transportados desde el nodo j al nodo k.

4.2.3 Función objetivo

4.2.3.1 Minimizar costo de transporte

El primer objetivo a considerar en la optimización multiobjetivo está orientado a satisfacer la demanda en los nodos de la manera más eficiente posible, para no desperdiciar recursos que podrían ser utilizados para atender a más nodos. Por ello, el valor que será minimizado es la suma total del producto de la cantidad de bienes transportados entre cada par de nodos por el costo unitario de transporte en dicha vía. De esta manera, una posible solución que llegue a su destino por una ruta más corta será preferible a una solución que movilice a los bienes por rutas innecesarias. La siguiente expresión indica la suma del costo de transporte unitario por la distancia entre los nodos j y k, multiplicada por la cantidad de bienes de tipo i que se desplacen desde j hacia k:

$$\sum_{j,k,i} \text{CostoTransp}_{j,k} * \text{Transp}_{j,k,i}$$

4.2.3.2 Minimizar inequidad en la distribución

Uno de los objetivos propios de la distribución de ayuda humanitaria es que no solo es importante considerar que la ayuda se envíe al menor costo, sino también que, en situaciones donde la cantidad de bienes no puede cubrir toda la demanda existente, la distribución se realice de la manera más equitativa posible, pues esto está directamente relacionado con el bienestar de la población, el cual es el objetivo principal de la respuesta ante un desastre.

Primeramente, para tomar en cuenta que la cantidad de demanda de cada punto es variable, es necesario trabajar con porcentajes de satisfacción, lo cual puede ser medido con una división entre la cantidad de bienes repartidos en un punto y la demanda de dicho punto. Por otra parte, para contar con una función objetivo que distinga entre un reparto equitativo y uno no equitativo, es importante notar que un objetivo lineal como el promedio de los porcentajes de satisfacción no tiene la capacidad de distinguir equidad en la distribución. Para ello, considere el siguiente ejemplo donde se tienen 4 puntos con demanda de 40 bienes, pero solo se tienen 40 bienes disponibles para distribuir.

La solución que realice un reparto equitativo destinaría 10 bienes a cada punto, lo cual haría que cada nodo tenga un porcentaje de satisfacción de 25% (10 / 40). El promedio simple de estos 4 porcentajes es 25% ($\frac{0.25+0.25+0.25+0.25}{4}$). Por otro lado, una solución que no realice un reparto equitativo y destine los 40 bienes a un solo punto haría que uno de los nodos tenga un porcentaje de satisfacción de 100% y los demás 0%. Ahora, resulta que el promedio simple de estos 4 porcentajes sigue siendo 25% ($\frac{1+0+0+0}{4}$). Por ello, el promedio simple no distingue la diferencia de equidad en la distribución.

Para lograr solucionar este hecho, es posible usar el promedio de los cuadrados de los porcentajes de satisfacción. Debido a que la función cuadrática crece de forma más rápida que la función identidad, esta función puede penalizar de manera más drástica a soluciones altamente no equitativas. Para el mismo ejemplo planteado anteriormente, la primera solución que realiza un reparto equitativo tendría un promedio de cuadrados de 0.0625, y llevándolo a la escala original tomando la raíz cuadrada, resulta en un

promedio de 25% ($\sqrt{\frac{0.25^2+0.25^2+0.25^2+0.25^2}{4}}$). Por otro lado, la solución que no realiza un reparto equitativo tiene un promedio de cuadrados de 0.25, y de la misma manera, llevándolo a la escala original tomando la raíz cuadrada, resulta en un promedio de 50% ($\sqrt{\frac{1^2+0^2+0^2+0^2}{4}}$), el cual es un valor mayor a la solución óptima. De hecho, se puede notar que cualquier otra combinación de distribución obtendrá un promedio mayor o igual a 25%. De esta manera, con este tipo de promedio, sí es posible distinguir la diferencia entre un reparto equitativo y uno no equitativo.

Con todo esto, la función objetivo queda definida por la siguiente expresión, que indica el promedio de los cuadrados de los porcentajes de satisfacción de cada tipo de bien i en cada punto de demanda k :

$$\sum_{k,i} \left(\frac{\sum_j Transp_{j,k}}{Demanda_k} \right)^2$$

Como se puede apreciar, la cantidad de bienes entregados al nodo k se obtiene sumando los bienes transportados desde todos los nodos j al nodo k .

4.2.4 Restricciones

4.2.4.1 Balance en los puntos de abastecimiento

Como en todo problema de transporte, es necesario asegurar que solo se puedan transportar bienes que se encuentren en el inventario, para los nodos que representan almacenes. Esto queda descrito por la siguiente restricción, que indica que para todos los almacenes j la suma de la cantidad de bienes de tipo i que sean transportados desde el nodo j hacia los demás nodos adyacentes k debe ser equivalente al inventario total en dicho nodo j :

$$\sum_k Transp_{j,k,i} = Inventario_{j,i} \quad , \text{ para todo almacén } j \text{ y tipo de bien } i$$

En casos donde la cantidad de bienes disponible es mayor a la demanda total, para mantener la equivalencia de la restricción, se incluyen nodos artificiales que absorban el excedente de bienes, representando una “demanda ficticia”. Esta es una técnica ampliamente usada en problemas de transporte, la cual se aplica como preprocesamiento de datos, siendo transparente al momento de evaluar los resultados, pues no se les asigna un costo de transporte (Hillier & Lieberman, 2015).

4.2.4.2 Balance en los puntos de demanda

De la misma manera, es necesario asegurar que los bienes que sean transportados a los puntos de demanda sean equivalentes a la demanda total existente en dichos nodos. Esto queda descrito por la siguiente restricción, que indica que para todos los puntos de demanda k la suma de la cantidad de bienes de tipo i que sean transportados hacia el nodo k desde los demás nodos adyacentes j debe ser equivalente a la demanda total en dicho nodo k :

$$\sum_j Transp_{j,k,i} = Demanda_{k,i} \quad , \text{ para todo punto de demanda } k \text{ y tipo de bien } i$$

De manera similar al balance en los almacenes, en casos donde la demanda supera la cantidad de bienes disponibles, es posible introducir nodos de abastecimiento artificiales que cubran la demanda excedente, representando un “abastecimiento ficticio”. Nuevamente, esta técnica de preprocesamiento es transparente al problema original.

4.2.4.3 Balance en los puntos de transbordo

Además de los almacenes y puntos de demanda, existen puntos intermedios o puntos de transbordo que forman parte de la red de distribución. Para estos puntos, es necesario establecer una equivalencia en el flujo de transporte, en otras palabras, que la cantidad de bienes que ingresa a un nodo de transbordo es equivalente a la misma cantidad de bienes que sale del nodo. Esto queda representado por la siguiente restricción, la cual expresa que para cada nodo de transbordo k , la suma de bienes de tipo i que ingresa desde los nodos adyacentes j es igual a la cantidad de bienes de tipo i que sale hacia los nodos adyacentes m :

$$\sum_j Transp_{j,k,i} = \sum_m Transp_{k,m,i} \quad , \text{ para todo punto de transbordo } k \text{ y tipo de bien } i$$

4.2.4.4 Evitar sobrepasar la capacidad de los almacenes

La cantidad de bienes que serán transportados será limitada también por la capacidad de los almacenes y puntos de transbordo. Esto queda descrito por las siguientes restricciones, que establecen que la cantidad total de bienes (de cualquier tipo i) que se transporten de j a k no debe ser superior a la capacidad total del nodo k (capacidad de entrada), y que la cantidad total de bienes (de cualquier tipo i) que se transporten de k a m no debe ser superior a la capacidad total del nodo k (capacidad de salida):

$$\sum_{j,i} \text{Transp}_{j,k,i} \leq \text{Capacidad}_k, \text{ para todo nodo } k$$

$$\sum_{m,i} \text{Transp}_{k,m,i} \leq \text{Capacidad}_k, \text{ para todo nodo } k$$

4.2.5 Rango de existencia

Dado que todas las variables y parámetros a excepción de los parámetros referentes a costo de transporte representan cantidades discretas, el rango de existencia es el conjunto de los números enteros no negativos: $Z_{\geq 0}$. El rango de existencia de los parámetros de costo de transporte es el conjunto de los números reales no negativos: $R_{\geq 0}$.

4.3 Relación con el objetivo

Esta formulación describe en su totalidad el problema que se plantea resolver en el proyecto de tesis mediante algoritmos de optimización metaheurísticos y algoritmos de optimización de programas no lineales. Para cada uno de los algoritmos, se requerirá adaptar esta formulación según corresponda.

Capítulo 5. Definición de las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria

5.1 Introducción

Una vez formulado el problema, debemos definir las estructuras de datos que representarán los parámetros y variables del problema, para que sean utilizadas durante la ejecución del algoritmo genético. Una vez definidas, se presentará la definición del cromosoma.

5.2 Resultado

5.2.1 Estructuras de datos para los parámetros iniciales y variables auxiliares del problema

El resumen de la lista de parámetros y variables auxiliares definidas en el capítulo anterior es la siguiente:

- Parámetros
 - **K**: número de nodos en la red de distribución
 - **I**: número de tipos de bienes (suministros) de ayuda humanitaria
 - **Inventario_{ki}**: inventario total de bienes de tipo i en el nodo k
 - **Demanda_{ki}**: demanda total de bienes de tipo i en el nodo k
 - **Capacidad_k**: capacidad total del nodo de distribución k
 - **CostoTransp_{jk}**: costo de transporte unitario entre los nodos j y k

Debido a que los primeros dos parámetros son escalares (K, I), su representación será la misma. Asimismo, el parámetro **CostoTransp** será representado por una matriz de distancias de dimensiones [K, K]:

Arreglo de reales [K, K] **CostoTransp**;

Los parámetros que indiquen propiedades de cada nodo de la red de distribución (**Demanda**, **Inventario**, **Capacidad**) serán agrupados en la clase nodo:

```
Clase Nodo {  
    Entero idNodo;  
    Arreglo de enteros [K, I] Demanda;  
    Arreglo de enteros [K, I] Inventario;
```

Arreglo de enteros [K] *Capacidad*;
}

5.2.2 Variable de decisión y cromosoma

La variable de decisión que se formuló en el capítulo anterior es la siguiente:

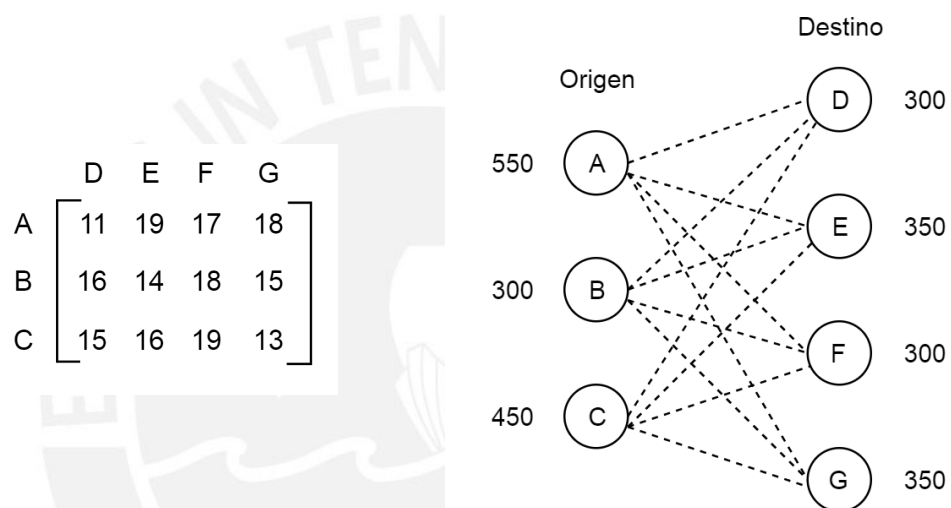
- ***Transp_{jki}***, que indica la cantidad de bienes (suministros de ayuda humanitaria) de tipo *i* que se transportan desde el nodo *j* hacia *k*

Esta variable de decisión es la que genera un plan de distribución y describe una posible solución para el problema de optimización. De acuerdo con primeras investigaciones en el tema de algoritmos genéticos aplicados al problema de transporte (Vignaux & Michalewicz, 1991), una manera inicial de definir el cromosoma es hacer que el cromosoma represente directamente dicha variable de decisión. Sin embargo, decidir representar la matriz de transporte directamente en el cromosoma hace complicado el uso de operadores genéticos como mutación y especialmente cruce, siendo una de las razones principales el hecho de que las redes de transporte no suelen estar completamente conectadas comúnmente, lo cual genera vacíos permanentes en las matrices de transporte, las cuales además deben mantener un equilibrio de transporte (lo que entra es igual a lo que sale), generando una alta probabilidad de crear individuos no factibles (o aberraciones) en un algoritmo genético y requiriendo operadores de reparación. Ante este hecho, ha habido diferentes propuestas para generar una representación más eficiente y compacta del cromosoma, como representar el cromosoma como una lista de aristas de la red de transporte (Sheng, Dechen, & Xiaofei, 2006) y como una lista de prioridades de los nodos de la red de distribución (Gen, Altıparmak, & Lin, 2006). Esta última representación se caracteriza por tener un menor tamaño debido a que la codificación es por nodo en lugar de arista y debido a que codifica un conjunto de prioridades - el cual representa mediante una permutación de números enteros - resulta bastante adecuada para los operadores de cruce y mutación, sin requerir operadores de reparación. Por estos motivos, esta será la representación sobre la cual nos basaremos para la definición del cromosoma de nuestro algoritmo genético, la cual se explica a continuación.

5.2.3 Cromosoma

De acuerdo a lo mencionado en la sección anterior, el cromosoma consiste en una asignación de prioridades a los nodos de la red de transporte. Por ejemplo, sea la

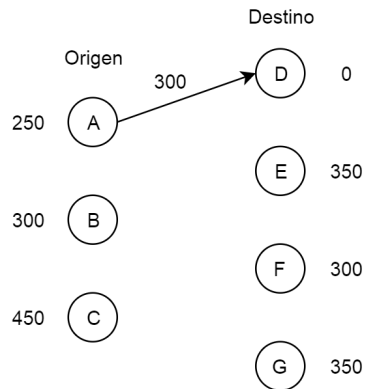
siguiente red de distribución, donde cada círculo representa un nodo de oferta (izquierda) o demanda (derecha), junto con la cantidad de oferta y demanda expresado en valores numéricos (en el contexto de ayuda humanitaria, la oferta son los bienes de ayuda disponibles en almacenes y la demanda es la necesidad de ayuda en puntos de desastre). La matriz de la izquierda representa la matriz de costos de la red de transporte, donde el elemento de la fila i y la columna j de la matriz indica el costo para llegar desde i hasta j :



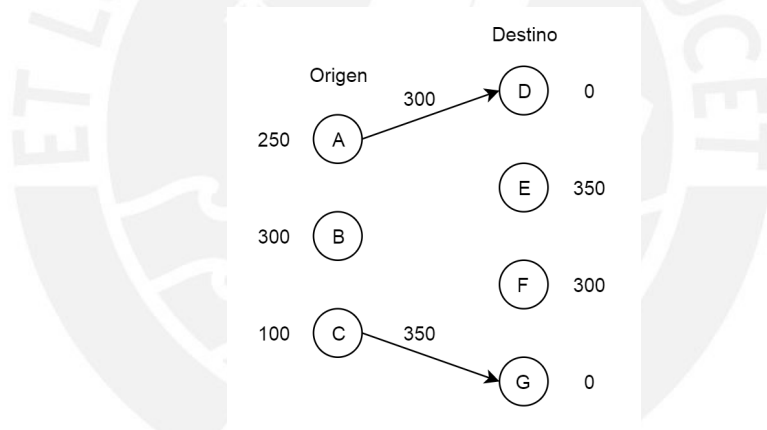
Un posible cromosoma para este problema sería el siguiente:

Nodo:	A	B	C	D	E	F	G
Prioridad:	2	5	3	7	4	1	6

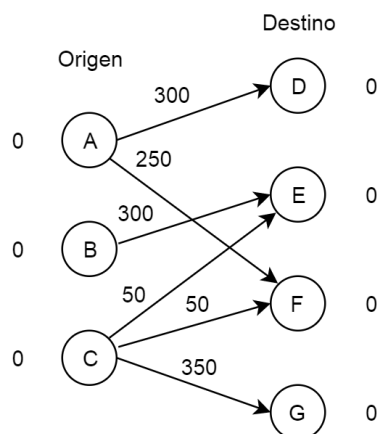
En este cromosoma, el valor de prioridad más alto (7) ocurre en el nodo destino D, lo cual significa que se debe atender primero, para lo cual se selecciona el nodo origen que tenga el menor costo para llegar a D. Observando la matriz de costos, el nodo origen A tiene el menor costo (11), por lo cual se escoge para transportar la cantidad de bienes de ayuda necesarios (300):



Se continúa con el siguiente nodo de mayor prioridad, el cual es el nodo G (prioridad=6), y según la matriz de distancias, el nodo origen de menor costo para llegar a G es el nodo C (costo=13), por lo cual se decide transportar la ayuda necesaria entre estos puntos (350):



Este proceso se repite hasta que se termine de satisfacer la demanda o hasta que ya no queden más bienes de ayuda en los almacenes:

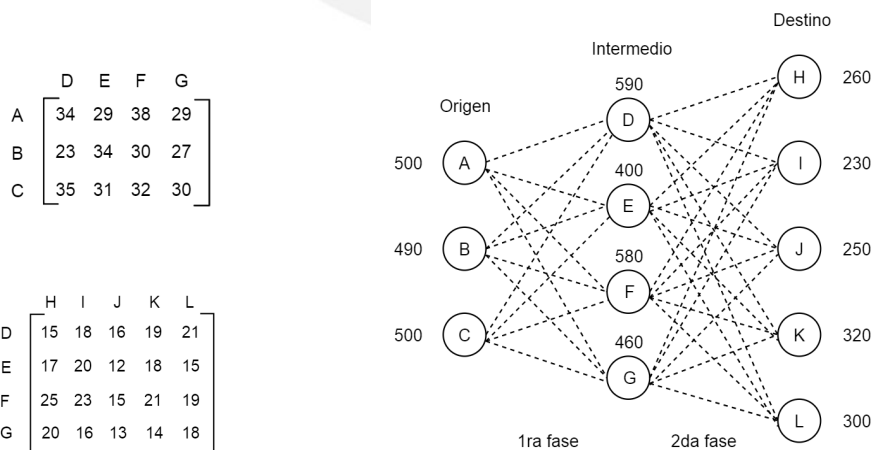


De esta manera, el cromosoma será un arreglo de tamaño N, donde N es el número de nodos en la red distribución. El contenido de cada gen del cromosoma será un número entero de 1 a N, el cual representará el nivel de prioridad de cada nodo, considerando que cada nodo tendrá un valor de prioridad diferente. En otras palabras, el cromosoma contendrá una permutación de valores de prioridad de 1 a N. Esto facilitará los operadores genéticos que se presentarán posteriormente.

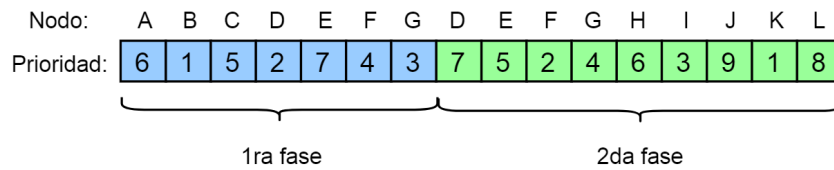
A partir del ejemplo previo, se define el siguiente algoritmo de decodificación de cromosoma:

1. **DecodificarCromosoma (cromosoma)**
2. Para cada nodo en el cromosoma hacer:
 3. *nodo* = escoger el siguiente nodo de mayor prioridad (*cromosoma*)
 4. Si *nodo* es un nodo de destino entonces:
 5. *origen* = seleccionar el nodo origen de menor costo
 6. *bienesAtransportar* = $\min(\text{oferta}_{\text{origen}}, \text{demanda}_{\text{nodo}})$
 7. ActualizarOfertaYDemanda(*origen*, *nodo*, *bienesAtransportar*)
 8. Si *nodo* es un nodo de origen entonces:
 9. *destino* = seleccionar el nodo destino de menor costo
 10. *bienesAtransportar* = $\min(\text{oferta}_{\text{nodo}}, \text{demanda}_{\text{destino}})$
 11. ActualizarOfertaYDemanda(*nodo*, *destino*, *bienesAtransportar*)

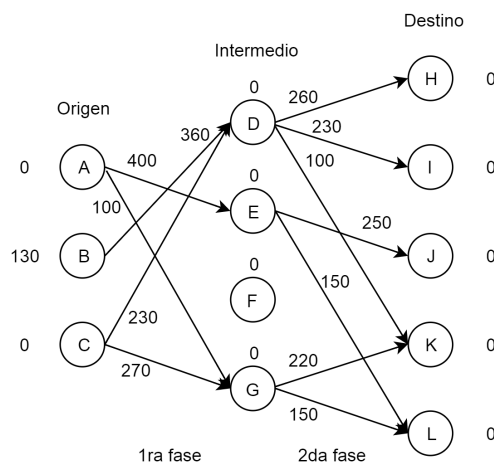
Para considerar múltiples períodos de distribución, se puede dividir la planificación en fases. Para ello se muestra el siguiente ejemplo:



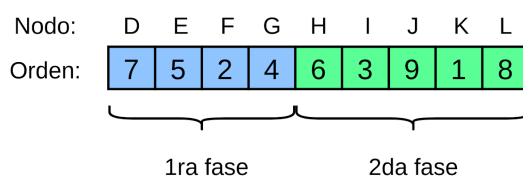
De la misma manera en que existen dos matrices para representar cada fase de distribución, el cromosoma puede ser extendido a:



Como se puede observar en la imagen, el cromosoma cuenta con una asignación de prioridades para cada fase de distribución. Dado que los nodos D, E, F y G son intermedios, aparecen en ambas fases. En este caso, el algoritmo de decodificación se aplicaría a cada fase por separado y en orden, cada una con su matriz de costos. Esta estrategia es extensible a más fases o períodos, con lo cual es posible manejar diversos tipos de redes de distribución. Para este ejemplo, a partir del cromosoma mostrado, la aplicación del algoritmo de decodificación daría como resultado la siguiente asignación:

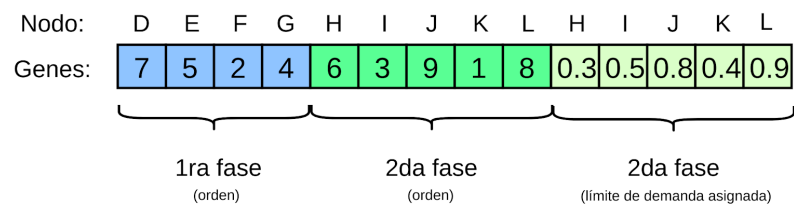


A partir de lo indicado por Costa et al. (2010) es posible realizar una representación aún más eficiente del cromosoma considerando solo los nodos destinos para el cálculo en cada fase, lo cual permite una representación más compacta y usar una sola permutación para todo el cromosoma. Para el ejemplo anterior, la nueva representación sería la siguiente:



A partir de la función objetivo correspondiente a la equidad de distribución, es posible notar que el algoritmo de decodificación no cuenta con mucha flexibilidad para realizar

entregas parciales, debido a que en el momento en que se encuentra la vía de menor costo entre dos puntos se intenta enviar todo lo posible hasta agotar la capacidad de dicha vía. Según lo indicado por Altıparmak et al. (2009), es posible agregar información adicional al cromosoma para que pueda codificar diversas variaciones de entregas parciales. Para ello, se utilizan D genes adicionales, donde D es el número de puntos de demanda. Cada uno de ellos tendrá un número real entre 0 y 1 que indica la proporción en que se limita la cantidad asignada a cada nodo de demanda. Con esta información adicional, el algoritmo de codificación asignará bienes a cada punto de demanda (en el orden dado por la permutación de prioridades) hasta el límite dado por el número real en cada pasada, hasta que se asignen la cantidad de bienes necesarios y disponibles. Para el ejemplo anterior, la representación del cromosoma sería la siguiente:



Con esta extensión, el cromosoma queda definido como un arreglo de tamaño $T + 2 * D$, donde T es el número de nodos de transbordo y D es el número de nodos de demanda. El contenido del cromosoma serán números enteros que representen el nivel de prioridad de cada nodo y números reales que representen la proporción límite de demanda asignada a los puntos de demanda:

```

Clase Cromosoma {
    Arreglo [T + 2 * D] cromosoma;
}

```

Es importante mencionar que los operadores de cruce y mutación se aplicarán por separado a los genes que contienen valores enteros (con operadores específicos para permutaciones) y a los que contienen valores reales (con operadores específicos para valores reales).

Se puede observar que el tamaño del cromosoma es aproximadamente N, donde N es el número total de nodos en la red de distribución, con lo cual se consigue tener una representación ligera y eficiente para aplicar los operadores genéticos, los cuales se describirán en el próximo capítulo.

5.3 Relación con el objetivo

Estas estructuras de datos describen cómo serán representados los parámetros y variables descritos en la formulación del problema de optimización, así como el cromosoma que será utilizado por el algoritmo genético. De esta manera, dichas estructuras servirán de entrada para los algoritmos de optimización que vienen a continuación.



Capítulo 6. Diseño del algoritmo genético para el problema de optimización de distribución de ayuda humanitaria

6.1 Introducción

Una vez definidas las estructuras de datos, se procederá a describir el algoritmo genético que será utilizado como método metaheurístico para resolver el problema de optimización de ayuda humanitaria. Para poder realizar la optimización considerando múltiples objetivos, se utilizará el criterio de dominancia, el cual se describirá posteriormente.

6.2 Esquema general del algoritmo genético

El esquema general del funcionamiento de un algoritmo genético es el siguiente:

1. **AlgoritmoGenético (N: tamaño población, G: número de generaciones, M: número máximo de generaciones sin mejora)**
2. $P = \text{ConstruirPoblaciónInicial (N)}$
3. $\text{mejor_solución} = []$
4. Repetir
5. $\text{EvaluarAptitud (P)}$
6. Para cada individuo P_i hacer
7. Si $\text{mejor_solución} = []$ o $\text{Aptitud (P}_i) > \text{Aptitud (mejor_solución)}$
8. $\text{mejor_solución} = P_i$
9. $\text{padre1, padre2} = \text{SeleccionarPadres (P)}$
10. $\text{hijo1, hijo2} = \text{Cruce (padre1, padre2)}$
11. $\text{hijo1, hijo2} = \text{Mutación (hijo1), Mutación (hijo2)}$
12. $\text{individuo1, individuo2} = \text{SeleccionarIndividuosAEliminar (P)}$
13. $P = P \cup \{\text{hijo1, hijo2}\} - \{\text{individuo1, individuo2}\}$
14. Hasta que se llegue al límite de **G** generaciones o hasta que no haya mejora en las últimas **M** generaciones
15. Retornar mejor_solución

A partir de este esquema, se detallarán las funciones utilizadas por el algoritmo.

6.3 Construcción de la población inicial

La estrategia general para crear la población inicial consiste en crear N individuos de forma aleatoria:

1. ConstruirPoblaciónInicial (N: tamaño población)

2. Población = []
3. Para i = 1 hasta N hacer
4. cromosoma = GenerarCromosoma ()
5. Población = Población U {cromosoma}
6. Retornar Población

6.3.1 Generación de cromosomas

Según el cromosoma definido en el capítulo anterior, el contenido del cromosoma es un conjunto de permutaciones por cada fase o período de distribución. Debido a ello, la estrategia para construir un cromosoma aleatorio será generar una permutación de números enteros por cada período de distribución:

1. GenerarCromosoma (T: número de períodos, R: red de distribución)

2. cromosoma = []
3. nodosOrigen = ObtenerNodosOrigen (R)
4. Para t = 1 hasta T hacer
5. nodosDestino = ObtenerNodosAdyacentes (R, nodosOrigen)
6. numeroNodos = ||nodosOrigen|| + ||nodosDestino||
7. prioridadesTransporte = GenerarPermutaciónAleatoria (1, numeroNodos)
8. cromosoma.añadirElementos(prioridadesTransporte)
9. nodosOrigen = nodosDestino // se avanza con la siguiente fase
10. Retornar cromosoma

En esta parte es importante haber asignado un número correlativo a cada nodo, como se definió en el capítulo de estructuras de datos, para mantener la consistencia en cada posición del cromosoma.

6.4 Evaluación de la aptitud

La aptitud de cada cromosoma será medida por las dos expresiones objetivo comentadas en la formulación del problema: minimización costo de transporte (F1) y minimización de inequidad en la distribución (F2):

$$\sum_{j,k,i} CostoTransp_{j,k} * Transp_{j,k,i}$$

$$\sum_{k,l} \left(\frac{\sum_j \text{Transp}_{j,k}}{\text{Demanda}_k} \right)^2$$

Luego de haberse decodificado el cromosoma según lo descrito en el capítulo anterior, se obtiene información de cuántos bienes de ayuda serán transportados en cada tramo, lo cual permite obtener los valores de F1 y F2. Ambos valores representan la aptitud del cromosoma, por lo cual se representan como una tupla (F1, F2). En este caso los objetivos pueden estar enfrentados, pues para tener un mínimo costo de transporte solo se distribuirían los bienes disponibles a los puntos de menor costo (puntos más cercanos, por ejemplo) lo cual originaría inequidad en la distribución. De similar manera, para obtener la máxima equidad en la distribución, puede ser necesario utilizar rutas de mayor costo (para alcanzar a puntos más alejados, por ejemplo). Ante esto, existen variadas maneras de definir el conjunto de “mejores opciones”, siendo la predominante el **frente de Pareto** del espacio de las soluciones candidato.

6.4.1 Frente de Pareto

Sea un par de posibles soluciones A y B. Se dice que A **domina** a B si A es tan buena como B en todos los objetivos y además es superior a B en por lo menos un objetivo. Esto hace que A sea una solución preferible a B indiscutiblemente.

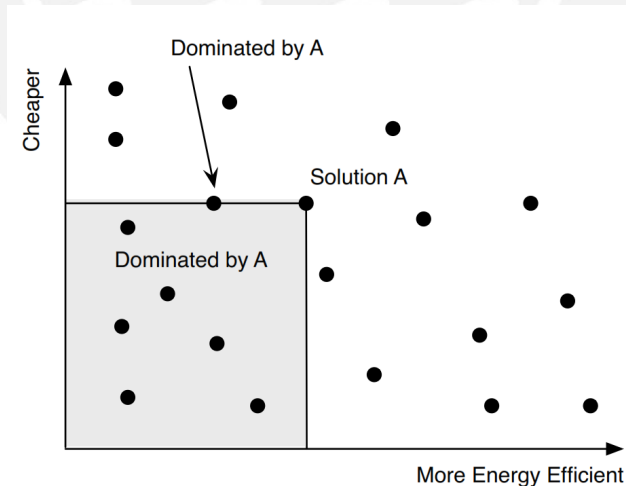


Ilustración 7. Región de soluciones dominadas por la solución A en un espacio de 2 objetivos (Luke, 2015)

A partir de esta definición, se puede establecer el conjunto de “mejores opciones” como el conjunto de soluciones que no son dominadas por ninguna otra. Esto conforma lo que se conoce como **frente de Pareto**:

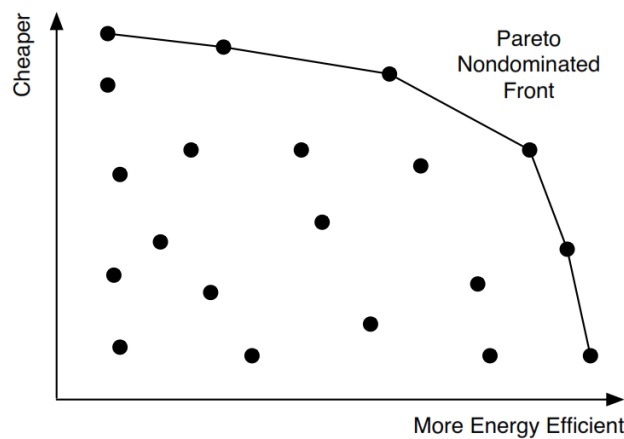


Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015).

Cada una de las soluciones del frente de Pareto representan un compromiso entre uno y otro objetivo, de tal manera que se presentan como mejores alternativas a escoger por el encargado de tomar las decisiones. A partir de esta definición, ya se cuenta con el criterio de dominancia para comparar soluciones candidatas. Esta comparación se realiza a detalle en el operador de selección, el cual viene a continuación.

6.5 Selección

El objetivo del operador de selección es escoger de forma probabilística a individuos con un alto nivel de aptitud para usarlos en los siguientes operadores, principalmente en el operador de cruce. El concepto de dominancia nos permite dividir a la población de posibles soluciones en grupos: el primer grupo corresponde a las soluciones no dominadas, a las cuales se asigna el **rango** de 1. Para encontrar a las soluciones del siguiente rango, basta con remover temporalmente a las soluciones de rango 1, lo cual origina un nuevo conjunto de soluciones no dominadas, a las cuales se asigna el rango de 2. Este proceso se repite sucesivamente, de la siguiente manera:

1. **AsignarRango (P: población)**
2. $P' = P$
3. Rango = [] // rango de cada individuo
4. $i = 1$
5. Repetir
6. frentePareto = []
7. Para cada individuo p de P' hacer
8. esDominado = EvaluarDominancia (P' , p)

9. Si no es Dominado entonces
10. Rango[p] = i
11. frentePareto.añadirElemento (p)
12. $P' = P' - \text{frentePareto}$ // remover los individuos de rango i para encontrar
13. los siguientes
14. $i = i + 1$ // para encontrar a los individuos del siguiente rango
15. Hasta que P' esté vacío
16. Retornar Rango

Con el valor de rango se tiene el primer criterio para ordenar a los individuos. Ahora, también sería preferible que las mejores soluciones estén distribuidas de forma homogénea a lo largo del frente de Pareto, de lo contrario, terminaríamos con soluciones no dominadas pero muy parecidas entre sí. Para ello, se define el concepto de **dispersión**: un individuo está en un espacio más disperso si sus vecinos están más alejados de él. En el caso de dos objetivos, esta distancia se puede medir mediante la distancia euclidiana. De esta manera, el valor de dispersión de un individuo se puede definir como la suma de las distancias a sus dos vecinos más cercanos.

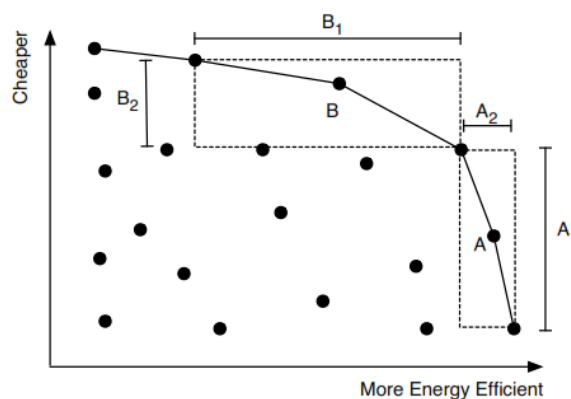


Ilustración 9. Dispersión de las soluciones A y B. B está más alejado de sus vecinos, por lo cual tendría un valor más alto de dispersión (Luke, 2015).

Este sería el segundo criterio para evaluar un individuo. En resumen, un individuo es más apto que otro si tiene un **rango** de dominancia más alto que otro y en caso de empate, el individuo más apto es el de mayor valor de **dispersión**.

6.5.1 Selección por torneo

La selección por torneo es un método de selección en algoritmos genéticos que se caracteriza por su invarianza a la escala del valor de la aptitud de los individuos. Esto la hace ideal para diferentes tipos de objetivos y en especial cuando el algoritmo genético

se encuentra cerca de la convergencia, donde los valores de aptitud serían cercanos entre sí y no se distinguirían mucho (Luke, 2015).

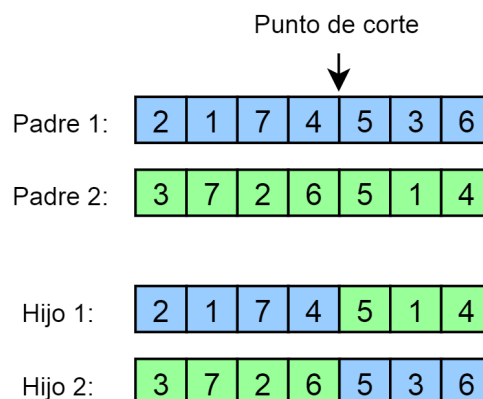
El algoritmo de selección por torneo es el siguiente:

1. **SelecciónPorTorneo (P: población, t: tamaño del torneo)**
2. $P' = P$
3. mejorIndividuo = EscogerIndividuoAlAzar (P)
4. Para $i = 2$ hasta t hacer:
 5. individuo = EscogerIndividuoAlAzar (P)
 6. Si $Aptitud(individuo) > Aptitud(mejorIndividuo)$ entonces:
 7. mejorIndividuo = individuo
8. Retornar mejorIndividuo

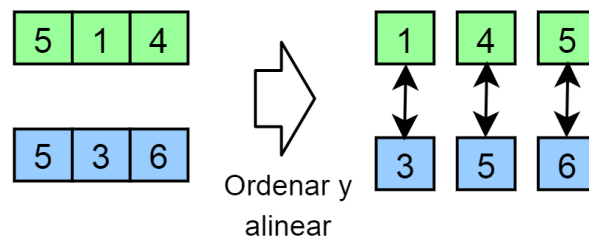
Para evaluar la aptitud se utilizará el concepto de aptitud por dominancia definido en la sección anterior.

6.6 Cruce

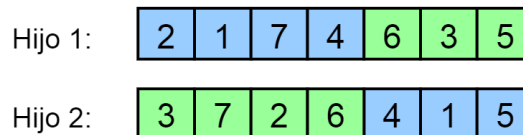
El operador de cruce permite juntar la información de dos individuos, preferiblemente los más aptos, con el fin de crear individuos que contengan las mejores características de ambos padres. Para la primera parte del cromosoma que representa permutaciones, es posible aplicar operadores de cruce específicos para permutaciones. La estrategia para realizar el cruce de dos permutaciones es escoger aleatoriamente un punto de corte e intercambiar secciones entre los cromosomas para generar un par de descendientes. Ahora, es posible que al realizar el intercambio se pierda la unicidad de los valores que conforman la permutación, como el siguiente ejemplo:



Se puede observar que el primer hijo repite los valores 1 y 4, lo cual lo hace un cromosoma inválido. Similar situación sucede con el segundo hijo, el cual repite los valores 3 y 6. Para solucionar esta situación es posible aplicar una reasignación entre los valores que fueron intercambiados, para ello, primero se ordenan las secciones que fueron intercambiadas para luego alinearlas, con lo cual se obtiene el mapeo:



Con aplicando dicho mapeo, los cromosomas hijos vuelven a conservar la unicidad de sus valores:



Esto queda reflejado en el siguiente algoritmo de cruce:

1. **Cruce (p1: primer cromosoma padre, p2: segundo cromosoma padre)**
2. $n = \text{longitudDeCromosoma}(p1)$
3. $\text{puntoDeCorte} = \text{EscogerNúmeroAleatorioEntre}(1, n)$
4. $\text{hijo1} = p2 [\text{puntoDeCorte} + 1 : n]$
5. $\text{hijo2} = p1 [\text{puntoDeCorte} + 1 : n]$
6. $\text{mapa1}, \text{mapa2} = \text{AlinearYCrearMapeo} (\text{Ordenar}(\text{hijo1}), \text{Ordenar}(\text{hijo2}))$
7. $\text{hijo1} = \text{AplicarMapeo}(\text{hijo1})$
8. $\text{hijo2} = \text{AplicarMapeo}(\text{hijo2})$
9. $\text{hijo1} = \text{Concatenar} (p1 [1 : \text{puntoDeCorte}], \text{hijo1})$
10. $\text{hijo2} = \text{Concatenar} (p2 [1 : \text{puntoDeCorte}], \text{hijo2})$
11. Retornar hijo1, hijo2

Para la segunda parte del cromosoma, que representa valores reales, se utilizará el cruce aritmético completo, que consiste en combinar los valores de los genes de dos

padres mediante una combinación lineal de ellos, lo cual queda representado en las siguientes expresiones:

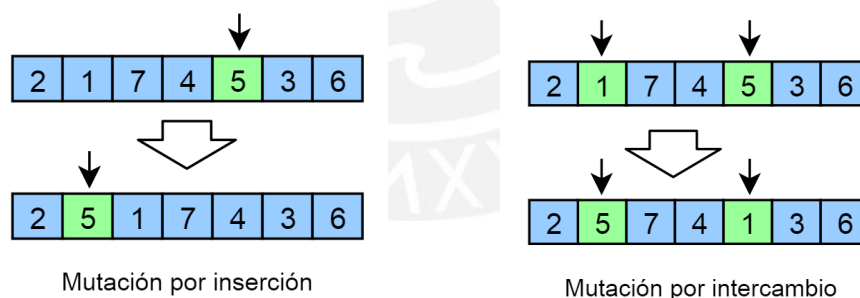
$$h_1 = \alpha * p_1 + (1 - \alpha) * p_2$$

$$h_2 = \alpha * p_2 + (1 - \alpha) * p_1$$

Donde h_1 y h_2 son los valores de los hijos, p_1 y p_2 son los valores de los padres, y α es un número real entre 0 y 1 escogido aleatoriamente que determina la influencia de cada padre en el valor resultante del hijo.

6.7 Mutación

El operador de mutación permite realizar cambios aleatorios a los individuos de la población, lo cual permite explorar nuevos espacios de búsqueda para encontrar el valor óptimo. Para la primera parte del cromosoma que representa una permutación, existen dos maneras principales para generar un cambio de manera aleatoria. La primera estrategia consiste en seleccionar un gen del cromosoma de forma aleatoria y moverlo a otra posición, también determinada de forma aleatoria. Esta estrategia es conocida como mutación por inserción. La segunda estrategia consiste en seleccionar un par de genes del cromosoma de forma aleatoria y luego intercambiar sus posiciones. Esta estrategia es conocida como mutación por intercambio.



De manera similar al cruce, para considerar múltiples períodos de distribución, el cromosoma se segmenta por cada período o fase según lo descrito en el capítulo anterior, y la mutación se realiza por cada período separado, debido a que cada fase representa una permutación independiente.

Para la segunda parte del cromosoma, que representa valores reales, se utilizará la mutación de punto flotante que consiste en sumar un delta aleatorio al valor de cada gen, lo cual queda representado por la siguiente expresión:

$$x' = x + \alpha * \Delta_{max}$$

Donde x' es el valor nuevo del gen, x es el valor original del gen, α es un número aleatorio entre -1 y 1 que determina la intensidad de la variación y δ es la máxima variación permitida.

6.8 Criterio de parada

Existirán dos criterios de parada. El primero será el número de generaciones del algoritmo genético, mientras que el segundo será un número máximo permitido de generaciones sin mejora. Si alguna de las condiciones se cumple, esto es, si se llevan a cabo todas las generaciones del algoritmo genético o si no se perciben mejoras en un número dado de generaciones, se detendrá la ejecución del algoritmo y se retornarán las mejores soluciones.



Capítulo 7. Diseño del modelo matemático de programación no lineal para el problema de optimización de distribución de ayuda humanitaria

7.1 Introducción

A partir de la definición del problema de optimización, se procederá a diseñar un modelo matemático que será usado para resolver el problema de optimización de forma exacta, el cual será comparado con el método metaheurístico para evaluar las diferencias y ventajas de cada método. Con el método exacto es posible tener una referencia clara de la efectividad del algoritmo genético y una línea base del tiempo de ejecución que evidencie la ventaja de usar un método aproximado para problemas de gran tamaño.

7.2 Formulación del modelo matemático

El modelo matemático queda definido por conjuntos (*sets*), parámetros, variables de decisión, funciones objetivo, restricciones y rango de existencia. La descripción de cada función objetivo y restricción se encuentra en el capítulo 4.

7.2.1 Conjuntos

S: puntos de abastecimiento

T: puntos de transbordo

D: puntos de demanda

K: nodos en la red de distribución ($K = S \cup T \cup D$)

I: tipos de bienes a distribuir

7.2.2 Parámetros

CostoTransp_{j,k}: costo de transporte entre *j* y *k* ($j \in K, k \in K$)

Inventario_{k,i}: inventario total de *i* en *k* ($i \in I, k \in S$)

Demanda_{k,i}: demanda total de *i* en *k* ($i \in I, k \in D$)

Capacidad_k: capacidad total en *k* ($k \in K$)

7.2.3 Variables de decisión

Transp_{j,k,i}: bienes *i* transportados entre *j* y *k* ($j \in K, k \in K, i \in I$)

7.2.4 Funciones objetivo

$$\min f1: \sum_{j \in K} \sum_{k \in K} \left(\text{CostoTransp}_{j,k} * \sum_{i \in I} \text{Transp}_{j,k,i} \right)$$

$$\min f2: \sum_{k \in D} \sum_{i \in I} \left(\frac{\sum_{j \in K} \text{Transp}_{j,k}}{\text{Demanda}_k} \right)^2$$

La primera función objetivo calcula la suma del costo de transporte en cada vía usada en la red de distribución por cada tipo de bien transportado, mientras que la segunda función objetivo estima la media cuadrática de los porcentajes de satisfacción¹. El valor mínimo de la segunda función objetivo se alcanza cuando todos los porcentajes de satisfacción son iguales, en otras palabras cuando se realiza una distribución perfectamente equitativa.

7.2.5 Restricciones

$$\sum_{k \in K} \text{Transp}_{j,k,i} = \text{Inventario}_{j,i} \quad , \forall j \in S, i \in I \quad \dots \dots \dots (1)$$

$$\sum_{j \in K} \text{Transp}_{j,k,i} = \text{Demanda}_{k,i} \quad , \forall k \in D, i \in I \quad \dots \dots \dots (2)$$

$$\sum_{j \in K} \text{Transp}_{j,k,i} = \sum_{m \in K} \text{Transp}_{k,m,i} \quad , \forall k \in T, i \in I \quad \dots \dots (3)$$

$$\sum_{j \in K} \sum_{i \in I} \text{Transp}_{j,k,i} \leq \text{Capacidad}_k \quad , \forall k \in K \quad \dots \dots \dots (4)$$

$$\sum_{m \in K} \sum_{i \in I} \text{Transp}_{k,m,i} \leq \text{Capacidad}_k \quad , \forall k \in K \quad \dots \dots \dots (5)$$

La restricción (1) asegura que solo se puedan transportar bienes que se encuentren en el inventario, para los nodos que representan almacenes. La restricción (2) asegura que los bienes que sean transportados a los puntos de demanda sean equivalentes a la demanda total existente en dichos nodos. En casos cuando la demanda total es mayor o menor a la cantidad total de bienes disponibles, se pueden introducir previamente nodos ficticios que cubran el excedente de demanda o de bienes disponibles (Hillier &

¹ Como se podrá notar, no se ha aplicado la división ni la raíz cuadrada a la media cuadrática, debido a que ambas operaciones son monótonas (conservan el orden de los resultados) y por tanto no afectan a los métodos de optimización.

Lieberman, 2015), con lo cual se mantiene el balance entre oferta y demanda. La restricción (3) asegura el balance del flujo de entrada y salida en los nodos intermedios (puntos de transbordo). Finalmente, la restricciones (4) y (5) establecen los límites de capacidad de entrada y de salida, respectivamente.

7.2.6 Rango de existencia

$$Transp_{j,k,i} \in Z_{\geq 0}$$

$$CostoTransp_{j,k} \in R_{\geq 0}$$

$$Inventario_{k,i} \in Z_{\geq 0}$$

$$Demanda_{k,i} \in Z_{\geq 0}$$

$$Capacidad_k \in Z_{\geq 0}$$

Todos los parámetros y variables con enteros positivos a excepción del costo de transporte que puede tomar valores reales.

7.3 Algoritmo de optimización exacto (*solver*)

El algoritmo (*solver*) que se utilizará para resolver el programa no lineal de forma exacta es Couenne. Couenne (*Convex Over and Under ENvelopes for Nonlinear Estimation*) es un algoritmo de tipo ramificación y poda (*branch & bound*) para resolver problemas de programación no lineal entera mixta (MINLP) de forma exacta (Couenne, 2006). Para utilizarlo, se debe implementar la formulación matemática en un lenguaje de modelado como AMPL (*A Mathematical Programming Language*). El lenguaje AMPL mantiene una sintaxis lo más similar posible a un modelo matemático, incluyendo operadores como sumatorias y cuantificadores lógicos que permiten expresar los problemas de forma compacta. El lenguaje AMPL permite representar 500 variables de decisión y 500 restricciones como máximo para problemas lineales, y 300 variables y 300 restricciones como máximo para problemas no lineales, en su versión gratuita (AMPL, 2018). La versión de pago no tiene límites de cantidades de variables y restricciones. Un ejemplo del lenguaje AMPL es el que se muestra a continuación:

```

set I;
set ST;
set D;
set K := ST union D;
set E within {K, K};

set DU;
set KDU := K union DU;
set EDU within {KDU, KDU};

param transp_cost {E};
param supply_demand {KDU, I};
param node_capacity {K};

var X {EDU, I} >= 0 integer;

minimize Objectives:
    (sum {(k,j) in E, i in I} transp_cost[k,j] * X[k,j,i]) +
    (sum {k in D, i in I} ((sum {(j,k) in E} X[j,k,i]) / -supply_demand[k,i]) ^ 2);

subject to Transportation_Balance {k in KDU, i in I}:
    sum {(k,j) in EDU} X[k,j,i] - sum {(j,k) in EDU} X[j,k,i] = supply_demand[k,i];

subject to Inbound_Capacity {k in K}:
    sum {(j,k) in E, i in I} X[j,k,i] <= node_capacity[k];

subject to Outbound_Capacity {k in K}:
    sum {(k,j) in E, i in I} X[k,j,i] <= node_capacity[k];

```

Ilustración 10: Lenguaje de modelado AMPL

Para ejecutar el algoritmo con una instancia del problema, los parámetros se deben especificar en un formato similar, como el que se muestra a continuación:

```

data;

set I := I1 I2 I3 I4 I5;
set ST := S1 T11;
set D := D1 D2 D3;
set DU := dummy_sup dummy_dem;

set E :=
    (S1,T11)
    (T11,D1) (T11,D2) (T11,D3)
;

set EDU :=
    (S1,T11)
    (T11,D1) (T11,D2) (T11,D3)
    (S1,dummy_dem)
    (dummy_sup,D1) (dummy_sup,D2) (dummy_sup,D3)
;

param transp_cost := S1 T11 802 T11 D1 501 T11 D2 829 T11 D3 655;

param supply_demand (tr): S1 T11 D1 D2 D3 dummy_sup dummy_dem :=
    I1 400 0 -120 -130 -250 100 0
    I2 400 0 -170 -130 -200 100 0
    I3 400 0 -120 -170 -210 100 0
    I4 400 0 -130 -180 -190 100 0
    I5 400 0 -130 -140 -230 100 0
;

param node_capacity := S1 2000 T11 2754 D1 670 D2 750 D3 1080;

```

Ilustración 11: Ingreso de datos en lenguaje AMPL

Finalmente, una vez especificado problema, se puede ejecutar el algoritmo de optimización Couenne desde la línea de comandos, como se muestra a continuación:

```

Couenne 0.5.6 -- an Open-Source solver for Mixed Integer Nonlinear Optimization
Mailing list: couenne@list.coin-or.org
Instructions: http://www.coin-or.org/Couenne

NLP0012I
      Num      Status      Obj      It      time      Location
NLP0014I      1      OPT 699857.11      20 0.034923
Couenne: new cutoff value 6.9986622058e+05 (0.059798 seconds)
NLP0014I      2      INFEAS 699997.89      0 0
Loaded instance "datasets/1013i.nl"
Constraints:      46
Variables:      78 (78 integer)
Auxiliaries:      56 (25 integer)

Coin0506I Presolve 135 (-17) rows, 109 (-25) columns and 553 (-55) elements
Clp0006I 0 Obj 0 Primal inf 3679.3383 (24)
Clp0006I 46 Obj 583280 Primal inf 64.451847 (24)
Clp0006I 73 Obj 690532.92
Clp0000I Optimal - objective value 690532.92
Clp0032I Optimal objective 690532.9167 - 73 iterations time 0.002, Presolve 0.00
Clp0006I 0 Obj 690532.92 Primal inf 1948.8729 (2)
Clp0006I 2 Obj 690532.92
Clp0000I Optimal - objective value 690532.92
NLP Heuristic: time limit reached.
Clp0000I Optimal - objective value 690532.92
Optimality Based BT: 26 improved bounds
Probing: 4 improved bounds
NLP Heuristic: time limit reached.
Cbc0031I 16 added rows had average density of 2
Cbc0013I At root node, 16 cuts changed objective from 690532.92 to 699857.11 in 11 passes
Cbc0014I Cut generator 0 (Couenne convexifier cuts) - 62 row cuts average 2.0 elements, 6 column cuts (6 active)
Cbc0010I After 0 nodes, 1 on tree, 1e+50 best solution, best possible -1.7976931e+308 (0.24 seconds)
Optimality Based BT: 6 improved bounds
Couenne: new cutoff value 6.9985880095e+05 (0.400187 seconds)
Optimality Based BT: 5 improved bounds
Optimality Based BT: 5 improved bounds
Couenne: new cutoff value 6.9985749617e+05 (0.673922 seconds)
Cbc0004I Integer solution of 699857.49 found after 100 iterations and 21 nodes (0.69 seconds)
Cbc0010I After 100 nodes, 8 on tree, 699857.49 best solution, best possible 699857.29 (0.80 seconds)
Cbc0010I After 200 nodes, 5 on tree, 699857.49 best solution, best possible 699857.29 (0.97 seconds)
Cbc0010I After 300 nodes, 2 on tree, 699857.49 best solution, best possible 699857.29 (1.13 seconds)
Cbc0001I Search completed - best objective 699857.4897937207, took 703 iterations and 366 nodes (1.23 seconds)
Cbc0035I Maximum depth 18, 0 variables fixed on reduced cost

couenne: Optimal

      "Finished"

Linearization cuts added at root node:      152
Linearization cuts added in total:      152 (separation time: 0.000983s)
Total solve time:      1.23846s (1.23846s in branch-and-bound)
Lower bound:      699857
Upper bound:      699857 (gap: 0.00%)
Branch-and-bound nodes:      366
Performance of      FBBT:      0.060361s,      319 runs. fix: 0.0047679 shrnk: 0.70
3747 ubd: 1.75777 2ubd:      0 infeas:      1

```

Ilustración 12: Ejecución y reporte de resultados de Couenne

A partir de este reporte de resultados, se puede obtener información sobre el valor óptimo calculado y el tiempo de ejecución del algoritmo, los cuales serán usados en la sección de experimentación numérica para ser comparados con los resultados del algoritmo genético.

Capítulo 8. Experimentación Numérica

8.1 Introducción

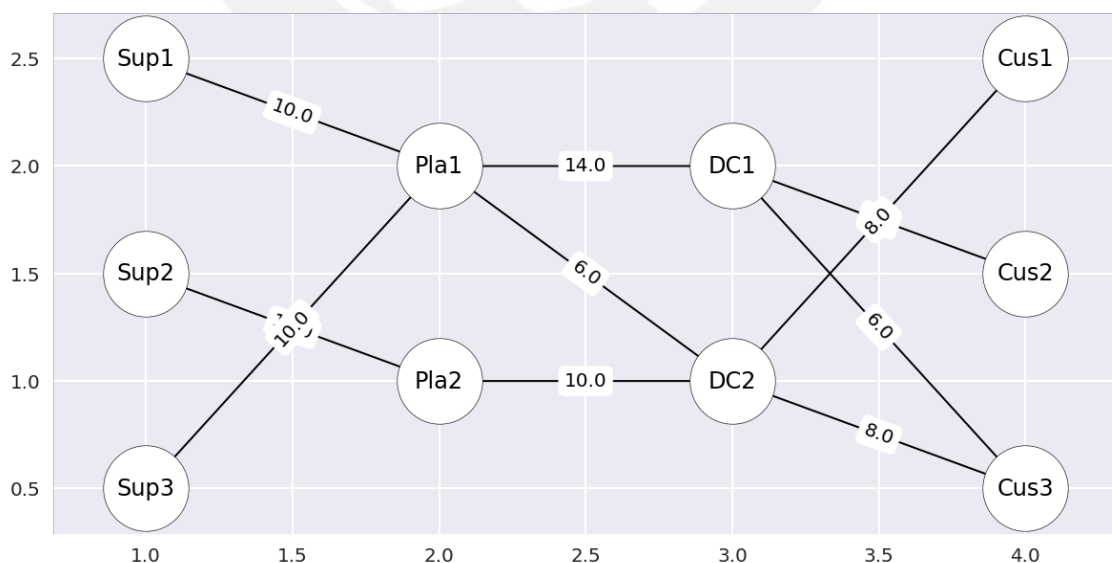
Una vez obtenidos los resultados del algoritmo genético y la optimización del programa no lineal entero mixto mediante el algoritmo Couenne (*solver*), es momento de comparar dichos resultados para validar las hipótesis de investigación del presente proyecto de fin de carrera.

8.2 Configuración de la experimentación

Se ejecutará cada método con 32 conjuntos de datos diferentes, resultado de combinar 8 configuraciones diferentes de tamaños de redes de distribución con 4 cantidades diferentes de tipos de bienes a distribuir. Las 8 configuraciones de tamaños de redes de distribución son 5, 10, 15, 20, 30, 50, 70 y 100 nodos de distribución, mientras que las 4 configuraciones de tipos de bienes a distribuir son 1, 3, 5 y 7 tipos diferentes de bienes a distribuir.

8.2.1 Generación de conjuntos de datos aleatorios

Una vez fijados los valores de cantidad de nodos a distribuir y cantidad de tipos de bienes a distribuir, se distribuyen los nodos en 3, 4 o 5 niveles según la cantidad de nodos. Este es un ejemplo para una red de distribución de 10 nodos dividida en 4 niveles:



El primer nivel se asigna para los almacenes, el último nivel para los puntos de demanda, y los niveles intermedios para los puntos de transbordo. Una vez determinados los

puntos de abastecimiento y demanda, se asigna una cantidad aleatoria de bienes disponibles en los almacenes y bienes requeridos en los puntos de demanda tomando muestras de la distribución de Poisson, la cual es una distribución discreta comúnmente usada para simular cantidades discretas (Montgomery, 2011).

Después de asignar las cantidades de oferta y demanda de bienes, se simulan los costos de transporte entre los nodos mediante una distribución uniforme, dado, que el costo de transporte es un valor continuo. Para los conjuntos de prueba generados se ha utilizado una distribución de Poisson con media de 400 y una distribución uniforme con rango de 100 a 1000, para otorgar suficiente variabilidad a los parámetros generados aleatoriamente.

8.2.2 Recolección de datos

Una vez generados los conjuntos de prueba, se ejecutaron con el método exacto (Couenne) y con el algoritmo genético configurado con unos criterios de parada de 200 generaciones y cantidad de generaciones máxima sin cambios igual a 20, con el fin de que el algoritmo se detenga si no detecta mejoras en las últimas 20 generaciones. Para que los valores de las funciones objetivo sean comparables, dado que el método exacto realiza la optimización sobre un valor único, se han juntado las dos funciones objetivo originales sumándolas en un solo valor. De esta manera, la función objetivo a optimizar es la siguiente:

$$\min f1 + \alpha f2: \sum_{j \in K} \sum_{k \in K} \left(CostoTransp_{j,k} * \sum_{i \in I} Transp_{j,k,i} \right) + \alpha \sum_{k \in D} \sum_{i \in I} \left(\frac{\sum_{j \in K} Transp_{j,k}}{Demanda_k} \right)^2$$

El valor de alpha se ha incluido debido a que f1 y f2 se encuentran en escalas diferentes, debido a que f1 representa una suma de costos y f2 un promedio de porcentajes de satisfacción. Para que ambos valores sean comparables, realizando pruebas se encontró un valor de alpha de 10,000. Finalmente, dado que el algoritmo genético es no determinístico, se han realizado 10 corridas por cada conjunto de pruebas, y se ha considerado el promedio de los valores objetivo y tiempos de ejecución en la comparación.

8.3 Resultados de la experimentación

Los resultados se han agrupado en el análisis del tiempo de ejecución y el análisis del valor objetivo óptimo alcanzado por cada método.

8.3.1 Tiempo de ejecución

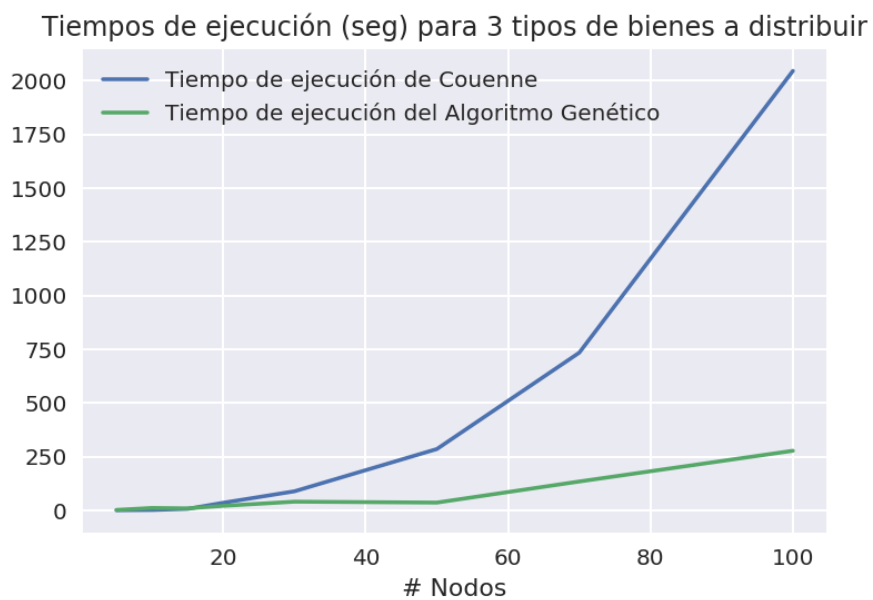
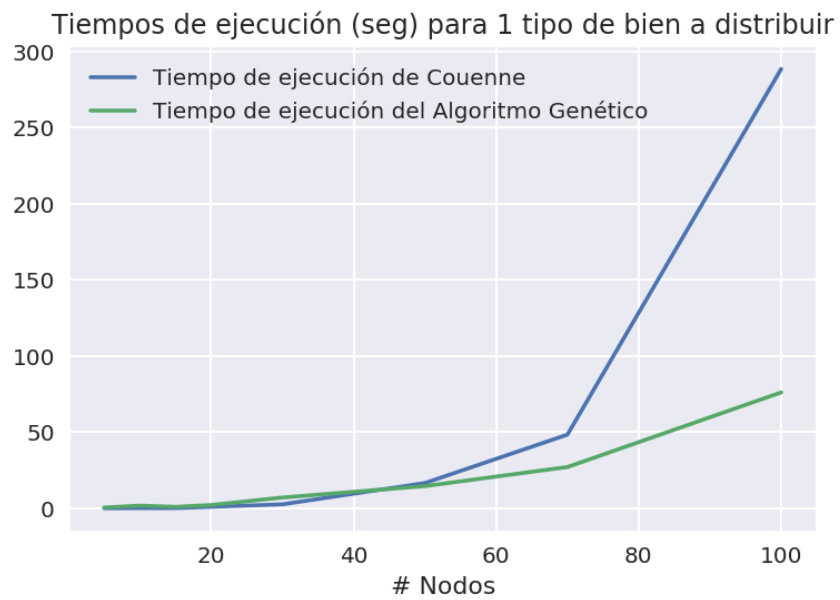
Los resultados de tiempos de ejecución expresados en segundos se muestran en la siguiente tabla, agrupadas por la cantidad de tipos de bienes a distribuir:

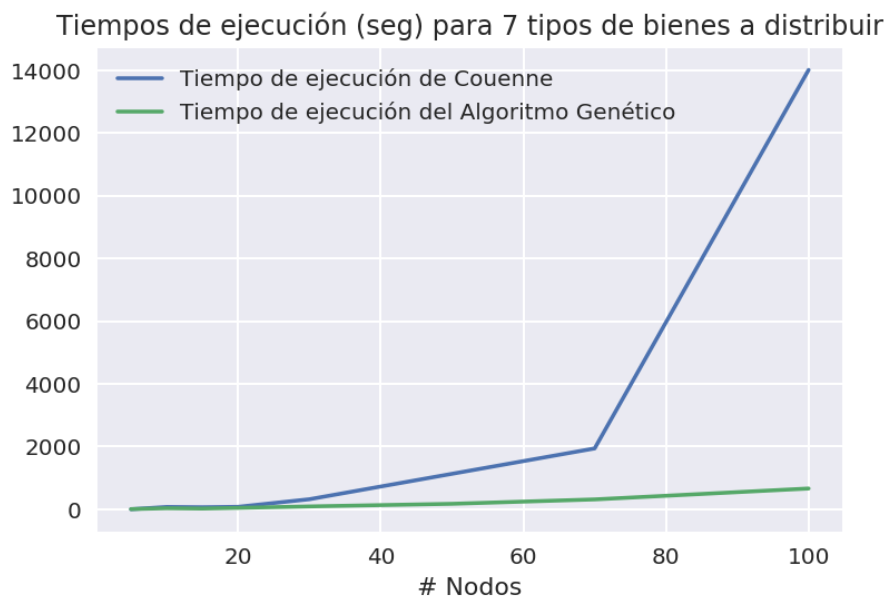
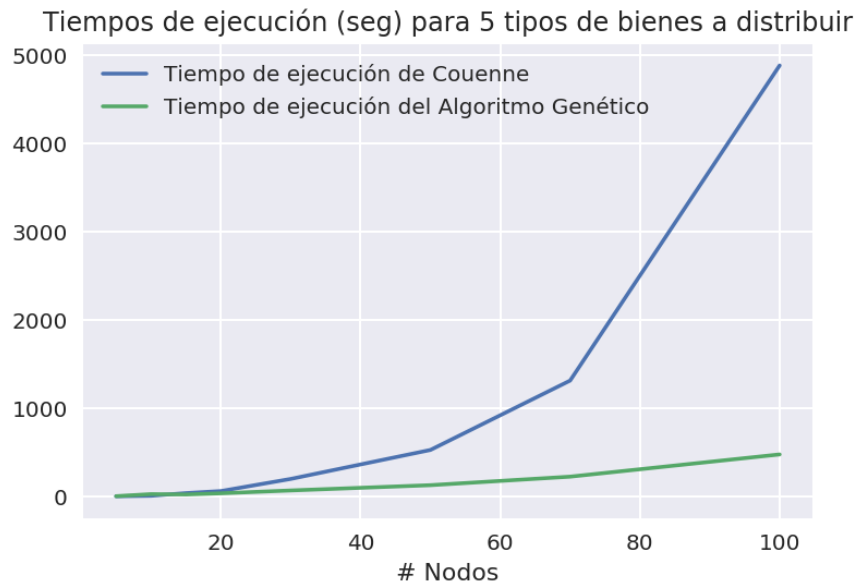
# Nodos	# Bienes	Tiempo de ejecución de Couenne (segundos)	Tiempo de ejecución del Algoritmo Genético (segundos)
5	1	0.01	0.5
10	1	0.09	1.72
15	1	0.09	0.93
20	1	0.95	2.13
30	1	2.57	7.05
50	1	16.57	14.55
70	1	48.37	27.08
100	1	288.7	76.13
5	3	0.12	1.64
10	3	1.24	10.7
15	3	6.48	8.78
20	3	34.89	20.58
30	3	88.29	39.99
50	3	285	35.72
70	3	733.61	133.92
100	3	2045.12	276.96
5	5	0.01	1.31
10	5	5.69	24.61
15	5	36.03	20.43
20	5	59.12	33.53
30	5	196.91	66.15
50	5	525.62	126.08
70	5	1312.68	222.72
100	5	4887.99	474.88
5	7	0.01	4.76
10	7	73.32	32.12
15	7	64.24	21.57
20	7	76.82	43.26

30	7	318.64	89.45
50	7	1130.16	173.11
70	7	1939.96	313.7
100	7	14018.52	661.44

Tabla 3: Resultados del tiempo de ejecución de los algoritmos

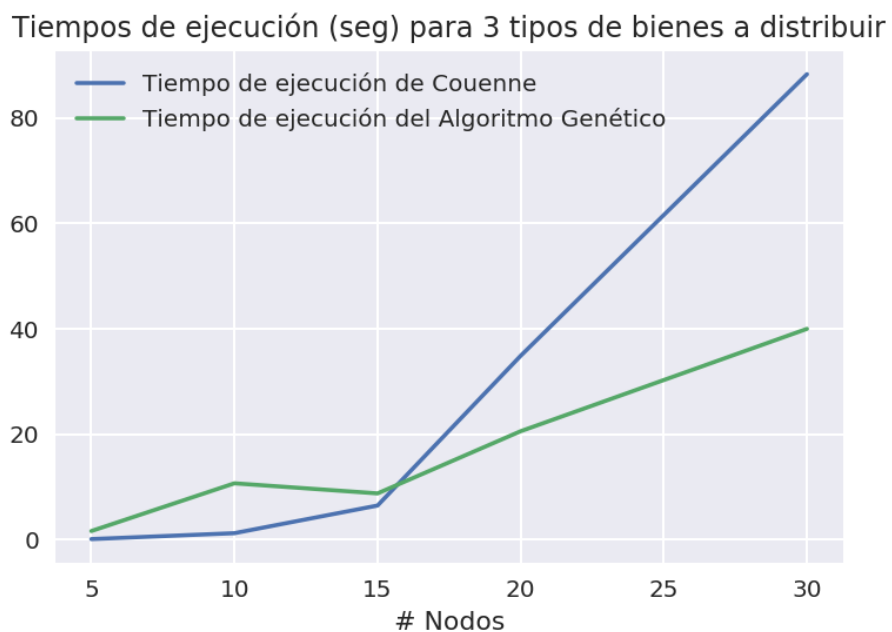
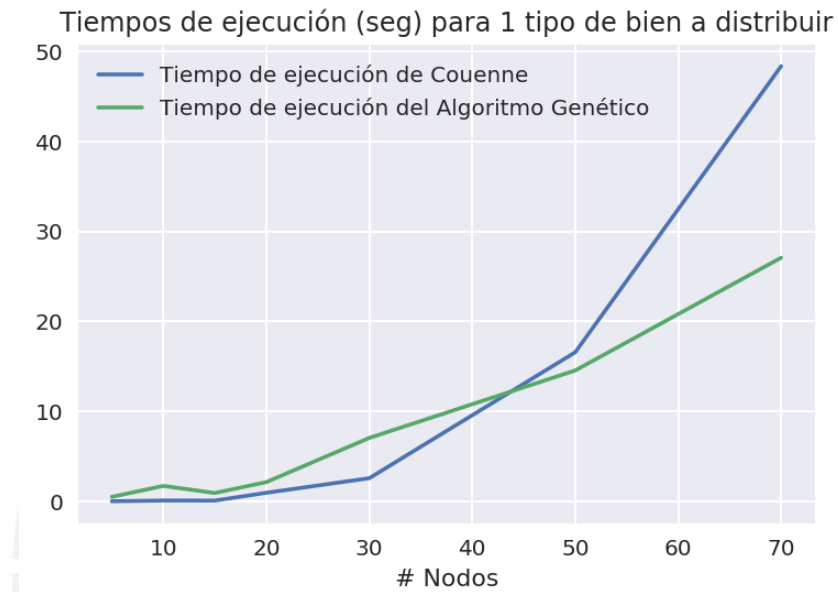
Para visualizar mejor los resultados se presentan los siguientes gráficos:



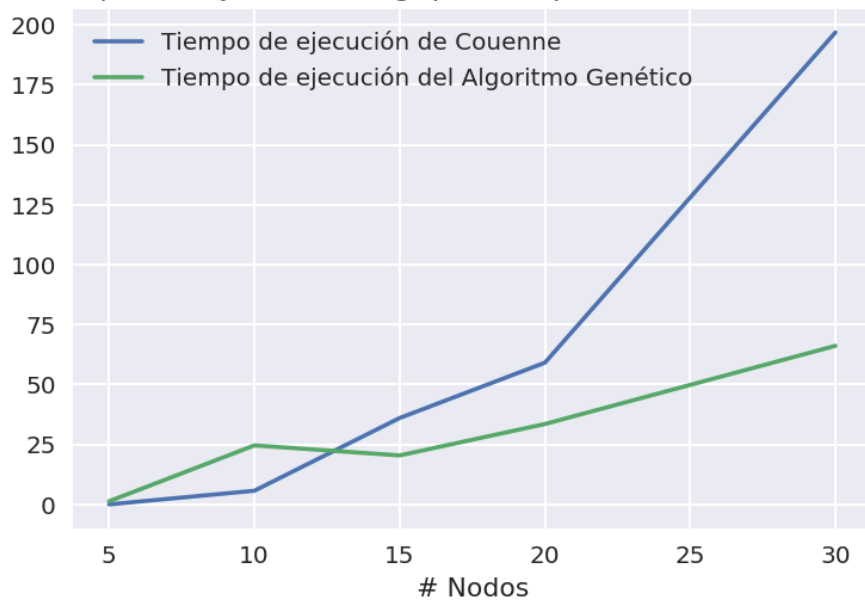


En estos gráficos, se puede apreciar que para problemas de tamaño pequeño (con pocos nodos en la red de distribución) el método exacto (Couenne) es igual de rápido o más rápido que el algoritmo genético. Sin embargo, a medida que la cantidad de nodos aumenta, el tiempo de ejecución del método exacto crece a una velocidad tal que, a partir de determinado número de nodos, supera al tiempo de ejecución del algoritmo genético. Al dividir los gráficos por número de tipos de bienes a distribuir, se puede apreciar el impacto de la cantidad de bienes diferentes en el tiempo de ejecución, donde para un mayor número de bienes el método exacto se vuelve más lento que el algoritmo genético con mayor prontitud.

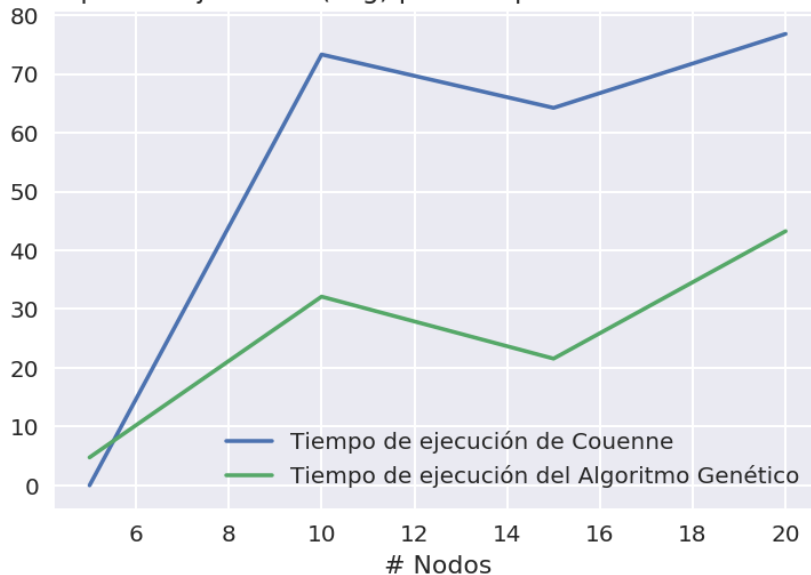
Esta información nos es de utilidad para determinar a partir de qué punto conviene cambiar de método de optimización, considerando su uso en un sistema de apoyo a la toma de decisiones. Para visualizarlo mejor, se presentan los siguientes gráficos que hacen un acercamiento a los puntos de corte:



Tiempos de ejecución (seg) para 5 tipos de bienes a distribuir



Tiempos de ejecución (seg) para 7 tipos de bienes a distribuir



En estos gráficos se puede apreciar que, para 1 tipo de bien a distribuir, a partir de aproximadamente 45 nodos en la red de distribución, el método exacto se vuelve más lento que el algoritmo genético, mientras que, para 5 tipos de bienes a distribuir, el punto de corte ocurre solo con aproximadamente 13 nodos en la red de distribución. Esto evidencia el efecto de la cantidad de tipos de bienes en el tiempo de ejecución, lo cual se debe a que, al aumentar el número de tipos de bienes a distribuir, se multiplica la cantidad de variables de decisión por el mismo factor.

8.3.2 Valor objetivo

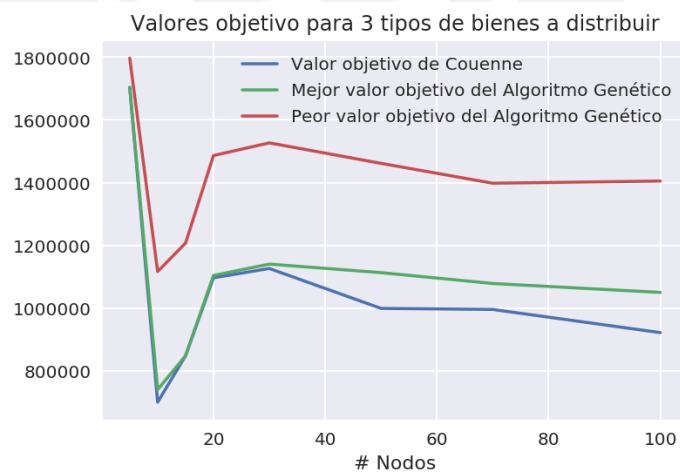
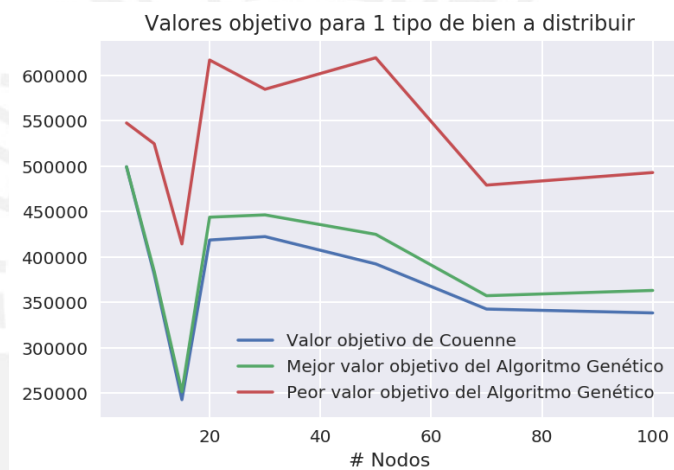
Además del tiempo de ejecución, la otra variable de interés es el valor objetivo resultante, con el fin evaluar la efectividad del algoritmo genético para encontrar soluciones cercanas al óptimo global. Para ello, se consideran tanto los mejores valores obtenidos como los peores valores obtenidos. Los resultados se muestran en la siguiente tabla:

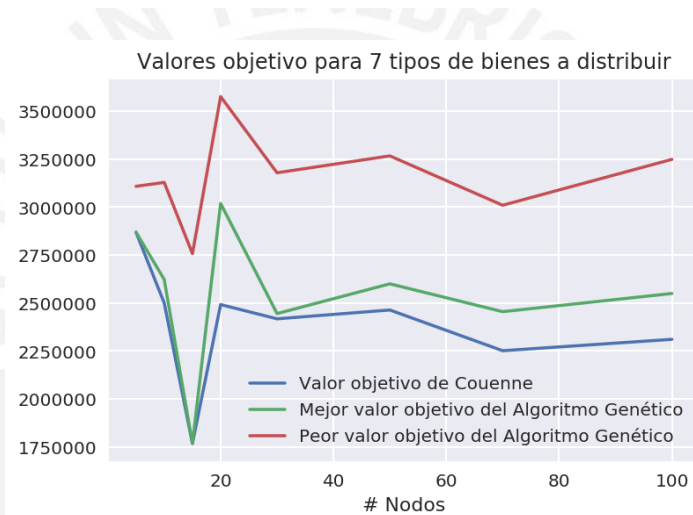
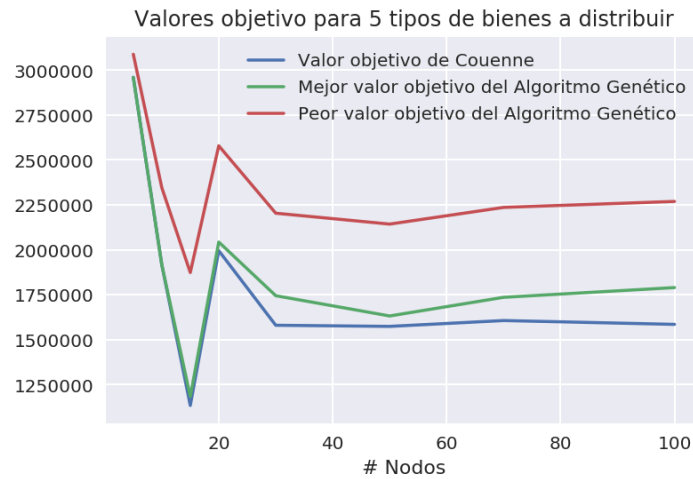
# Nodos	# Bienes	Peor valor objetivo del Algoritmo Genético	Mejor valor objetivo del Algoritmo Genético	Valor objetivo de Couenne
5	1	547553.77	499292.64	499292.64
10	1	524685.11	384201.46	380800.4
15	1	414280.11	251291.44	242676.12
20	1	616894.16	443748.42	418605.89
30	1	584682	446278.61	422356.78
50	1	619375	424834.44	392256.93
70	1	479091.87	357237.89	342532.38
100	1	492932.16	363049.36	338277.44
5	3	1797069.51	1704023.75	1703631.2
10	3	1116437.65	740021.43	699857.49
15	3	1207264.07	847907.14	847907.14
20	3	1486432.66	1103971.02	1096563.1
30	3	1526787.38	1140261.98	1126224.3
50	3	1461477.84	1112968.86	998924.96
70	3	1398111.2	1078372.33	995572.95
100	3	1405081.53	1049903.03	921732.25
5	5	3088212.02	2959872.23	2959872.2
10	5	2346120.96	1917033.44	1917033.44
15	5	1873761.91	1183565.39	1134136.7
20	5	2579031.88	2043506.42	1995369.1
30	5	2203965.49	1744899.61	1580413.7
50	5	2143339.98	1632100.51	1574122.5
70	5	2236221.32	1735917.18	1606624.1
100	5	2269445.8	1790014.89	1585387.9
5	7	3108264.28	2869135.73	2868928.14

10	7	3128556.94	2622670.2	2499959.2
15	7	2757574.81	1767543.47	1767543.47
20	7	3576091.68	3018924.63	2491801
30	7	3178879.11	2444808.87	2417517
50	7	3266988.99	2599899.79	2463475.4
70	7	3009646.86	2454758.82	2251286.2
100	7	3248711.79	2549610.36	2310516.7

Tabla 4: Resultados del valor objetivo de los algoritmos

Para visualizar mejor los resultados se presentan los siguientes gráficos:





A partir de los gráficos se puede notar cómo en los casos de menor tamaño, el algoritmo genético llega a alcanzar el valor óptimo, mientras que, para problemas de mayor tamaño, el algoritmo genético no obtiene el valor óptimo, pero alcanza valores muy cercanos. Para poder cuantificar la diferencia entre los valores objetivo del algoritmo genético y el método exacto, se calcula la diferencia relativa (*relative gap*). La fórmula de la diferencia relativa es la siguiente, donde *opt* es el valor óptimo y *obj* es el valor objetivo alcanzado por el método aproximado:

$$\frac{|obj - opt|}{opt}$$

De esta manera se puede calcular la diferencia porcentual de valores objetivo ajustada a la escala de dichos valores. Los valores de diferencias relativas expresadas en porcentajes para los 32 conjuntos de datos generados son los siguientes:

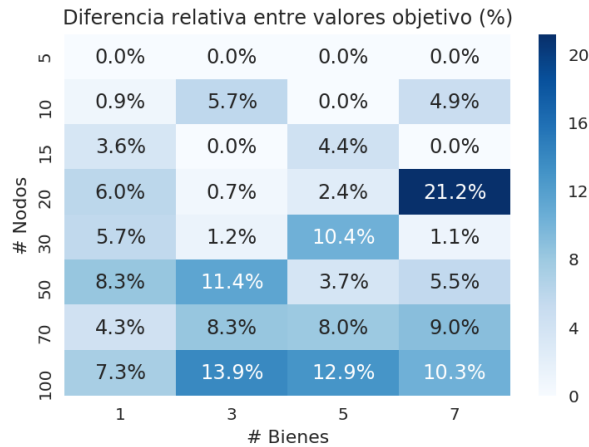


Tabla 5: Diferencia relativa entre valores objetivo

De manera similar a los gráficos anteriores, se puede notar que para una cantidad pequeña de nodos el algoritmo genético llega a alcanzar el valor óptimo (diferencia relativa de 0%), mientras que para problemas de tamaño mayor la diferencia relativa crece hasta alrededor de 10%, con excepción del caso con 7 tipos de bienes y 20 nodos, el cual llegó a un 21.2% de diferencia. Esto puede deberse a que el algoritmo genético se haya quedado en un óptimo local por mucho tiempo, y como no se percibía mejora, se activó la condición de parada de número máximo de generaciones sin mejora.

Para poder entender mejor el beneficio de usar un algoritmo de optimización frente a no usar ningún método de optimización, se presenta a continuación la tabla de diferencias relativas para los valores objetivo más altos encontrados en la generación 0, donde se inicializa la población de forma aleatoria:

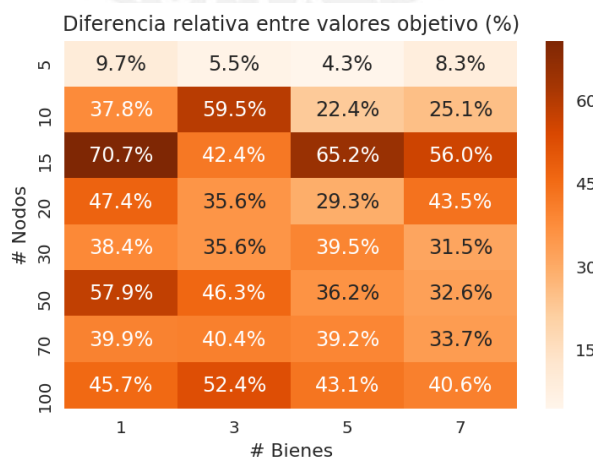


Tabla 6: Diferencia relativa entre valores objetivo - línea base

En el gráfico se puede apreciar que las diferencias relativas van desde 4% en los problemas de menor tamaño hasta 70% en los problemas de mayor tamaño. En general, la mediana se sitúa alrededor de 40%. Esto, comparado con el 10% de diferencia relativa al usar un algoritmo genético evidencia la efectividad del algoritmo genético para obtener soluciones razonables a un mucho menor costo que métodos exactos.

8.4 Discusión y conclusiones

Se ha podido evaluar la efectividad y eficiencia del algoritmo genético diseñado en el problema de distribución de ayuda humanitaria, tomando como referencia un método exacto, el cual es el algoritmo Couenne (*solver*). Mientras que la eficiencia fue evaluada analizando el tiempo de ejecución, donde se encontró que el algoritmo genético era más rápido notoriamente en instancias de mayor tamaño (a partir de 40 nodos en la red de distribución para 1 tipo de bien a distribuir y 10 nodos para 7 tipos de bienes a distribuir), la eficacia fue evaluada analizando el valor de la suma ponderada de las dos funciones objetivo definidas y aplicando la métrica de diferencia relativa entre los valores objetivo encontrados por el algoritmo genético y los valores óptimos encontrados por el método exacto. A partir de esta evaluación se encontró que el algoritmo genético alcanzaba el valor óptimo en instancias pequeñas y obtenía una diferencia relativa del orden de 10% para problemas de mayor tamaño. Adicionalmente se evaluaron soluciones aleatorias obtenidas de la población inicial como línea base, donde se encontró que la diferencia relativa alcanzaba valores del orden de 40%, siendo mucho mayor al 10% obtenido por el algoritmo genético. Esto confirma la hipótesis de que un algoritmo genético es una alternativa aproximada de gran valor práctico para situaciones reales, las cuales se caracterizan por consistir en redes de distribución de gran tamaño, usualmente con más de 30 nodos en la red de distribución.

Capítulo 9. Evaluación del algoritmo genético multiobjetivo basado en el análisis de datos de un escenario real

9.1 Introducción

Después de realizar la evaluación de la eficacia y eficiencia del algoritmo genético, una ventaja adicional del algoritmo genético es su capacidad de manejar varios objetivos al mismo tiempo, sin necesidad de juntarlos en un solo valor, como fue necesario en la sección anterior para que los resultados sean comparables al método exacto. En esta sección se mostrará el funcionamiento de la optimización multiobjetivo aplicado a un escenario real de distribución de ayuda humanitaria en el Perú.

9.2 Contexto del escenario real

El escenario consiste en la distribución de ayuda humanitaria ante los fenómenos de heladas y friaje en el Perú. Este fenómeno se da cada año, entre los meses de abril a setiembre, se presenta un descenso de temperatura que llega a niveles por debajo de los 0°C. Las consecuencias de este fenómeno comprenden el aumento de las infecciones respiratorias, mayor ausentismo escolar, enfermedades y muertes de los animales que son criados por las familias que viven en dichas zonas, además de la pérdida de los cultivos locales (Presidencia del Consejo de Ministros, 2017).

Ante esta situación, el gobierno ha tratado siempre de responder enviando ayuda a estos lugares. Desde el año 2012 se aprueba anualmente el Plan Multisectorial ante Heladas y Friaje, que comprende la participación de diversos organismos del Estado Peruano, como el Centro Nacional de Estimación, Prevención y Reducción del Riesgo de Desastres (CENEPRED), el Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI), los Ministerios de Salud, Vivienda, Educación, Agricultura y Riego (AGRORURAL) y el Instituto Nacional de Estadística e Informática (INEI). En la fase inicial del proceso de planificación anual, dichas entidades proporcionan información para la determinación de las zonas de intervención, los cuales consisten en diversos indicadores tales como: rangos de temperatura, índice de daños de salud, desnutrición crónica infantil en niños menores de 5 años, indicador del material predominante de las viviendas, porcentaje de instituciones educativas con escasa capacitación, número de locales escolares sin acondicionamiento térmico y en riesgo ante heladas, pobreza extrema, analfabetismo, así como dificultad de accesibilidad del Estado. Todos estos niveles de susceptibilidad y exposición socioeconómicos son considerados para el cálculo de la probabilidad de riesgo por distrito, como parte de la elaboración del

escenario de riesgos y la priorización de las zonas de intervención (Presidencia del Consejo de Ministros, 2017).

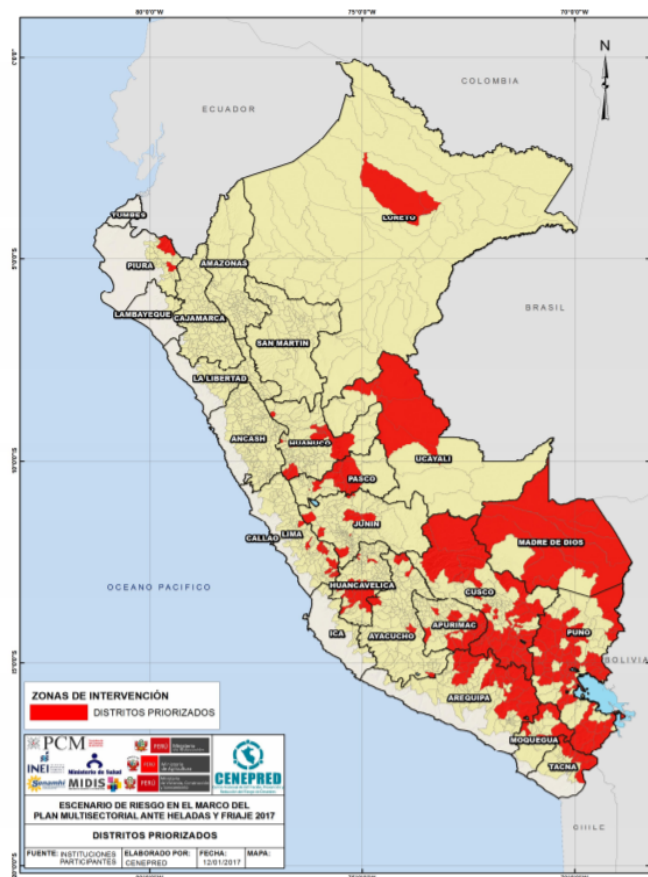


Ilustración 13: Escenario de riesgos por heladas y friaje 2017 (CENEPRED, SGRD, MINSA, MVCS, MINEDU, MIMP y MINAGRI)

De esta manera, en el año 2017 se identificaron los 193 distritos con zonas más críticas ante heladas en 13 departamentos de la zona alto andina del país. Para la distribución de los bienes de ayuda, se tienen 12 almacenes regionales distribuidos en el territorio nacional. Los bienes de ayuda se distribuyen a través de tambos, los cuales son puntos intermedios para los bienes recibidos de los almacenes regionales.

9.3 Recolección y construcción del conjunto de datos

Para construir el conjunto de datos de entrada a ser usados por el algoritmo de optimización, se realizó un proceso de y limpieza de información a partir de datos disponibles en el Plan Multisectorial antes Heladas y Friaje 2017, así como el portal del Centro Nacional de Estimación, Prevención y Reducción del Riesgo de Desastres (CENEPRED). Dentro de ellos se encontraba información sobre el nivel de riesgos por

heladas y friaje según distrito, junto con el listado de tambos, con los cuales fue posible determinar 54 puntos de demanda y 53 tambos. Asimismo, se obtuvo la información de los almacenes nacionales y regionales disponibles a partir del compendio estadístico de prevención y desastres del INDECI, con el que se obtuvo un total de 61 almacenes regionales situados en los departamentos de los distritos priorizados para la distribución de ayuda humanitaria y un total de 25 almacenes situados en otros departamentos del país.

N°	Departamento	Provincia	Distrito	Parámetros de Susceptibilidad		Parámetros de Exposición Socioeconómicos						Elementos expuestos				
				Tª Mínima F10 (°C)	Frecuencia de heladas (días/mes)	Indicador de daños a la Salud (IDS)		Tasa de Desnutrición Crónica Infantil	Tasa de Analfabetismo	Porcentaje de viviendas según el material utilizado para cocinar	Indicador de Material Predominante de la Vivienda	Indicador de Características de la Institución Educativa	Población Estimada al año 2016			
						Tasa de mortalidad (menores a 5 años)	Tasa de neumonías (menores a 5 años)						Total del distrito	Población menor a 5 años	Población de 60 años a más	
1	APURIMAC	ABANCAY	ORCA	-14 a -11	5 a 10 días	0.0	0.0	40.43	20.0	21.91	98.79	0.10	73.08	2500	248	418
2	APURIMAC	ABANCAY	LAMBRAÑA	-14 a -11	5 a 10 días	0.0	4.1	67.48	24.8	36.39	96.25	0.27	81.25	5568	468	598
3	APURIMAC	ANDAHUAYLAS	PAMPACHIRI	-14 a -11	5 a 10 días	0.0	2.7	75.28	38.3	25.90	97.63	0.23	72.73	2795	371	331
4	APURIMAC	ANDAHUAYLAS	SAN JERÓNIMO	-14 a -11	5 a 10 días	0.0	2.7	38.15	30.6	22.21	76.45	0.09	80.63	2897	342	1620
5	APURIMAC	ANTABAMBA	ANTABAMBA	-14 a -11	5 a 10 días	0.0	2.8	71.23	13.8	21.69	89.96	0.37	80.03	3164	395	451
6	APURIMAC	ANTABAMBA	JUAN ESPINOZA MEDRANO	-14 a -11	5 a 10 días	0.0	0.0	59.30	35.4	21.95	93.50	0.33	63.64	2049	241	326
7	APURIMAC	ANTABAMBA	OROPEZA	-14 a -11	10 a 20 días	0.0	0.0	66.62	32.1	22.36	90.01	0.14	80.04	3113	409	283
8	APURIMAC	ANTABAMBA	SABAINO	-14 a -11	5 a 10 días	0.0	0.0	66.63	29.3	16.81	87.53	0.33	62.50	1653	188	250
9	APURIMAC	COTABAMBA	TAMBORIMBA	-11 a -5	10 a 20 días	0.0	0.7	64.48	37.8	31.06	91.32	0.22	75.41	1160	148	850
10	APURIMAC	COTABAMBA	COYLLURQUI	-11 a -5	10 a 20 días	0.0	1.7	70.56	27.3	29.06	90.29	0.16	68.00	856	1164	725
11	APURIMAC	COTABAMBA	HAQUIRA	-14 a -11	20 a 31 días	0.0	1.4	79.06	35.7	25.25	91.36	0.39	66.67	11833	1471	1064
12	APURIMAC	COTABAMBA	MARA	-11 a -5	10 a 20 días	0.0	2.4	67.88	33.2	28.27	91.92	0.50	68.52	6603	822	631
13	APURIMAC	COTABAMBA	CHALLHUAJUACHO	-14 a -11	10 a 20 días	0.0	0.7	77.43	24.8	24.05	70.16	0.17	70.27	9933	1358	672
14	APURIMAC	GRAU	HUANILATI	-11 a -5	10 a 20 días	0.0	9.2	61.94	39.7	22.46	89.00	0.38	66.67	1667	216	166
15	APURIMAC	GRAU	MICHAELA BASTIDAS	-11 a -5	10 a 20 días	0.0	0.0	59.76	27.2	18.73	97.75	0.27	83.33	1680	172	178
16	APURIMAC	GRAU	PATAYPAMPA	-14 a -11	10 a 20 días	0.0	7.1	74.68	38.3	23.27	98.78	0.18	57.14	1122	142	143
17	APURIMAC	GRAU	PROGRESO	-14 a -11	10 a 20 días	0.0	0.0	60.47	31.1	24.82	83.02	0.31	73.68	3327	461	250
18	APURIMAC	GRAU	TURPAY	-11 a -5	10 a 20 días	0.0	14.9	50.33	34.9	14.01	97.05	0.14	0.00	755	67	109
19	AREQUIPA	AREQUIPA	SAN JUAN DE TARUCANI	-14 a -11	10 a 20 días	0.0	12.8	76.67	26.3	10.93	97.73	0.38	37.63	2169	154	283
20	AREQUIPA	AREQUIPA	YURA	-14 a -11	10 a 20 días	0.0	11.8	18.44	9.0	5.17	8.58	0.05	66.67	26774	2472	1937
21	AREQUIPA	CASTILLA	CHOCO	-14 a -11	10 a 20 días	0.0	0.0	70.89	34.4	29.57	96.02	0.33	75.00	991	86	216
22	AREQUIPA	CAYLLOMA	CALLALI	-20 a -14	20 a 31 días	0.0	6.8	44.18	25.5	13.32	64.40	0.30	66.67	1946	135	343
23	AREQUIPA	CAYLLOMA	CAYLLOMA	-14 a -11	20 a 31 días	0.0	2.8	49.93	31.6	17.15	65.73	0.30	73.68	3076	337	289
24	AREQUIPA	CAYLLOMA	CHUPAMPAMPA	-20 a -14	20 a 31 días	0.0	0.0	28.15	35.2	12.53	85.80	0.17	0.00	653	83	138
25	AREQUIPA	CAYLLOMA	LARI	-20 a -14	20 a 31 días	0.0	0.0	30.89	24.3	18.13	75.35	0.31	0.00	1528	140	247
26	AREQUIPA	CAYLLOMA	SAN ANTONIO DE CHUCA	-20 a -14	20 a 31 días	0.0	39.2	65.19	28.2	12.53	78.17	0.30	83.33	1561	156	215
27	AREQUIPA	CAYLLOMA	SIBAYO	-14 a -11	20 a 31 días	0.0	0.0	41.26	29.5	12.80	63.64	0.32	0.00	661	66	94
28	AREQUIPA	CAYLLOMA	TARAY	-20 a -14	10 a 20 días	0.0	0.0	54.72	11.1	24.55	95.27	0.89	100.00	531	54	132
29	AREQUIPA	CAYLLOMA	TISCO	-20 a -14	20 a 31 días	0.0	0.0	52.55	38.7	12.44	93.53	0.39	60.00	149	149	225
30	AREQUIPA	CAYLLOMA	TUTI	-20 a -14	20 a 31 días	0.0	0.0	22.91	20.4	22.51	81.29	0.17	0.00	744	80	134
31	AREQUIPA	CAYLLOMA	YANQUE	-20 a -14	20 a 31 días	0.0	0.0	35.74	21.2	13.75	59.81	0.16	100.00	2112	147	354
32	AREQUIPA	CONDESUYOS	CAVARANI	-14 a -11	10 a 20 días	0.0	43.7	73.62	30.1	23.85	88.47	0.32	61.94	3085	356	209
33	AREQUIPA	CONDESUYOS	SALAMANCA	-14 a -11	5 a 10 días	0.0	0.0	39.93	38.1	25.64	86.60	0.18	75.00	853	40	199
34	AREQUIPA	LA UNIÓN	ALCA	-14 a -11	5 a 10 días	0.0	0.0	44.85	36.4	24.32	90.18	0.30	71.43	1993	207	304
35	AREQUIPA	LA UNIÓN	HUAYNACOTAS	-14 a -11	5 a 10 días	0.0	7.7	59.64	42.9	20.23	96.95	0.35	73.33	2217	256	290
36	AREQUIPA	LA UNIÓN	PURCA	-14 a -11	10 a 20 días	0.0	2.4	64.31	30.9	33.52	94.26	0.39	57.89	2789	405	324
37	AVACUCHO	CANGALLO	PARAS	-14 a -11	10 a 20 días	0.0	0.0	80.14	40.3	24.58	83.59	0.26	59.09	4542	594	434
38	AVACUCHO	PAUCAR DEL SABA SABA	PAUSA	-11 a -5	10 a 2 días	0.0	0.0	32.63	30.2	10.03	47.43	0.17	66.67	2804	263	313
39	AVACUCHO	PAUCAR DEL SABA SABA	SABA SABA	-5 a 0	2 a 5 días	0.0	0.0	59.99	19.7	22.61	73.91	0.18	50.00	738	76	122
40	AVACUCHO	SUCRE	HUACARÍA	-14 a -11	5 a 10 días	0.0	0.0	61.54	25.3	19.03	96.79	0.17	100.00	681	78	97
41	AVACUCHO	SUCRE	MORCOLLA	-14 a -11	5 a 10 días	0.0	0.0	64.13	40.1	20.88	100.00	0.17	85.71	1050	113	150
42	CUSCO	CUSCO	LLURICA	-14 a -11	10 a 20 días	0.0	0.0	35.64	26.9	34.41	97.75	0.16	86.67	2308	269	325
43	CUSCO	ACOMAYO	ACOPA	-11 a -5	20 a 31 días	0.0	0.0	46.37	24.7	21.46	84.91	0.25	100.00	2351	234	314
44	CUSCO	ACOMAYO	SANGARRA	-11 a -5	20 a 31 días	0.0	0.0	60.09	38.7	21.03	95.10	0.21	81.82	3726	393	469
45	CUSCO	ANTA	HUAROCONDO	-14 a -11	10 a 20 días	0.0	0.0	38.60	21.4	19.70	95.23	0.14	84.21	5803	510	708

Ilustración 14: Vista previa de los datos del Plan Multisectorial ante Heladas y Friaje (PCM, 2017)

En el siguiente gráfico se presenta un resumen de las principales variables de los distritos designados como zonas de intervención:

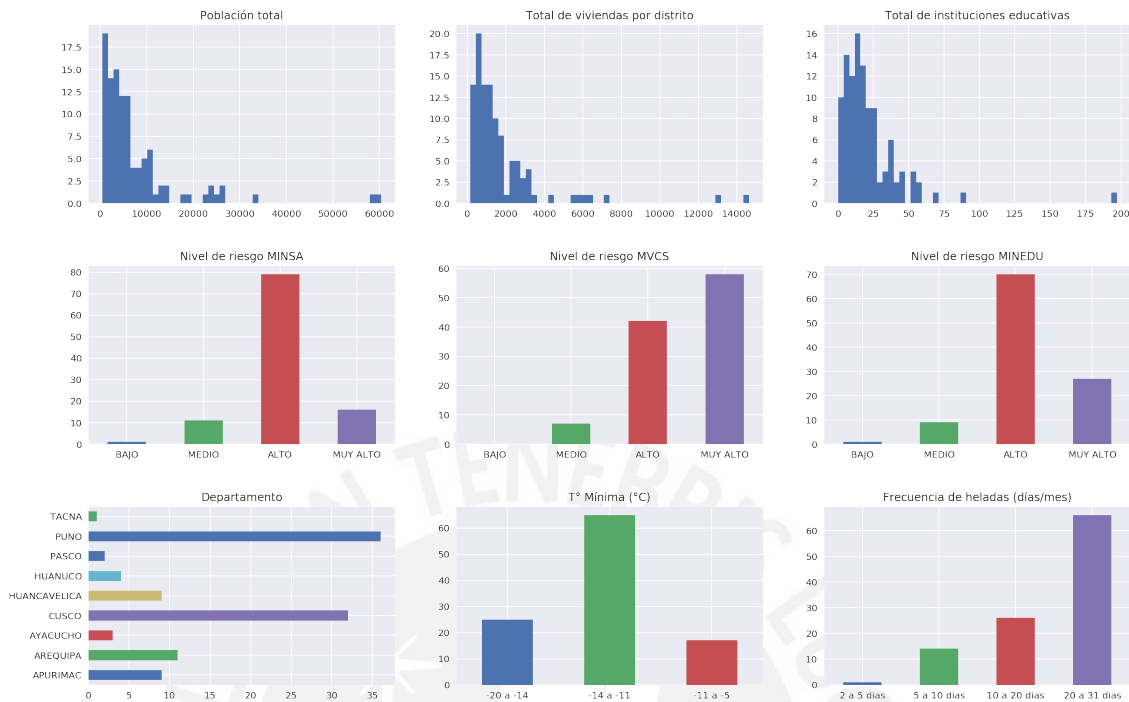
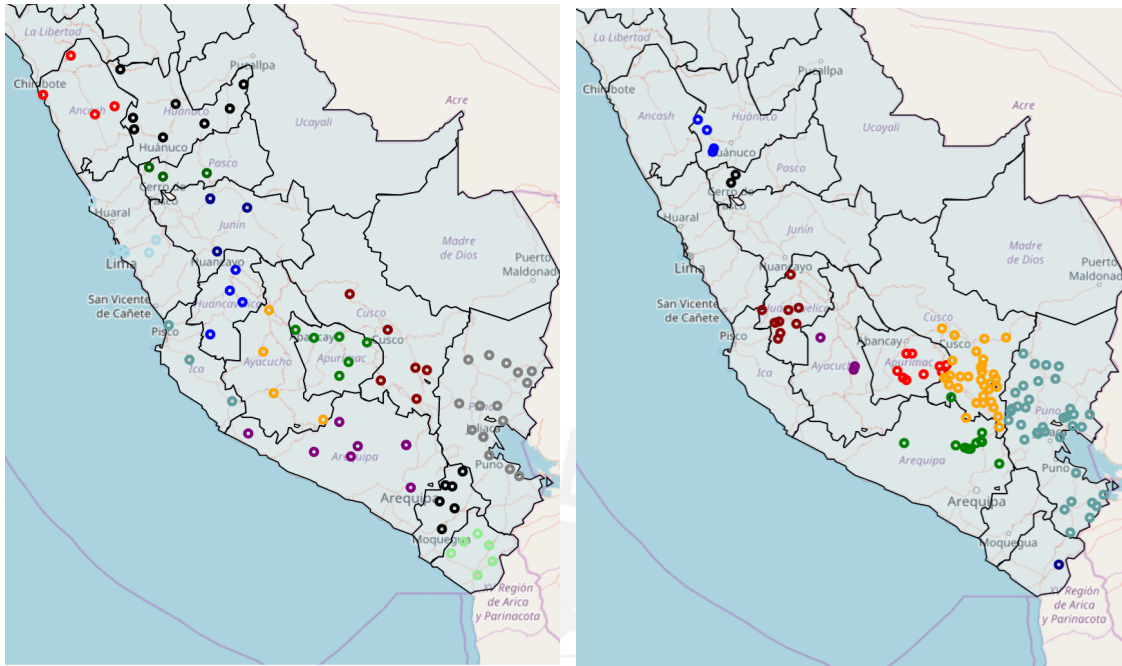


Ilustración 15: Estadísticas de los datos del Plan Multisectorial antes Heladas y Frijaje 2017 (PCM)

En los gráficos se puede notar que la mayoría de los distritos de intervención pertenecen a los departamentos de Puno y Cusco, la población total de cada distrito llega por lo general hasta 15,000 personas, la cantidad de viviendas alcanza generalmente las 4,000 viviendas y la cantidad de instituciones educativas llega por lo general a 50.

Si bien la información de los nodos de la red de distribución incluye el departamento, provincia y distrito de cada ubicación, no se tiene información numérica de las coordenadas de cada punto, lo cual es necesario para estimar la distancia y costo de transporte entre cada par de nodos. Para obtener dichas coordenadas geográficas, se utilizó el servicio web de Google Maps Geolocation API (Google Maps, 2016), el cual permite obtener latitudes y longitudes a partir de nombres de ubicaciones. Para maximizar la efectividad de este servicio, se realizó un proceso de limpieza y procesamiento de texto sobre los campos de departamento, provincia y distrito, con el objetivo de estandarizar y homologar su información. El siguiente gráfico muestra una visualización de los puntos de demanda de los almacenes (izquierda) y los puntos de demanda (derecha) a partir de las coordenadas obtenidas:



Aquí se puede verificar nuevamente que la mayoría de los puntos de demanda provienen de los departamentos de Puno y Cusco.

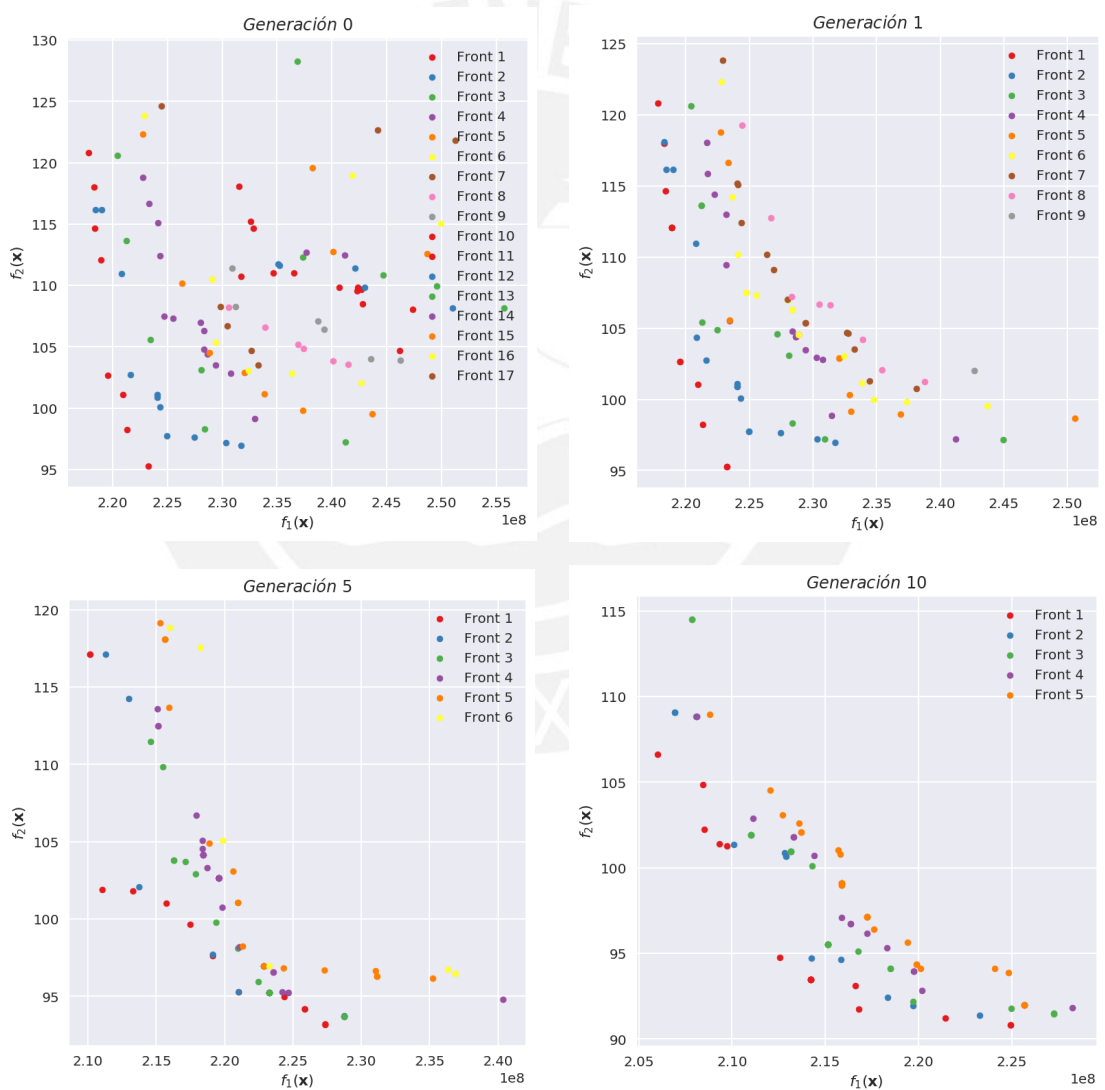
Para determinar la distancia entre cada par de ubicaciones geográficas se utilizó la distancia haversine (Van Brummelen, 2013), la cual mide la distancia entre un par de latitudes y longitudes considerando la forma esférica de la Tierra. Para dos puntos de coordenadas (ϕ_1, ϕ_2) y (λ_1, λ_2) (convertidas a radianes), y el radio de la Tierra R , la distancia viene dada por:

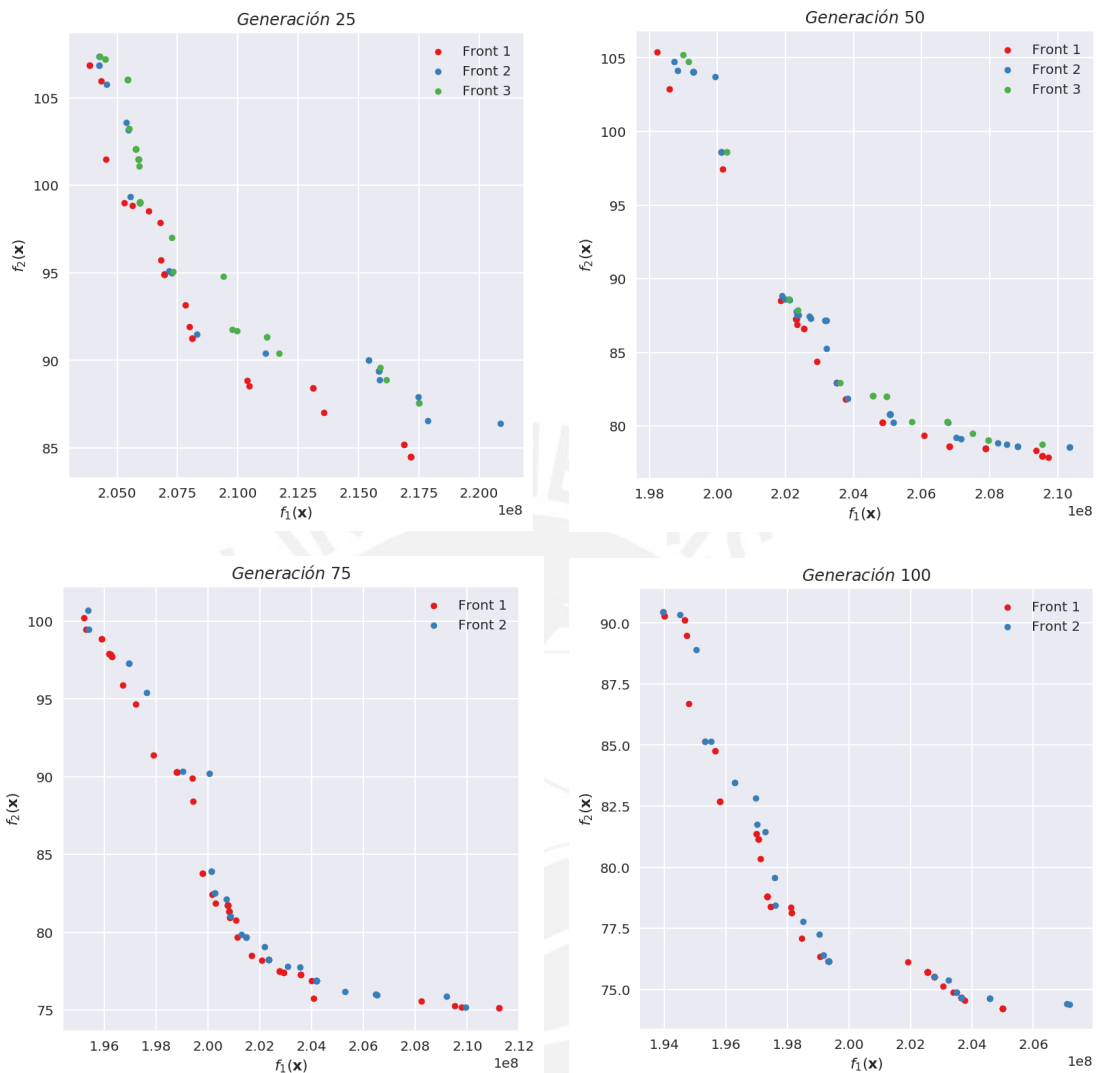
$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

La demanda se estimó a partir de la población total, la cantidad de viviendas y la cantidad de instituciones de cada distrito, con lo cual se tienen 3 tipos de bienes o kits diferentes a distribuir, por persona, vivienda e institución educativa. Para la cantidad total de bienes a distribuir, se asumió una cantidad disponible de bienes del 75% de la demanda por cada tipo de bien, para simular un escenario donde no se puede atender la demanda en su totalidad. De esta manera, se cuenta con una red de distribución de 193 nodos (25 almacenes de departamentos externos, 61 almacenes de los departamentos de las zonas de intervención, 53 tambos y 54 distritos) y 3 tipos de bienes a distribuir.

9.4 Ejecución de los algoritmos de optimización

Se ejecutó el algoritmo genético con una población de 100 individuos durante 100 generaciones, esta vez considerando las dos funciones objetivo por separado, construyendo fronteras de Pareto para determinar los individuos más aptos (individuos no dominados), tal como se describió en la sección 6. El tiempo de ejecución fue de 17 minutos y 51 segundos. El valor de ambas funciones objetivo (costo de transporte e inequidad en la distribución) para la población en diferentes generaciones se presenta en los siguientes gráficos:

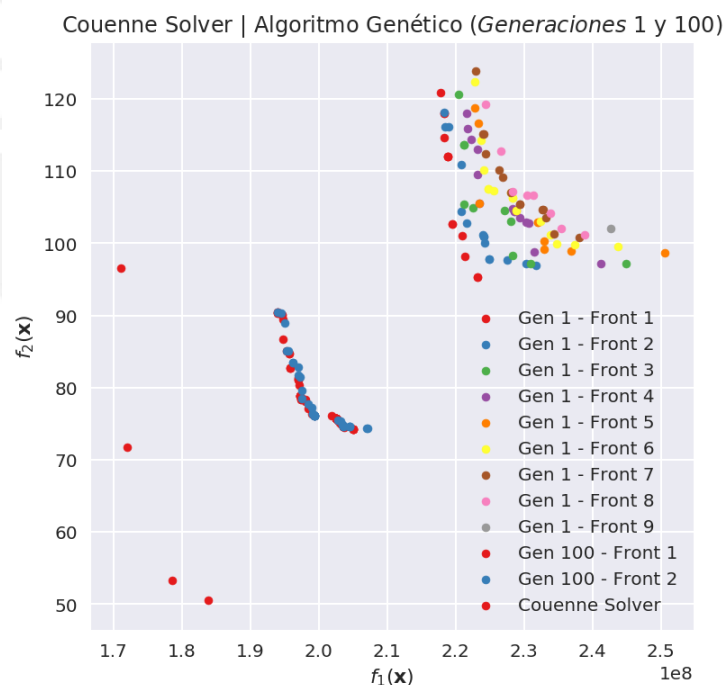




Como el objetivo es minimizar ambas funciones objetivo, se puede apreciar cómo a lo largo de las generaciones, los individuos de la población se van desplazando hacia el origen, lo cual se puede notar observando el desplazamiento los ejes de los gráficos. Asimismo, se consigue observar que los individuos se van agrupando en las fronteras de Pareto descritas en la sección 6. En la población inicial (generación 0), no se ha aplicado el criterio de selección, por lo que la población se encuentra dispersa. A medida que se van escogiendo los individuos dominantes, los individuos dominados van desapareciendo de la población, hasta que al final (generación 100) solo quedan dos frentes, encontrándose en el primer frente el conjunto de soluciones que representa diferentes compromisos entre ambos objetivos de optimización. Hacia la esquina superior izquierda se encuentran las soluciones que priorizan la minimización el costo de transporte, mientras que hacia la esquina inferior derecha se encuentran las

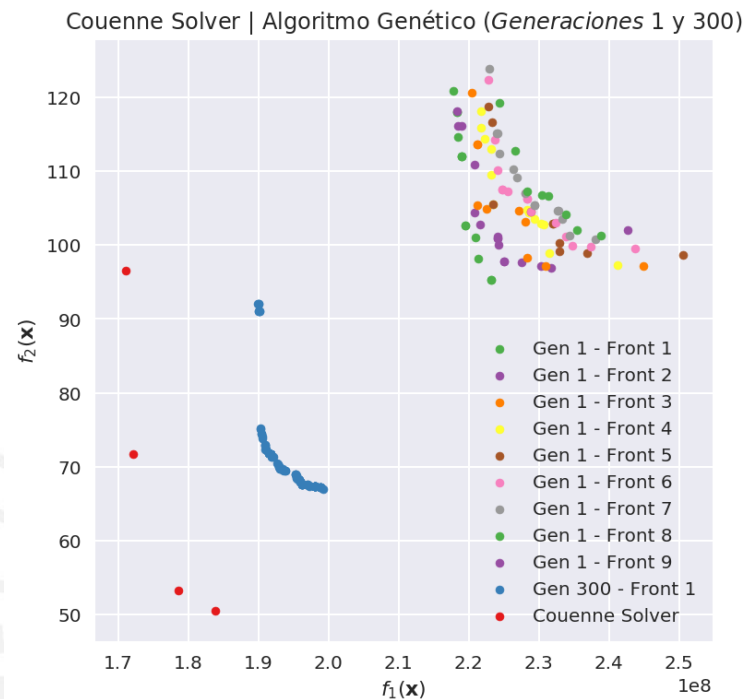
soluciones que priorizan la minimización de la inequidad de distribución. Es importante notar que ninguna solución del primer frente es estrictamente mejor que otra del mismo frente, dado que si presentan un menor valor en un objetivo es porque ha sido compensado con un valor mayor en el otro objetivo.

Para comparar resultados, el método exacto (Couenne) se ejecutó sobre el mismo conjunto de datos, y dado que solo maneja un objetivo a la vez, se probaron 4 diferentes combinaciones de promedio ponderado entre ambos objetivos, dando un peso de 10,000, 100,000, 1,000,000 y 10,000,000 al segundo objetivo (minimización de la inequidad en la distribución). Recordemos que los pesos son valores altos debido a que el segundo objetivo se encuentra en una escala mucho menor que el primer objetivo para que sean comparables. Siendo una red de distribución de 193 nodos y 3 tipos de bienes de distribución, el tiempo de ejecución del método exacto fue de 3 horas y 43 minutos. Los resultados se muestran en el siguiente gráfico:



Los puntos rojos situados en la zona inferior izquierda representan los valores óptimos para las 4 combinaciones de promedios de objetivos usados en el método exacto. El conjunto de puntos situado en la zona superior derecha representa la población durante la primera generación del algoritmo genético. Finalmente, el grupo de puntos al medio del gráfico representa la población en la generación 100. Se puede notar como a lo largo de las generaciones la población se ha ido desplazando hacia los valores óptimos. Para

evaluar cuánto más podrían mejorar las soluciones en el siguiente gráfico se muestra la población en la generación 300, lo cual tomó un tiempo de ejecución de 34 minutos y 40 segundos adicionales. Los resultados se muestran en el siguiente gráfico:



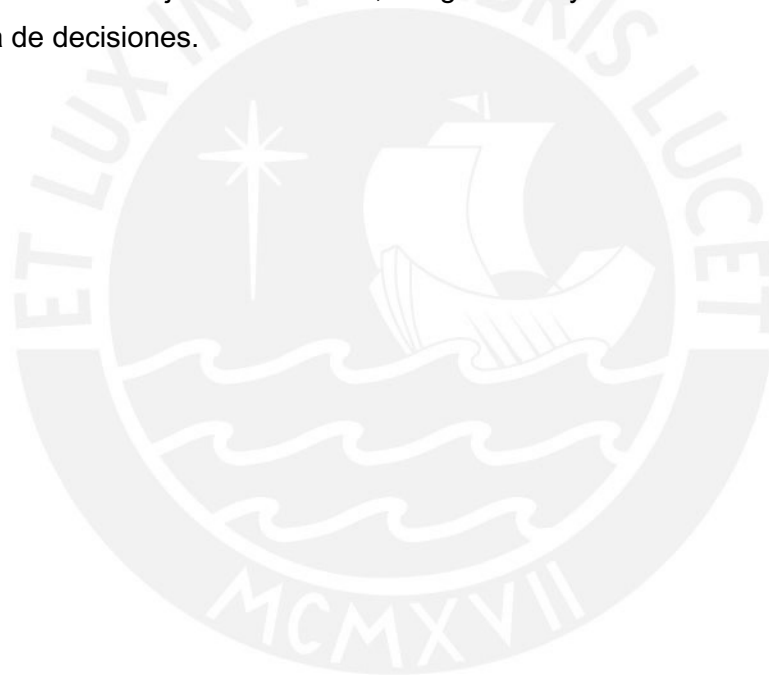
Se puede notar que efectivamente la solución alcanzada por el algoritmo genético mejoró, y toda la población se encuentra en un solo frente de Pareto. Para evaluar la diferencia de valores óptimos se calcula la diferencia relativa por cada objetivo por separado, considerando el mejor valor obtenido para cada objetivo entre todo el frente. Para el primer objetivo (costo de transporte), la diferencia relativa es de 10.97%, mientras que para el segundo objetivo (inequidad en la distribución), la diferencia relativa es de 32.66%. Sin embargo, recordemos que el segundo objetivo representa la suma de cuadrados de los porcentajes de satisfacción. Para llevarlo a la escala original que corresponde a la media cuadrática, los valores del segundo objetivo son divididos entre el total de los puntos de demanda (promedio) y se le aplica la raíz cuadrada:

$$\left(\sqrt{\frac{\sum_i^n x_i^2}{n}} \right)$$

Una vez en la escala original, se vuelve a calcular la diferencia relativa, con lo cual se obtiene un valor de 15.18%. Estos resultados corresponden con los resultados encontrados en la sección de experimentación numérica, donde se encontró que la

diferencia relativa de las soluciones del algoritmo genético con los valores óptimos llegaba era del orden de 10%.

De esta manera, se puede comprobar que las soluciones obtenidas por el algoritmo genético son razonablemente aproximadas al valor óptimo, a un tiempo de ejecución mucho menor. Por otra parte, es importante notar que mientras que se tuvieron que realizar 4 corridas del método exacto para obtener las 4 soluciones óptimas que se visualizan en el gráfico, el algoritmo genético genera todo el conjunto de mejores soluciones en una sola corrida. Este hecho constituye una clara ventaja del algoritmo genético, debido a que al manejar una constante población de soluciones ofrece un conjunto variado de mejores soluciones, otorgando mayor flexibilidad a los responsables de la toma de decisiones.



Capítulo 10. Interfaz de usuario para la ejecución del algoritmo genético y el método exacto

10.1 Introducción

Habiendo concluido la evaluación del algoritmo genético multiobjetivo, en esta sección se describe la interfaz de usuario que se plantea para utilizarlo, junto con el método exacto, el cual es preferible en instancias de menor tamaño, como se vio en la experimentación.

10.2 Ingreso de datos

La entrada de los parámetros del problema de optimización se realiza mediante un archivo de texto, donde se determina la estructura de la red de distribución (número de nodos y las conexiones entre ellos), la cantidad de demanda en cada nodo de la red por cada tipo de bien a distribuir, la cantidad de bienes disponibles en cada nodo asignado como almacén, las capacidades de cada nodo en la red de distribución y los costos de transporte en cada conexión de la red. Una vez elaborado el archivo de entrada, se procede a cargarlo mediante la interfaz para ejecutar los algoritmos de optimización. En la interfaz se encuentra un selector de archivos que permite seleccionar el archivo de entrada que se desea usar para la ejecución de los algoritmos.

10.3 Ejecución de los algoritmos

Mediante la interfaz se puede decidir ejecutar el método exacto o el algoritmo genético. Para el algoritmo genético se puede controlar el número de generaciones que se desea ejecutar el algoritmo y el tamaño de la población que sea evaluar. A medida que se realiza la ejecución de cada algoritmo, se muestra un registro del estado actual de la ejecución, como el número de generaciones transcurridas. Una vez terminados de ejecutarse los algoritmos, se obtiene el registro de ejecución, y la solución óptima (en el caso del método exacto) o las mejores soluciones encontradas por el algoritmo genético multiobjetivo (primer frente de Pareto), los cuales pueden ser exportados a un archivo de texto para ser utilizados por otro programa, de ser necesario. La siguiente figura muestra una vista de la interfaz de usuario:

Dashboard

...

Tamaño de Población 100

Generaciones 40

Dataset

gen	nevals	Avg	Std	Min	Max
0	100	36187.6	2131.91	31460	43480
1	64	35159.6	1819.17	31460	40770
2	56	34125.7	1508.34	31460	37810
3	50	33491.1	1672.24	31190	40730
4	62	32869.2	1462.86	31190	38570
5	59	32513.8	1737.43	29460	40490
6	59	32155.5	1443.29	29460	36770
7	53	31864.4	1203.42	29460	36570
8	62	31696.6	1131.45	29460	36160
9	68	31367.3	1185.29	29460	38230
10	63	31084	1282.39	29600	35650
11	69	31203.3	1845.14	29600	39590
12	60	30918.3	1749.55	29600	39400
13	59	30851.1	1460.89	29600	35470
14	58	30916.1	1742.46	27920	37660
15	60	30864.8	1872.05	27920	39000
16	66	31073.8	2191.62	27920	37760
17	73	30584.9	2053.85	27730	35910
18	64	29958.8	2062.34	27720	37150
19	62	29305.5	2038.67	27720	35980
20	52	28765	1779.03	27720	34360

Ejecutar Algoritmo Genético Mostrar registro de optimización
 Ejecutar Couenne Solver Mostrar solución

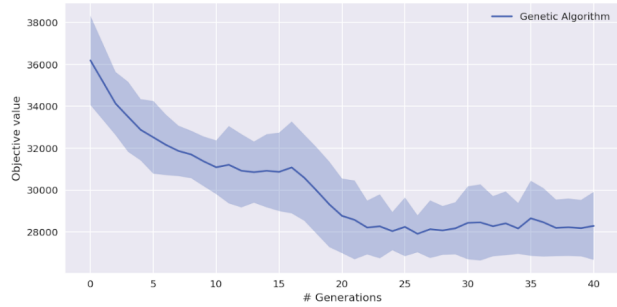
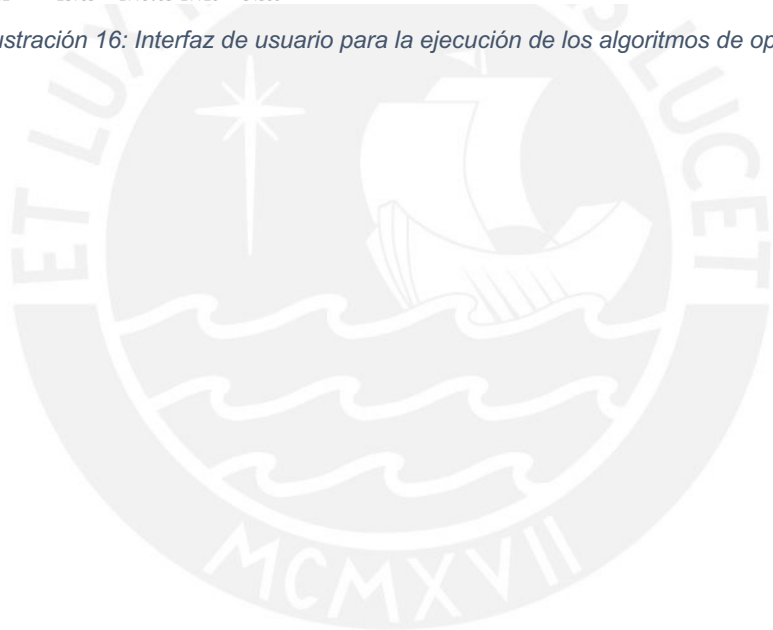


Ilustración 16: Interfaz de usuario para la ejecución de los algoritmos de optimización



Capítulo 11. Conclusiones y trabajos futuros

11.1 Conclusiones

Luego de haber realizado el presente proyecto de fin de carrera, se ha podido llegar a las siguientes conclusiones para cada objetivo planteado:

Para el objetivo de la formulación del problema, se ha visto que el problema de distribución de ayuda humanitaria toma como base la formulación del problema de transporte, considerando restricciones de flujos de entrada y salida de bienes, así como capacidades máximas y costos de transporte. Sin embargo, en el caso de distribución de ayuda humanitaria, es importante tomar en cuenta la equidad en la distribución, el cual expande la formulación a un problema de optimización no lineal entero mixto con múltiples objetivos, debido además a que la variable de decisión debe mantener un valor entero. Ante esto se ha propuesto un algoritmo genético para abordar dicho problema, bajo la hipótesis que presentaría una solución razonablemente aproximada al valor óptimo, pero a un tiempo de ejecución mucho menor.

Sobre el objetivo de estructuras de datos que soporten el algoritmo genético, se ha visto que la representación del cromosoma es uno de los factores más importantes para garantizar la efectividad de los algoritmos genéticos. Mediante la elección de una representación más compacta que la variable de decisión original, se ha simplificado el espacio de búsqueda y los operadores del algoritmo genético, lo cual ha permitido que alcance el desempeño visto durante la experimentación numérica.

Para el objetivo de diseño del algoritmo genético, se ha encontrado que gracias a la representación realizada para el cromosoma los operadores de generación de población inicial, selección, cruce y mutación han sido compactos sin requerir operadores de reparación. Asimismo, para la fase de selección se ha presentado cómo se pueden manejar múltiples objetivos a la vez mediante los conceptos de dominancia y frentes de Pareto, a partir de los cuales se tienen criterios para seleccionar a los individuos más aptos a lo largo de las diferentes generaciones.

Para la implementación del algoritmo genético se ha tomado en cuenta consideraciones como habilitar la optimización con múltiples objetivos y con un solo objetivo, debido a que la experimentación numérica requería un solo valor de optimización para que sea comparable con el método exacto.

Sobre el objetivo de la formulación del modelo matemático de programación no lineal entero mixto para resolver el problema de distribución de ayuda humanitaria de forma exacta, se ha definido el modelo matemático a partir de la formulación del problema previamente planteada, y se ha determinado un algoritmo de optimización exacto (Couenne) que realice la búsqueda del valor óptimo a partir de dicho modelo matemático implementado en el lenguaje de modelado AMPL.

A través de la experimentación numérica realizada con diversos conjuntos de datos de diferentes tamaños generados aleatoriamente y a través de la comparación con un algoritmo exacto para resolver el mismo problema, se ha confirmado la hipótesis de que el algoritmo genético ofrece soluciones razonablemente aproximadas del orden de 10% de diferencia con el valor óptimo y a un tiempo de ejecución mucho menor, en especial para instancias de tamaño mayor. Además, se ha encontrado a partir de qué tamaño del conjunto de datos resulta preferible utilizar un algoritmo genético en lugar de un método exacto, el cual es aproximadamente a partir de 45 nodos en la red de distribución para casos con un solo tipo de bien a distribuir y a partir de 10 nodos para casos con 7 tipos diferentes de bienes a distribuir.

A través del análisis de un escenario real de distribución de ayuda humanitaria, se ha conseguido mostrar una de las ventajas adicionales de la optimización multiobjetivo con algoritmos genéticos. A través del concepto de dominancia y fronteras de Pareto se ha visto cómo determinar la aptitud de los individuos de una población tomando en cuenta varios objetivos a la vez, así como la gran utilidad de determinar un conjunto de mejores soluciones bajo diferentes compromisos entre los objetivos de optimización como parte de un sistema de apoyo a la toma de decisiones.

Finalmente, se ha elaborado una interfaz de usuario que permita utilizar ambos métodos de optimización para nuevas instancias del problema de optimización de ayuda humanitaria, permitiendo también visualizar el desempeño de los algoritmos y exportar las soluciones generadas.

Como objetivo general, se ha presentado el diseño e implementación de un algoritmo genético multiobjetivo para el problema de optimización de distribución de ayuda humanitaria en caso de desastres y fenómenos naturales en el Perú. Se ha evaluado la efectividad y eficiencia del algoritmo genético frente a métodos exactos y se puede concluir que es viable su utilización para escenarios reales y además resulta muy adecuado como un componente de un sistema de apoyo a la toma de decisiones, debido

a la diversidad de mejores soluciones que genera gracias a la utilización de múltiples objetivos de optimización.

11.2 Trabajos futuros

A partir de lo desarrollado en el presente proyecto de fin de carrera se proponen las siguientes investigaciones futuras:

- Implementar un sistema de apoyo a la toma de decisiones de distribución de ayuda humanitaria que maneje la carga y procesamiento de información y utilice los algoritmos de optimización planteados para generar planes de distribución de manera automática.
- Incluir más objetivos de optimización, como la prioridad de bienes específicos a distribuir y más tipos de costos asociados al proceso de distribución de ayuda humanitaria.
- Incluir el subproblema de optimizar la asignación de bienes de distribución a vehículos disponibles en los nodos de distribución. Este problema toma como base el problema de la mochila.
- Realizar la comparación del algoritmo genético planteado con otros tipos de metaheurísticas que también realicen optimización con múltiples objetivos.

Referencias

- AMPL. (2018). AMPL | Streamlined Modeling for Real Optimization. Retrieved April 14, 2018, from <https://ampl.com/>
- Anderson, D. J. (2014). *Kanban: successful evolutionary change in your technology business*.
- Berkoune, D., Renaud, J., Rekik, M., & Ruiz, A. (2012). Transportation in disaster response operations. *Socio-Economic Planning Sciences*. <https://doi.org/10.1016/j.seps.2011.05.002>
- Blecken, A., Danne, C., Dangelmaier, W., Rottkemper, B., & Hellingrath, B. (2010). Optimal Stock Relocation under Uncertainty in Post-Disaster Humanitarian Operations. In *2010 43rd Hawaii International Conference on System Sciences* (pp. 1–10). IEEE. <https://doi.org/10.1109/HICSS.2010.296>
- Campbell, A. M., Vandenbussche, D., & Hermann, W. (2008). Routing for Relief Efforts, *42*(2), 127–145. <https://doi.org/10.1287/trsc.1070.0209>
- Caunhye, A. M., Nie, X., & Pokharel, S. (2012). Optimization models in emergency logistics: A literature review. *Socio-Economic Planning Sciences*, *46*(1), 4–13. <https://doi.org/10.1016/J.SEPS.2011.04.004>
- Christopher, M., Tatham, P., & Chartered Institute of Logistics and Transport in the UK. (2011). *Humanitarian logistics: meeting the challenge of preparing for and responding to disasters*. Kogan Page. Retrieved from <https://muse.jhu.edu/article/481563/pdf>
- Couenne. (2006). Couenne. Retrieved May 27, 2018, from <https://projects.coin-or.org/Couenne>
- Foundation, P. S. (2018). Welcome to Python.org. Retrieved April 14, 2018, from <https://www.python.org/>
- Gen, M., Altıparmak, F., & Lin, L. (2006). A genetic algorithm for two-stage transportation problem using priority-based encoding. *OR Spectrum*, *28*(3), 337–354. <https://doi.org/10.1007/s00291-005-0029-9>
- Geolocation API | Google Developers. (2016). Retrieved May 27, 2018, from <https://developers.google.com/maps/documentation/geolocation/intro>
- Google. (2018). Google Optimization Tools | Optimization | Google Developers. Retrieved April 14, 2018, from <https://developers.google.com/optimization/>
- Google Maps. (2016). Geolocation API | Google Developers. Retrieved May 27, 2018, from <https://developers.google.com/maps/documentation/geolocation/intro>
- Hillier, F. S., & Lieberman, G. J. (2015). *Introduction to operations research*.
- Huang, M., Smilowitz, K., & Balcik, B. (2011). Models for Relief Routing: Equity, Efficiency and Efficacy. *Procedia Social and Behavioral Sciences Procedia -Social and Behavioral Sciences*, *17*(0), 416–437. <https://doi.org/10.1016/j.sbspro.2011.04.525>
- INDECI. (2008). Lecciones Aprendidas del Sur. Retrieved from <http://bvpad.indeci.gob.pe/doc/pdf/esp/doc1259/doc1259-contenido.pdf>
- Jupyter, P. (2018). Project Jupyter. Retrieved April 14, 2018, from <http://jupyter.org/>
- Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering and System Safety*. <https://doi.org/10.1016/j.ress.2005.11.018>
- Kumar, S., & Havey, T. (2013). Before and after disaster strikes: A relief supply chain decision support framework. *International Journal of Production Economics*. <https://doi.org/10.1016/j.ijpe.2013.05.016>
- La Jornada. (2007). La ayuda no llega a los más necesitados, denuncian en Perú - La Jornada. *La Jornada*. Retrieved from

- <http://www.jornada.unam.mx/2007/08/20/index.php?section=mundo&article=027n1mun>
- Luke, S. (2015). *Essentials of metaheuristics : a set of undergraduate lecture notes*.
- Microsoft. (2018). Visual Studio Code - Code Editing. Redefined. Retrieved April 14, 2018, from <https://code.visualstudio.com/>
- Nolz, P. C., Doerner, K. F., Gutjahr, W. J., & Hartl, R. F. (2010). A Bi-objective Metaheuristic for Disaster Relief Operation Planning (pp. 167–187). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-11218-8_8
- Ortuño, M. T., Cristóbal, P., Ferrer, J. M., Martín-Campo, F. J., Muñoz, S., Tirado, G., & Vitoriano, B. (2013). Decision Aid Models and Systems for Humanitarian Logistics. A Survey. *Atlantis Computational Intelligence Systems*, 7. https://doi.org/10.2991/978-94-91216-74-9_2
- Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization : algorithms and complexity*. Prentice Hall.
- Presidencia del Consejo de Ministros. (2017). PLAN MULTISECTORIAL ANTE HELADAS Y FRIAJE 2017. Retrieved from http://www.pcm.gob.pe/wp-content/uploads/2017/05/PMAHF_2017.pdf
- Razali, N. M., & Wah, Y. B. (2011). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests - Semantic Scholar. Retrieved April 14, 2018, from <https://www.semanticscholar.org/paper/Power-comparisons-of-Shapiro-Wilk%2C-Lilliefors-and-Razali-Wah/dcdc0a0be7d65257c4e6a9117f69e246fb227423>
- SAP. (2018). VSR Optimization. Retrieved from https://help.sap.com/doc/saphelp_tm80/8.0/en-US/4a/5a56e958b14204b9d9a65fed58ea6f/content.htm?no_cache=true
- Schrijver, A. (2003). *Combinatorial optimization : polyhedra and efficiency*. Springer.
- Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3–4), 591–611. <https://doi.org/10.1093/biomet/52.3-4.591>
- Sheng, S., Dechen, Z., & Xiaofei, X. (2006). Genetic Algorithm for the Transportation Problem with Discontinuous Piecewise Linear Cost Function. *IJCSNS International Journal of Computer Science and Network Security*, 6(7A). Retrieved from <https://pdfs.semanticscholar.org/e591/b7cc76501ed2b371008d286ced7a2a4ded37.pdf>
- SimpliRoute. (2018). SimpliRoute. Retrieved April 14, 2018, from <https://www.simpliroute.com/>
- Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics. Retrieved from <https://dl.acm.org/citation.cfm?id=505847>
- Tzeng, G.-H., Cheng, H.-J., & Huang, T. D. (2007). Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E* 43. <https://doi.org/10.1016/j.tre.2006.10.012>
- Van Brummelen, G. (2013). *Heavenly mathematics : the forgotten art of spherical trigonometry*. Princeton University Press. Retrieved from https://books.google.com.pe/books?id=0BCCz8Sx5wkC&pg=PR7&redir_esc=y#v=onepage&q&f=false
- Vignaux, G. A., & Michalewicz, Z. (1991). A genetic algorithm for the linear transportation problem. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(2), 445–452. <https://doi.org/10.1109/21.87092>
- Vitoriano, B., Ortuño, M. T., Tirado, G., Montero, J., Vitoriano, B., Ortuño, M. T., ... Montero, J. (2011). A multi-criteria optimization model for humanitarian aid distribution. *J Glob Optim*, 51(51). <https://doi.org/10.1007/s10898-010-9603-z>
- Vitoriano, B., Ortuño, M. T., Tirado, G., & Montero, J. (2011). A multi-criteria optimization

- model for humanitarian aid distribution. *Journal of Global Optimization*, 51(2), 189–208. <https://doi.org/10.1007/s10898-010-9603-z>
- Yang, X.-S. (2014). Nature-Inspired Optimization Algorithms.
- Young, K. R., & Le??n, B. (2009). Natural Hazards in Peru. Causation and Vulnerability. *Developments in Earth Surface Processes*. [https://doi.org/10.1016/S0928-2025\(08\)10009-8](https://doi.org/10.1016/S0928-2025(08)10009-8)



