

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**FACULTAD DE CIENCIAS E INGENIERÍA**



**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**

**20112449**

**Asesor: Ing. Rony Cueva Moscoso**

**Lima, Marzo de 2018**

## Resumen

## **Tema FCI**

## Tabla de Contenido

Resumen.....	2
Tema FCI .....	3
Tabla de Contenido .....	4
Índice de Figuras.....	<b>Error! Bookmark not defined.</b>
Índice de Tablas.....	<b>Error! Bookmark not defined.</b>
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.....	19
1.3.4 AMPL .....	20
1.3.5 AMPL IDE .....	20
1.3.6 GNU Linear Programming Kit.....	20
1.3.7 Microsoft Excel.....	20
1.3.8 Kanban .....	21
1.3.9 Prueba F de Fisher .....	21
1.3.10 Prueba t de Student .....	21
1.3.11 Prueba Kolmogorov-Smirnov .....	21
1.3.12 Prueba de Shapiro-Wilk .....	22
1.4 Viabilidad .....	22
1.4.1 Viabilidad Técnica .....	22

1.4.2	Viabilidad Temporal .....	22
1.4.3	Viabilidad Económica.....	23
1.4.4	Conclusión .....	23
1.5	Alcance, Limitaciones y Riesgos .....	24
1.5.1	Alcance .....	24
1.5.2	Limitaciones .....	24
1.5.3	Riesgos.....	25
Capítulo 2.	Marco Conceptual .....	26
2.1	Introducción .....	26
2.2	Peligros y desastres .....	26
2.3	Gestión del desastre y logística humanitaria .....	26
2.4	Logística humanitaria en el Perú .....	28
2.5	Optimización Combinatoria .....	28
2.6	Optimización de múltiples objetivos.....	29
2.7	Problema de transporte.....	29
2.8	Problema de transbordo.....	30
2.9	Métodos para la solución de problemas de optimización.....	31
2.9.1	Métodos exactos .....	31
2.9.2	Métodos heurísticos .....	31
2.9.3	Métodos metaheurísticos .....	32
2.9.3.1	Algoritmos Genéticos.....	33
Capítulo 3.	Estado del Arte .....	34
3.1	Introducción .....	34
3.2	Investigaciones Académicas .....	34
3.2.1	Transporte en operaciones de respuesta a desastres ( <i>Transportation in disaster response operations</i> ) .....	34

3.2.2	Enrutamiento para esfuerzos de ayuda humanitaria ( <i>Routing for relief efforts</i> )	34
3.2.3	Modelos para enrutamiento de ayuda humanitaria: Equidad, eficiencia y eficacia ( <i>Models for relief routing: Equity, efficiency and efficacy</i> )	35
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> )	36
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> )	36
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> )	37
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> )	37
3.3	Productos comerciales	38
3.3.1	SimpliRoute	38
3.3.2	SAP Transportation Management VSR Optimization	39
3.3.3	Google Optimization Tools	40
3.4	Revisión y discusión	40
3.5	Conclusiones	41
Capítulo 4. Definición de la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria		
		42
4.1	Introducción	42
4.2	Resultado	42
4.2.1	Parámetros del problema de optimización	42
4.2.1.1	Red de distribución	42
4.2.1.2	Horizonte de planificación de la distribución	42
4.2.1.3	Oferta y demanda inicial en los nodos de la red de distribución	42

4.2.1.4	Vehículos para el transporte de bienes.....	43
4.2.1.5	Volumen y peso de los bienes .....	43
4.2.2	Variables de decisión .....	44
4.2.2.1	Cantidad de vehículos que se movilizan .....	44
4.2.2.2	Cantidad de bienes que se transportan .....	44
4.2.3	Variables auxiliares .....	44
4.2.3.1	Oferta y demanda a lo largo del tiempo .....	44
4.2.3.2	Consumo de bienes.....	44
4.2.3.3	Vehículos disponibles .....	45
4.2.4	Función objetivo .....	45
4.2.4.1	Minimizar demanda actual.....	45
4.2.4.2	Minimizar costo de transporte.....	45
4.2.5	Restricciones .....	46
4.2.5.1	Evitar enviar más bienes de los que se dispone .....	46
4.2.5.2	Evitar usar más vehículos de los que se dispone .....	46
4.2.5.3	Evitar sobrepasar la capacidad de los vehículos .....	47
4.2.5.4	Balance en el inventario .....	47
4.2.5.5	Balance en la cantidad de vehículos disponibles .....	48
4.2.5.6	Determinación del consumo .....	48
4.2.5.7	Actualización de la demanda .....	49
4.2.5.8	Condiciones iniciales del problema.....	49
4.2.6	Rango de existencia.....	49
4.3	Relación con el objetivo .....	50
4.4	Validación de la construcción .....	50
Capítulo 5. Definición de las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria.....		55
5.1	Introducción .....	55

5.2	Resultado.....	55
5.2.1	Estructuras de datos para los parámetros iniciales y variables auxiliares del problema .....	55
5.2.2	Variables de decisión y cromosoma .....	57
5.2.3	Cromosoma .....	58
5.3	Relación con el objetivo .....	62
Capítulo 6. Diseño del algoritmo genético para el problema de optimización de distribución de ayuda humanitaria .....		63
6.1	Introducción .....	63
6.2	Esquema general del algoritmo genético .....	63
6.3	Construcción de la población inicial.....	63
6.3.1	Generación de cromosomas .....	64
6.4	Evaluación de la aptitud .....	64
6.4.1	Frente de Pareto .....	65
6.5	Selección .....	66
6.5.1	Selección por torneo .....	67
6.6	Cruce .....	68
6.7	Mutación .....	70
6.8	Criterio de parada .....	70
Capítulo 7. Experimentación Numérica .....		71
7.1	Introducción .....	71
7.2	Configuración de experimentación .....	71
7.2.1	Tiempo de ejecución .....	71
7.2.1.1	Verificar normalidad.....	72
7.2.1.2	Verificar igualdad de varianzas.....	72
7.2.1.3	Verificar igualdad de medias.....	72
7.2.1.4	Verificar comparación de medias.....	73
7.2.2	Optimalidad.....	73



7.2.2.1	Verificar normalidad.....	74
7.2.2.2	Verificar igualdad de varianzas.....	74
7.2.2.3	Verificar igualdad de medias.....	75
7.2.2.4	Verificar comparación de medias.....	75
Capítulo 8.	Conclusiones y trabajos futuros .....	76
8.1	Conclusiones .....	76
8.2	Trabajos futuros .....	76
Referencias.....		77
Anexos .....		a

## 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). ....	35
Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010). ....	36
Ilustración 4. Matriz de recompensa para cada objetivo de optimización (Vitoriano, Ortuño, Tirado, Montero, et al., 2011).....	38
Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018).....	39
Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018).....	40
Ilustración 7. Región de soluciones dominadas por la solución A en un espacio de 2 objetivos (Luke, 2015) .....	65
Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015).....	66
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). ....	67

## Índice de Tablas

Tabla 1. Cronograma del proyecto .....	23
Tabla 2. Matriz de riesgos .....	25

## 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 los vehículos están sujetos a restricciones de peso y volumen, los cuales limitan la cantidad de bienes que puedan ser transportados y hacen que la decisión de asignación de bienes a vehículos no sea trivial (Christopher, Tatham, & Chartered Institute of Logistics and Transport in the UK., 2011). Es más, en escenarios de distribución a gran escala, la distribución suele realizarse en horizonte de tiempo de días, lo cual añade una dimensión de complejidad al problema de crear un plan de distribución óptimo. Finalmente, el concepto de optimalidad de un plan de distribución puede estar dado por diversos factores, incluyendo el costo de transporte, la atención de la demanda 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, una metaheurística que se presenta como alternativa ideal son los algoritmos evolutivos o genéticos, debido a que ofrecen la robustez necesaria para afrontar este tipo de problemas y su optimización es a partir una población de soluciones candidatas, lo cual facilita tener un conjunto de mejores soluciones, lo cual es importante en optimización multiobjetivo debido a que los objetivos pueden estar enfrentados, lo cual hace necesario llegar a un compromiso entre ellos y contar con un abanico o *pool* de mejores opciones.

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 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 e implementar un programa lineal a partir de la formulación del problema de optimización.

- O 6. Diseñar y desarrollar la experimentación numérica para comparar el desempeño del algoritmo genético con el método de programación lineal.
- 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. Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver (O5).
- R 6. Componente de software que contenga el programa lineal y ejecute el solver (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 basada 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 y vehículos a transportar (O6).
- R 10. Instancias de prueba de diferente tamaño generadas aleatoriamente (O6).
- R 11. Informe de evaluación y comparación de tiempo de ejecución y optimalidad de los algoritmos de optimización mediante experimentación numérica (O6).
- R 12. Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el programa lineal (O7).

#### 1.2.4 Mapeo de objetivos, resultados y verificación

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



Algoritmo genético implementado y contenido en un componente reutilizable	Software	<ul style="list-style-type: none"> <li>- Código del algoritmo genético</li> <li>- Interfaz en consola para ejecutar el algoritmo genético</li> </ul>
<b>Objetivo (O5):</b> Diseñar e implementar un programa lineal a partir de la formulación del problema de optimización		
<b>Resultado (R5)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver	Software	<ul style="list-style-type: none"> <li>- Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver</li> </ul>
<b>Resultado (R6)</b>	<b>Meta física</b>	<b>- Medio de verificación</b>
Componente de software que contenga el programa lineal y ejecute el solver	Software	<ul style="list-style-type: none"> <li>- Código del componente</li> <li>- Interfaz en consola para ejecutar el solver</li> </ul>
<b>Objetivo (O6):</b> Diseñar y desarrollar la experimentación numérica para comparar el desempeño del algoritmo genético con el método de programación lineal		
<b>Resultado (R7)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Análisis y desarrollo de un generador de datos de prueba para los algoritmos de optimización	Software	<ul style="list-style-type: none"> <li>- Código del generador de datos</li> <li>- Interfaz en consola para generar datos</li> </ul>
<b>Resultado (R8)</b>	<b>Meta física</b>	<b>Medio de verificación</b>

Análisis de datos de un caso real para generar un caso de prueba basada en un escenario real	Documento	- Análisis de los datos utilizados de un caso real describiendo dimensiones del problema de un escenario real
<b>Resultado (R9)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Instancia de prueba basada en un escenario real conteniendo información sobre la red distribución, así como la oferta, demanda y tipos de bienes y vehículos a transportar	Dataset	- Archivo con datos describiendo los datos de entrada de un escenario real del problema de optimización
<b>Resultado (R10)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Instancias de prueba de diferente tamaño generadas aleatoriamente	Dataset	- Archivo con datos describiendo datos de entrada de diferente tamaño para los algoritmos de optimización
<b>Resultado (R11)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Informe de evaluación y comparación de tiempo de ejecución y optimalidad de los algoritmos de optimización mediante experimentación numérica	Documento	- Resultados de evaluación y comparación de tiempo de ejecución y optimalidad de los algoritmos de optimización

<b>Objetivo (O7):</b> Desarrollar la interfaz de usuario para la ejecución y visualización de los algoritmos de optimización		
<b>Resultado (R12)</b>	<b>Meta física</b>	<b>Medio de verificación</b>
Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el programa lineal	Software	<ul style="list-style-type: none"> <li>- Código de la interfaz de usuario</li> <li>- Interfaz gráfica para ejecutar los algoritmos de optimización</li> </ul>

### 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 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 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 GNU Linear Programming Kit**

GLPK (GLPK, 2012) es un solver de código abierto orientado a resolver problemas de programación lineal de gran escala (LP), así como programación entera mixta (MIP). 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 Microsoft Excel**

Microsoft Excel (Microsoft, 2016) es una aplicación compuesta de hojas de cálculo desarrollada como parte de la suite Microsoft Office. Posee herramientas de manejo de datos mediante tablas, así como herramientas de visualización y un lenguaje de programación conocido como Visual Basic for Applications. Dado que Microsoft Excel posee funciones estadísticas, se utilizará en el proyecto para realizar la fase de experimentación numérica.

### **1.3.8 Kanban**

Kanban (Anderson, 2014) es un 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.

### **1.3.9 Prueba F de Fisher**

La prueba F de Fisher es una prueba de hipótesis estadística en la que el estadístico de prueba presenta una distribución F bajo una hipótesis nula. Se usa frecuentemente para comparar modelos estadísticos que han sido ajustados a un conjunto de datos, con el fin de identificar el modelo que mejor se adapta a la población de la cual se toma una muestra. En el presente proyecto, forma parte de las pruebas a ser realizadas en la sección de experimentación numérica.

### **1.3.10 Prueba t de Student**

La prueba t de Student es una prueba de hipótesis estadística en la cual el estadístico presenta una distribución t bajo una hipótesis nula. Se usa frecuentemente para determinar si dos conjuntos de datos son significativamente diferentes entre sí. En el presente proyecto, forma parte de las pruebas a ser realizadas en la sección de experimentación numérica.

### **1.3.11 Prueba Kolmogorov-Smirnov**

La prueba de Kolmogorov-Smirnov es una prueba no paramétrica para evaluar la igualdad de distribuciones de probabilidad unidimensional y continua. Se usa frecuentemente para evaluar qué tan bien una función de hipótesis encaja con una función empírica. En el presente proyecto, forma parte de las pruebas a ser realizadas en la sección de experimentación numérica.

### **1.3.12 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	- Sábado 07/07/18

#### 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 de diferentes cantidades y tipos de recursos a diferentes tipos de vehículos, así como una propuesta de ruta a seguir por cada uno de ellos. Esta propuesta estará alineada a múltiples objetivos, tales como la rapidez de distribución, 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.

Asimismo, cabe decir que actualmente solo se están considerando vehículos terrestres en la resolución del problema, mas no vehículos aéreos o marítimos.

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*

<b>Riesgo Identificado</b>	<b>Impacto en el proyecto</b>	<b>Medidas correctivas para mitigar</b>
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 tiene como objetivo encontrar un objeto óptimo en una colección finita de objetos (Papadimitriou & Steiglitz, 1982). Típicamente, dicha colección posee una representación concisa, como una red de transporte, por ejemplo, mientras que el número de objetos es enorme (todas las posibilidades de rutas sobre la red), de tal manera que la estrategia de evaluar todos los objetos uno por uno no es una opción viable.

## 2.6 Optimización de múltiples objetivos

La optimización de múltiples objetivos aborda un tipo especial de problemas de optimización que involucran más de una función objetivo 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 para el caso de múltiples objetivos que no pueden ser comparadas 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  $\mathbf{x}$  es el conjunto de variables de decisión,  $f(\mathbf{x})$  es la función objetivo y  $g(\mathbf{x})$  son funciones que representan las restricciones del problema, 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 aborda encontrar distribuir productos de un grupo de puntos de distribución, 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). Este problema presenta la siguiente formulación:

$$\begin{aligned} \min \quad & \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} \quad & \sum_{j=1}^n x_{ij} = s_i, \text{ para } i = 1, 2, \dots, m \\ & \sum_{i=1}^m x_{ij} = d_j, \text{ para } j = 1, 2, \dots, n \\ & x_{ij} \geq 0, \text{ para todo } i \text{ y } j \end{aligned}$$

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). Este problema presenta la siguiente formulación:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s. t.} \quad & \sum_{j=1}^n x_{ij} - \sum_{j=1}^n x_{ji} = b_i, \text{ para cada nodo } i \\ & 0 \leq x_{ij} \leq u_{ij} \end{aligned}$$

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, pero aún siendo aproximados. Sin embargo, la mayor complejidad de estos métodos los hace más lentos, por lo que su requerimiento computacional es mayor. 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 de menor optimalidad 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 aleatorizada 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 datasets 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 datasets 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:

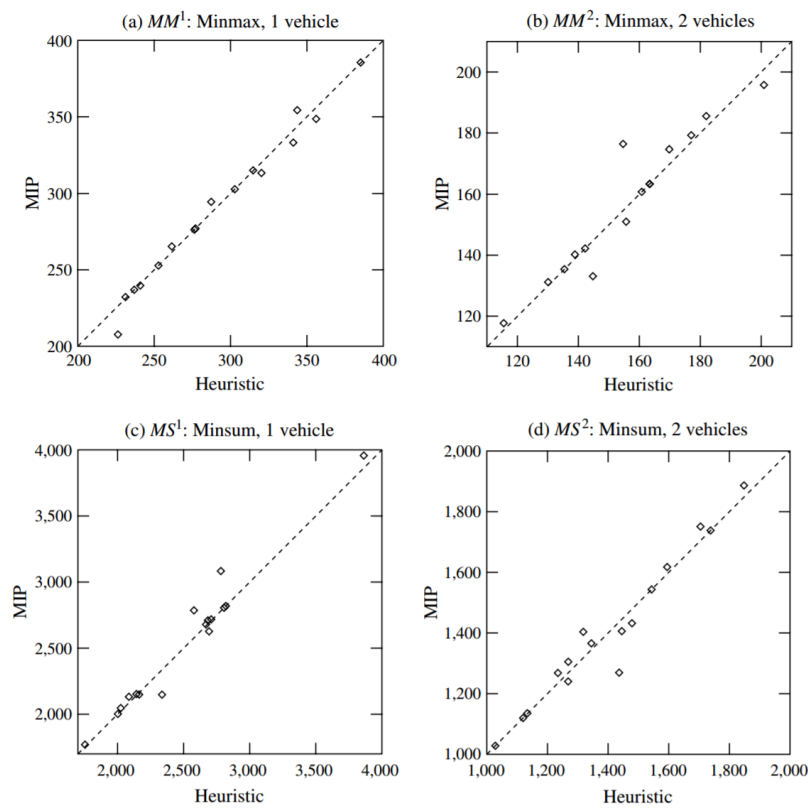


Ilustración 2. Comparación entre resultados de programación entera mixta y heurística (Campbell et al., 2008).

### 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 ejecutó el programa lineal en un 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 dataset compuesto de 6 regiones de Myanmar, con una distribución sobre 13 períodos de tiempo. Los resultados se pueden apreciar en el siguiente gráfico:

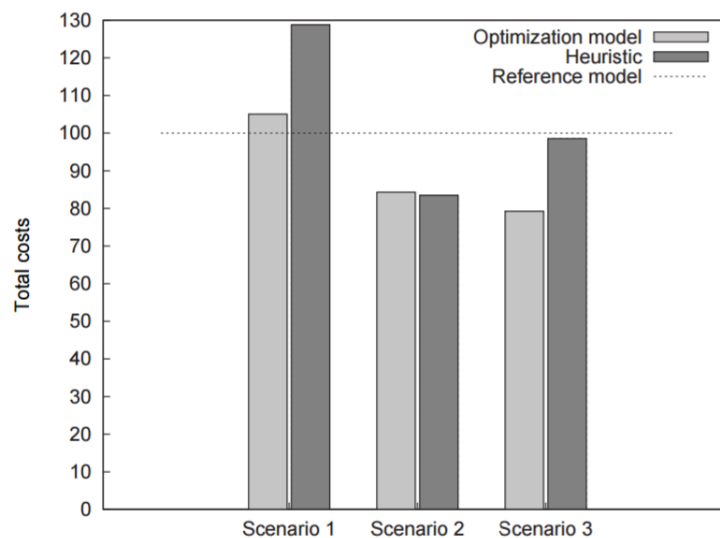


Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010).

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

*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 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.

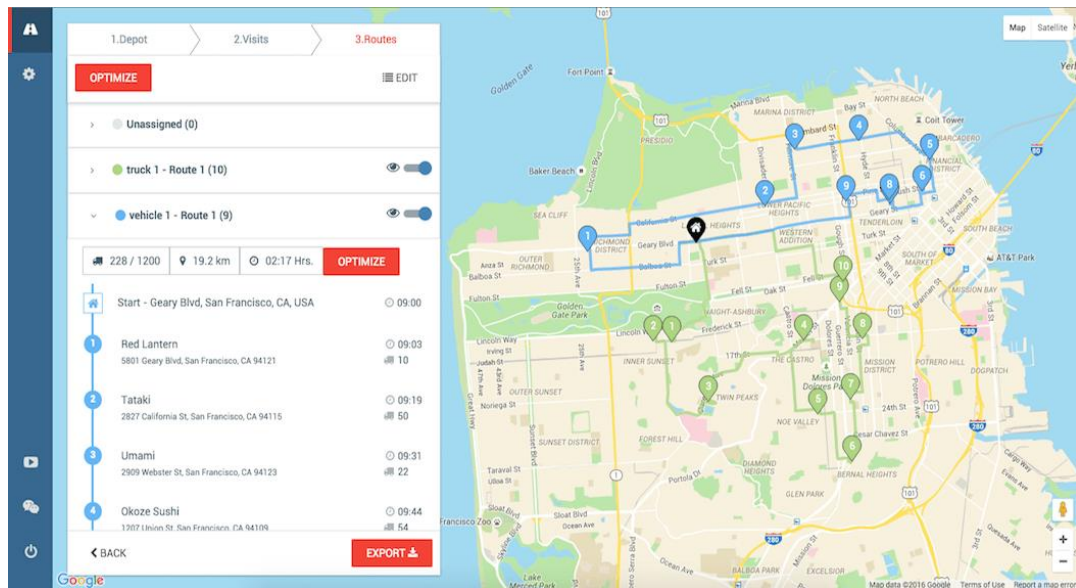


Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018).

### 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).

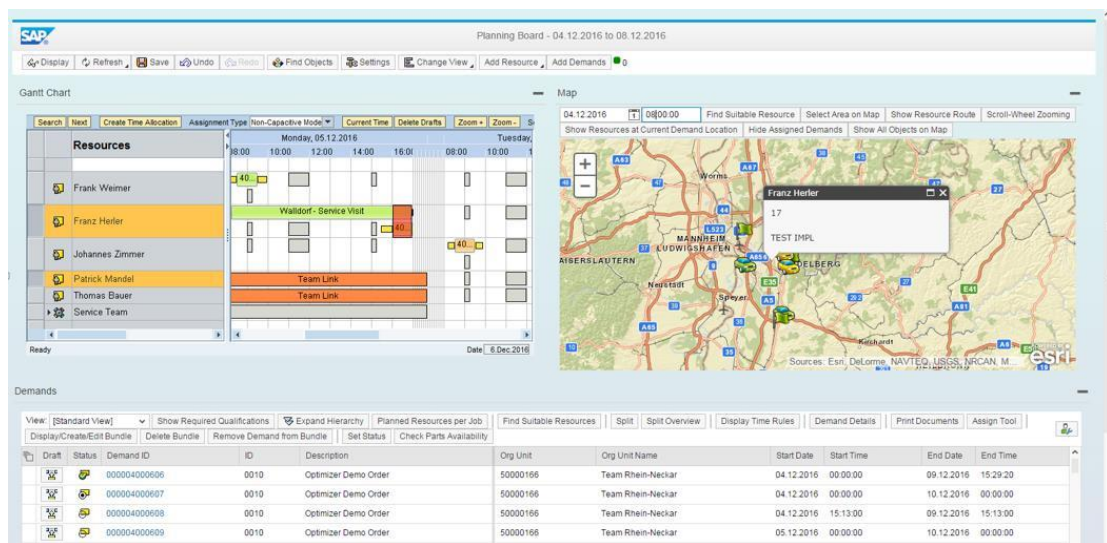


Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018).

### 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  $k$ . En el caso que se está abordando, el cual es distribución a gran escala, los nodos suelen representar distritos conectados por carreteras, siendo la distancia de cada carretera representada por el parámetro ***Distancia<sub>jk</sub>*** la cual influirá en el cálculo del costo de distribución.

##### **4.2.1.2 Horizonte de planificación de la distribución**

Como la distribución de ayuda humanitaria es un proceso que suele abarcar varios días, es necesario un parámetro que acote el horizonte de planificación del problema, la cual queda definida por  **$T$**  que representa el número de períodos (normalmente días).

##### **4.2.1.3 Oferta y demanda inicial 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 período cero. 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 y demanda irá cambiando a lo largo del tiempo según las decisiones de transporte que se tomen como

resultado del algoritmo de optimización. La oferta de bienes se define por el parámetro ***Inventario0<sub>k</sub>***, que representa la cantidad de bienes que existe en el nodo k en el período cero, mientras que la demanda de bienes se define por el parámetro ***Demanda0<sub>k</sub>***, que representa la cantidad de bienes que se requieren en el nodo k en el período cero.

Para considerar bienes de diferentes tipos, estos parámetros se pueden expandir a ***Inventario0<sub>ki</sub>*** y ***Demanda0<sub>ki</sub>***, que indican la oferta y demanda inicial de bienes de tipo i en el nodo k.

#### 4.2.1.4 Vehículos para el transporte de bienes

Los bienes se movilizarán entre los nodos mediante los vehículos, para los cuales se define la capacidad máxima de bienes que pueden transportar tanto en términos de volumen (***CapacidadVehV***) como peso (***CapacidadVehW***), así como el costo de transporte por unidad de distancia (***CostoTransp***). Asimismo, para saber de cuántos vehículos disponemos en cada lugar, también se establece la cantidad inicial de vehículos disponibles en el nodo k, la cual queda definida por el parámetro ***VehDisponibles0<sub>k</sub>***. A medida que los vehículos se desplacen a lo largo de los períodos, la cantidad de vehículos disponibles en cada nodo irá cambiando a lo largo del tiempo según las decisiones de transporte que se tomen como resultado del algoritmo de optimización.

Para considerar vehículos de diferentes tipos, estos parámetros se pueden expandir a ***CapacidadVehV<sub>v</sub>***, ***CapacidadVehW<sub>v</sub>***, ***CostoTransp<sub>v</sub>***, los cuales indican las capacidades y costos de cada tipo de vehículo, y ***VehDisponibles0<sub>kv</sub>***, que indica la cantidad inicial de vehículos disponibles de tipo v en el nodo k.

#### 4.2.1.5 Volumen y peso de los bienes

Para poder restringir la cantidad de bienes que se pueden movilizar en los vehículos, se establecen los parámetros ***Volumen*** y ***Peso***.

Para considerar bienes de diferentes tipos, estos parámetros se pueden expandir a ***Volumen<sub>i</sub>*** y ***Peso<sub>i</sub>***, que indican el volumen y peso de cada bien i. Asimismo, en el contexto de ayuda humanitaria existen bienes que son más prioritarios que otros según el caso, por ejemplo, agua durante una sequía y artículos de primeros auxilios en escenarios en un terremoto. Para considerar esto en el problema, se define el parámetro ***Prioridad<sub>i</sub>***, que indica el nivel de prioridad de cada bien i. Cuanto mayor el valor del parámetro, más alta será la prioridad de dicho bien.

## 4.2.2 Variables de decisión

### 4.2.2.1 Cantidad de vehículos que se movilizan

Los bienes se movilizarán entre los nodos mediante los vehículos, los cuales cambiarán de ubicación a lo largo del tiempo, lo cual origina la variable de decisión que establezca cuántos vehículos serán movilizadas desde el nodo  $j$  al nodo  $k$  en el período  $t$  (***Veh<sub>jkt</sub>***).

Para considerar diferentes tipos de vehículos, esta variable se puede expandir a ***Veh<sub>jkvt</sub>***, que indica el número de vehículos de tipo  $v$  que serán movilizadas del nodo  $j$  al  $k$  en el período  $t$ .

### 4.2.2.2 Cantidad de bienes que se transportan

De la misma manera, se requiere una variable de decisión que establezca cuántos bienes serán transportados en los vehículos, lo cual queda origina la variable de decisión que indique cuántos bienes serán transportados en los vehículos desde el nodo  $j$  al nodo  $k$  en el período  $t$  (***Transp<sub>jkt</sub>***).

Para considerar diferentes tipos de vehículos, esta variable se puede expandir a ***Transp<sub>jkvit</sub>***, que indica la cantidad bienes de tipo  $i$  que serán transportados en vehículos de tipo  $v$  desde el nodo  $j$  hacia el  $k$  en el período  $t$ .

## 4.2.3 Variables auxiliares

### 4.2.3.1 Oferta y demanda a lo largo del tiempo

A partir de las decisiones de transporte que se tomen, la oferta y la demanda en cada ubicación irá siendo actualizada, lo cual quedará indicado por las variables ***Inventario<sub>kt</sub>*** y ***Demanda<sub>kt</sub>***, las cuales indican cuánto de oferta y demanda existe en el nodo  $k$  en el período  $t$ .

Para considerar diferentes tipos de bienes, estas variables se pueden expandir a ***Inventario<sub>kit</sub>*** y ***Demanda<sub>kit</sub>***, que indican cuánto de oferta y demanda del bien  $i$  existe en el nodo  $k$  en el período  $t$ .

### 4.2.3.2 Consumo de bienes

En el momento en que los bienes lleguen al nodo actual, dicho bien será consumido para satisfacer la posible demanda existente, lo cual quedará registrado en la variable ***Consumo<sub>kt</sub>***, que indica cuántos bienes se consumen en el nodo  $k$  en el período  $t$ .

De la misma manera, para considerar diferentes tipos de bienes, esta variable puede ser expandida a **Consumo<sub>kit</sub>**, para indicar el consumo del bien i en el nodo k en el período t.

#### 4.2.3.3 Vehículos disponibles

De una manera análoga al control del inventario de los bienes, se debe tener un control de cuántos vehículos se encuentran en cada ubicación en cada período. Esto queda representado por la variable **VehDisponibles<sub>kt</sub>**, que indica la cantidad de vehículos disponibles en el nodo k en el período t.

Para considerar diferentes tipos de vehículos, esta variable se puede expandir a **VehDisponibles<sub>kvt</sub>**, que indica la cantidad de vehículos de tipo v disponibles en el nodo k en el período t.

#### 4.2.4 Función objetivo

##### 4.2.4.1 Minimizar demanda actual

El primer objetivo a considerar en la optimización multiobjetivo está orientado a satisfacer la demanda en los nodos lo más pronto posible. Por ello, el valor que será minimizado es la suma total de la demanda en todos los nodos y períodos. De esta manera, una posible solución que satisfaga la demanda de todos los nodos en 3 días será preferible a una solución que solo satisfaga la demanda de la mitad de los nodos en 6 días. La siguiente expresión indica la suma de la demanda de cada nodo k en todos los períodos t:

$$\sum_{k,t \geq 1} Demanda_{k,t}$$

Para considerar la demanda de diferentes tipos de bienes, así como la prioridad de determinados tipos de bienes, esta expresión se puede expandir a:

$$\sum_{k,i,t \geq 1} Prioridad_i * Demanda_{k,i,t}$$

##### 4.2.4.2 Minimizar costo de transporte

Así como se desea atender la demanda lo más pronto posible, también se debe considerar realizarlo de la manera más eficiente posible, para no desperdiciar recursos que podrían ser utilizados para atender a más nodos. Debido a que el costo de transporte tiende a ser proporcional a la distancia recorrida, el valor total del segundo

objetivo de optimización será la suma del costo unitario de transporte (por unidad de distancia) multiplicado por la distancia recorrida de todos los vehículos. 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 vehículos 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 vehículos que se desplacen desde  $j$  hacia  $k$  en el período  $t$ :

$$\sum_{j,k,t} \text{CostoTransp} * \text{Distancia}_{j,k} * \text{Veh}_{j,k,t}$$

Para considerar diferentes tipos de vehículos, esta expresión se puede expandir a:

$$\sum_{j,k,v,t} \text{CostoTransp}_v * \text{Distancia}_{j,k} * \text{Veh}_{j,k,v,t}$$

#### 4.2.5 Restricciones

##### 4.2.5.1 Evitar enviar más bienes de los que se dispone

Como en todo problema de transporte, es necesario asegurar que solo se puedan transportar bienes que se encuentren en el inventario. Esto queda descrito por la siguiente restricción, que indica que, en cualquier período  $t$ , la suma de la cantidad bienes que sean transportados desde el nodo  $j$  hacia los demás nodos adyacentes  $k$  (bienes que salen desde  $j$ ) no debe ser superior al inventario total en el nodo  $j$ :

$$\sum_k \text{Transp}_{j,k,t} \leq \text{Inv}_{j,t} \quad , \text{para todo nodo } j \text{ y período } t$$

Para considerar diferentes tipos de bienes y vehículos, esta restricción se puede expandir a:

$$\sum_{k,v} \text{Transp}_{j,k,v,i,t} \leq \text{Inv}_{j,i,t} \quad , \text{para todo nodo } j, \text{ bien } i \text{ y período } t$$

##### 4.2.5.2 Evitar usar más vehículos de los que se dispone

De manera análoga, se debe garantizar que no se intente usar más vehículos de los que se dispone para transportar los bienes. Esto queda descrito por la siguiente restricción, que indica que, en cualquier período  $t$ , la suma de los vehículos que sean movilizadas desde el nodo  $j$  hacia los demás nodos adyacentes  $k$  (vehículos que salen desde  $j$ ) no debe ser superior a la cantidad de vehículos disponibles en el nodo  $j$ :

$$\sum_k Veh_{j,k,t} \leq VehDisponibles_{j,t} \quad , para todo nodo j y período t$$

Para considerar diferentes tipos de vehículos, esta restricción se puede expandir a:

$$\sum_k Veh_{j,k,v,t} \leq VehDisponibles_{j,v,t} \quad , para todo nodo j, vehículo v y período t$$

#### 4.2.5.3 Evitar sobrepasar la capacidad de los vehículos

La cantidad de bienes que serán transportados será limitada también por la capacidad de los vehículos que se tengan disponibles, tanto de volumen, como peso. Esto queda descrito por las siguientes restricciones, que establecen que, en cualquier período t, la cantidad de bienes que se transporten de j a k no debe ser superior a la capacidad total de la cantidad de vehículos que se movilicen de j a k:

$$Transp_{j,k,t} * Volumen \leq Veh_{j,k,t} * CapacidadVehV$$

$$Transp_{j,k,t} * Peso \leq Veh_{j,k,t} * CapacidadVehW$$

, para cada par de nodos j, k y período t

Para considerar diferentes tipos de bienes y vehículos, estas restricciones se pueden expandir a:

$$\sum_i Transp_{j,k,v,i,t} * Volumen_i \leq Veh_{j,k,v,t} * CapacidadVehV_v$$

$$\sum_i Transp_{j,k,v,i,t} * Peso_i \leq Veh_{j,k,v,t} * CapacidadVehW_v$$

, para cada par de nodos j, k, tipo de vehículo v y período t

#### 4.2.5.4 Balance en el inventario

El movimiento de los bienes entre los diferentes nodos actualizará el inventario en cada período, generando reglas de balance de inventario. De esta manera, el inventario al comienzo del período t en el nodo k será equivalente al inventario que había al inicio del período anterior t-1 menos la cantidad de bienes que se consumieron durante dicho período t-1, menos la cantidad de bienes que salieron de la ubicación actual para ser transportados a otros destinos, más los bienes que acaban de llegar de otros lugares (los cuales partieron en t-1). Esta regla queda definida por la siguiente expresión:

$$Inv_{k,t} = Inv_{k,t-1} - Consumo_{k,t-1} - \sum_j Transp_{k,j,t-1} + \sum_j Transp_{j,k,t-1}$$

, para cada nodo  $k$  y período  $t \geq 1$

Para considerar diferentes tipos de vehículos y bienes, esta restricción puede expandirse a:

$$Inv_{k,i,t} = Inv_{k,i,t-1} - Consumo_{k,i,t-1} - \sum_{j,v} Transp_{k,j,v,i,t-1} + \sum_{j,v} Transp_{j,k,v,i,t-1}$$

, para cada nodo  $k$ , bien  $i$  y período  $t \geq 1$

#### 4.2.5.5 Balance en la cantidad de vehículos disponibles

De manera análoga, el movimiento de los bienes entre los diferentes nodos también actualizará la cantidad de vehículos disponibles, generando reglas de balance de vehículos disponibles. De esta manera, la cantidad de vehículos disponibles al comienzo del período  $t$  en el nodo  $k$  será equivalente a la cantidad de vehículos que había al inicio del período anterior  $t-1$  menos la cantidad de bienes que salieron de la ubicación actual para ser movilizarse a otros destinos, más los vehículos que acaban de llegar de otros lugares (los cuales partieron en  $t-1$ ). Esta regla queda definida por la siguiente expresión:

$$VehDisponibles_{k,t} = VehDisponibles_{k,t-1} - \sum_j Veh_{k,j,t-1} + \sum_j Veh_{j,k,t-1}$$

, para cada nodo  $k$  y período  $t \geq 1$

Para considerar diferentes tipos de vehículos, esta restricción puede expandirse a:

$$VehDisponibles_{k,t} = VehDisponibles_{k,t-1} - \sum_{j,v} Veh_{k,j,v,t-1} + \sum_{j,v} Veh_{j,k,v,t-1}$$

, para cada nodo  $k$  y período  $t \geq 1$

#### 4.2.5.6 Determinación del consumo

La decisión de cuántos bienes se consumen en un nodo para satisfacer su demanda está limitada por la demanda existente, así como la cantidad de bienes disponibles en ese momento. Esto queda representado por la siguiente restricción:

$$Consumo_{k,t} = \min(Demanda_{k,t}, Inv_{k,t}), \text{ para cada nodo } k \text{ y período } t$$

Para considerar diferentes tipos de productos, la restricción puede expandirse a



$$Consumo_{k,i,t} = \min(Demanda_{k,i,t}, Inv_{k,i,t}), \text{ para cada nodo } k, \text{ bien } i \text{ y período } t$$

#### 4.2.5.7 Actualización de la demanda

El valor del consumo en cada período traerá como consecuencia la reducción de la demanda, para lo cual se tienen las siguientes reglas:

$$Demanda_{k,t} = Demanda_{k,t-1} - Consumo_{k,t-1}$$

*, para cada nodo k y período  $t \geq 1$*

Para considerar diferentes tipos de bienes, esta restricción puede expandirse a:

$$Demanda_{k,i,t} = Demanda_{k,i,t-1} - Consumo_{k,i,t-1}$$

*, para cada nodo k, bien i y período  $t \geq 1$*

#### 4.2.5.8 Condiciones iniciales del problema

Para el período inicial, las variables ***Demanda***, ***Inventario*** y ***VehDisponibles*** se inicializarán con los parámetros ***Demanda0***, ***Inventario0*** y ***VehDisponibles0***, respectivamente:

$$Demanda_{k,0} = Demanda0_k$$

$$Inv_{k,0} = Inv0_k$$

$$VehDisponibles_{k,0} = VehDisponibles0_k$$

*, para cada nodo k*

Para considerar diferentes tipos de bienes, estas restricciones pueden expandirse a:

$$Demanda_{k,i,0} = Demanda0_{k,i}$$

$$Inv_{k,i,0} = Inv0_{k,i}$$

$$VehDisponibles_{k,i,0} = VehDisponibles0_{k,i}$$

*, para cada nodo k y bien i*

#### 4.2.6 Rango de existencia

Dado que todas las variables y parámetros a excepción de los parámetros referentes a volumen y peso 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 volumen y peso es el conjunto de los números reales no negativos:  $R_{\geq 0}$ .

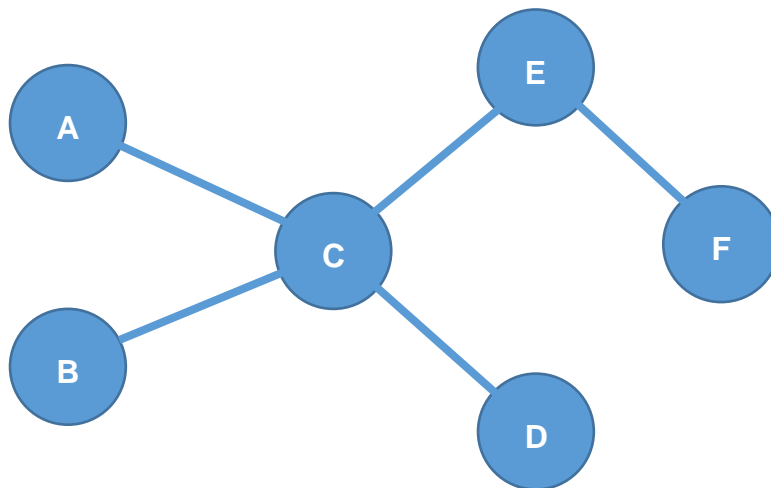
### 4.3 Relación con el objetivo

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 lineales. Para cada uno de los algoritmos, se requerirá adaptar esta formulación según corresponda.

### 4.4 Validación de la construcción

Con el fin de ejemplificar y realizar una validación de la formulación previamente planteada, se propone el siguiente ejemplo.

Sea la siguiente red de distribución con  $K=6$  nodos, tal que la distancia entre cada par de nodos es igual a 5 unidades ( $Distancia_{jk} = 5$ ):



Sea un horizonte de tiempo de 4 días, donde cada día corresponde a un período ( $T=4$ ). La información de inventario y demanda inicial ( $t=0$ ), así como la disponibilidad de vehículos inicial es la siguiente:

Nodo ( $k, t=0$ )	Inventario ( $Inv_{k0}$ )	Demanda ( $Demanda_{k0}$ )	Vehículos Disponibles ( $VehDisp_{k0}$ )
A	0	0	3
B	40	0	5

C	30	0	0
D	0	20	0
E	0	30	0
F	0	10	0

Sea la capacidad de volumen de los vehículos de 20 unidades (**CapacidadVehV** = 20) y la capacidad de peso de los vehículos de 20 unidades (**CapacidadVehW** = 20). Asimismo, sea el costo de transporte de los vehículos igual a 5 unidades por unidad de distancia (**CostoTransp** = 5). Finalmente, para este ejemplo, sea el peso y volumen de los bienes a distribuir igual a 2 unidades (**Volumen** = 2, **Peso** = 2).

Obsérvese que en este problema será necesario que se envíen vehículos vacíos desde A hacia C, con el fin de recoger y movilizar los bienes de C a los puntos de demanda (D, E, F).

Después de ejecutar algún algoritmo de optimización, obtenemos los siguientes valores para las variables de decisión:

- En  $t=0$ , enviar 3 vehículos conteniendo 10 bienes cada uno desde B a C, y enviar 3 vehículos vacíos desde A a C.

<b>Veh<sub>jk0</sub> / Transp<sub>jk0</sub></b>	<b>A</b>	<b>B</b>	<b>C</b>
<b>A</b>			3 / 0
<b>B</b>			3 / 30

Esta decisión actualiza las variables auxiliares Inventario, Consumo y Vehículos Disponibles en el siguiente período ( $t=1$ ), según las reglas de balance de inventario. De esta manera, en el ejemplo a continuación **Inv<sub>B,1</sub>** representa la cantidad de bienes que hay en el nodo B al comienzo del período 1, **Inv<sub>B,0</sub>** representa la cantidad de bienes que había en el nodo B al comienzo del período 0, **Consumo<sub>B,0</sub>** representa la cantidad de bienes que se consumen en el nodo B

en el período 0 (en otras palabras la necesidad de ayuda humanitaria atendida en dicha ubicación y momento) y **Transp<sub>B,j</sub>** representa la cantidad de bienes que se transportan desde B hacia algún nodo j en el período 0:

$$Inv_{B,1} = Inv_{B,0} - Consumo_{B,0} - \sum_j Transp_{B,j,0} + \sum_j Transp_{j,B,0}$$

$$= 40 - 0 - 30 + 0 = 10$$

$$Inv_{C,1} = Inv_{C,0} - Consumo_{C,0} - \sum_j Transp_{C,j,0} + \sum_j Transp_{j,C,0}$$

$$= 30 - 0 - 0 + 30 = 60$$

En este par de ejemplos se puede ver que el inventario en B se reduce de 40 unidades a 10 unidades debido a que en el período anterior salieron 30 unidades desde B hacia C. De la misma manera, el inventario en C aumenta de 30 a 60 debido a que en el período anterior se movilizaron 30 unidades hacia C desde B.

Nodo (k, t=1)	Inventario (Inv <sub>k1</sub> )	Demanda (Demanda <sub>k1</sub> )	Consumo (Consumo <sub>k1</sub> )	Vehículos Disponibles (VehDisp <sub>k1</sub> )
A	0	0	0	0
B	10	0	0	2
C	60	0	0	6
D	0	20	0	0
E	0	30	0	0
F	0	10	0	0

- En t=1, enviar 2 vehículos conteniendo 10 bienes cada uno desde C a D, y enviar 4 vehículos conteniendo desde C a E.

Veh <sub>jk1</sub> / Transp <sub>jk1</sub>	D	E	F

<b>C</b>	2 / 20	4 / 40	
----------	--------	--------	--

De manera similar, esto actualiza las variables auxiliares:

<b>Nodo (k, t=2)</b>	<b>Inventario (Inv<sub>k2</sub>)</b>	<b>Demanda (Demanda<sub>k2</sub>)</b>	<b>Consumo (Consumo<sub>k2</sub>)</b>	<b>Vehículos Disponibles (VehDisp<sub>k2</sub>)</b>
A	0	0	0	0
B	10	0	0	2
C	0	0	0	0
D	20	20	20	2
E	40	30	30	4
F	0	10	0	0

- Al llegar los vehículos a C y D, se satisface la demanda de estos puntos y quedan sobrando 10 unidades en E, las cuales serán destinadas al nodo F. Finalmente, en t=2, enviar 1 vehículo conteniendo 10 bienes desde E a F.

<b>Veh<sub>jk2</sub> / Transp<sub>jk2</sub></b>	<b>D</b>	<b>E</b>	<b>F</b>
<b>E</b>			1 / 10

<b>Nodo (k, t=3)</b>	<b>Inventario (Inv<sub>k3</sub>)</b>	<b>Demanda (Demanda<sub>k3</sub>)</b>	<b>Consumo (Consumo<sub>k3</sub>)</b>	<b>Vehículos Disponibles (VehDisp<sub>k3</sub>)</b>
A	0	0	0	0
B	10	0	0	2

C	0	0	0	0
D	0	0	0	2
E	0	0	0	3
F	10	10	10	1

Finalmente, una vez que los 10 últimos bienes llegan a F y se consumen, se observa que en  $t=4$  ya no existe demanda pendiente de atención:

<b>Nodo (k, t=4)</b>	<b>Inventario (Inv<sub>k4</sub>)</b>	<b>Demanda (Demanda<sub>k4</sub>)</b>	<b>Consumo (Consumo<sub>k4</sub>)</b>	<b>Vehículos Disponibles (VehDisp<sub>k4</sub>)</b>
A	0	0	0	0
B	10	0	0	2
C	0	0	0	0
D	0	0	0	2
E	0	0	0	3
F	0	0	0	1

El valor de las funciones objetivo para esta solución es la siguiente:

$$\sum_{k,t \geq 1} Demanda_{k,t} = (20 + 30 + 10) + (20 + 30 + 10) + (10) + (0) = 130$$

$$\sum_{j,k,t} CostoTransp * Distancia_{j,k} * Veh_{j,k,t} = (5 * 5 * 6) + (5 * 5 * 6) + (5 * 5 * 1) = 325$$

En la primera función objetivo se suma el total de la demanda en todos los nodos por cada período, mientras que en la segunda función objetivo se acumulan los costos de transporte por el desplazamiento de todos los vehículos en cada período.

## 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
  - **V**: número de tipos de vehículos
  - **I**: número de tipos de bienes (suministros) de ayuda humanitaria
  - **T**: número de períodos del horizonte de planificación
  - **Distancia<sub>jk</sub>**: distancia entre los nodos j y k
  - **Demanda0<sub>ki</sub>**: demanda inicial de bienes de tipo i en el nodo k
  - **Inv0<sub>ki</sub>**: inventario inicial de bienes de tipo i en el nodo k
  - **CapacidadVehV<sub>v</sub>**: capacidad (volumen) de vehículos de tipo v
  - **CapacidadVehW<sub>v</sub>**: capacidad (peso) de vehículos de tipo v
  - **CostoTransp<sub>v</sub>**: costo de transporte de vehículos de tipo v
  - **VehDisponibles0<sub>kv</sub>**: cantidad inicial de vehículos de tipo v en el nodo k
  - **Volumen<sub>i</sub>**: volumen de los bienes de tipo i
  - **Peso<sub>i</sub>**: peso de los bienes de tipo i
  - **Prioridad<sub>i</sub>**: prioridad de distribución de los bienes de tipo i
- Variables auxiliares
  - **Inv<sub>kit</sub>**: inventario de bienes de tipo i en el nodo k y período t
  - **Demanda<sub>kit</sub>**: demanda de bienes de tipo i en el nodo k y período t

- **Consumo<sub>kit</sub>**: consumo de bienes de tipo i en el nodo k y período t
- **VehDisponibles<sub>kvt</sub>**: cantidad de vehículos de tipo v en el nodo k y período t

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

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

Los parámetros de capacidad y costo de transporte serán agrupados en la clase vehículo:

```
Clase Vehículo {
    Entero idTipoVehiculo; // número correlativo
    Real CapacidadVehV;
    Real CapacidadVehW;
    Real CostoTransp;
}
```

Los parámetros de volumen y peso serán agrupados en la clase bien:

```
Clase Bien {
    Entero idTipoBien;
    Real Volumen;
    Real Peso;
    Entero Prioridad;
}
```

Ahora, las variables que varíen en el tiempo (**Demanda**, **Inv**, **VehDisponibles**, **Consumo**) serán agrupados en la clase período y se inicializarán con los parámetros correspondientes (**Demanda0**, **Inv0**, **VehDisponibles0**) en el período 0:

```
Clase Período {
    Entero idPeríodo;
    Arreglo de enteros [K, I] Demanda;
    Arreglo de enteros [K, I] Inv;
    Arreglo de enteros [K, I] Consumo;
    Arreglo de enteros [K, V] VehDisponibles;
}
```



Finalmente, para agrupar todas las estructuras de datos de los parámetros y variables auxiliares, se ha definido la siguiente clase Problema:

```
Clase Problema {
    Entero K;
    Entero V;
    Entero I;
    Entero T;
    Arreglo de reales [K, K] Distancia;
    Lista<Vehículo> Vehículos;
    Lista<Bien> Bienes;
    Lista<Período> Períodos;
}
```

### 5.2.2 Variables de decisión y cromosoma

El resumen de las variables de decisión que se formularon en el capítulo anterior es el siguiente:

- **$Veh_{jkt}$** , que indica la cantidad de vehículos de tipo *v* que se movilizarán entre cada par de nodos *j* y *k* en cada período *t*.
- **$Transp_{jkvit}$** , que indica la cantidad de bienes (suministros de ayuda humanitaria) de tipo *i* que se transportan en vehículos de tipo *v* desde el nodo *j* hacia *k* en el período *t*.

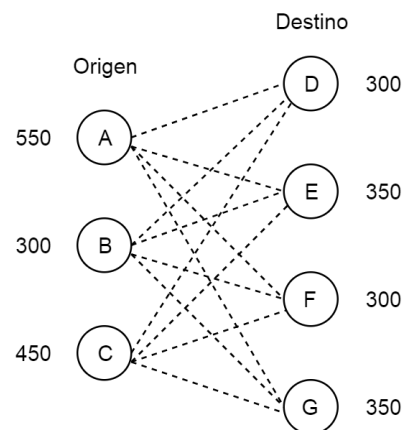
Estas variables de decisión son las que generan un plan de distribución y describen una posible solución para el problema de optimización. De acuerdo con primeras investigaciones en el tema de algoritmos genéticos aplicados al problema de transporte (Vignaux & Michalewicz, 1991), una manera inicial de definir el cromosoma es hacer que el cromosoma represente directamente dichas variables 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  $j$ :

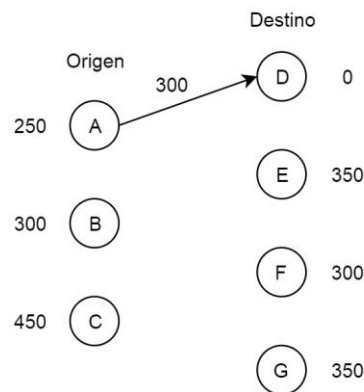
	D	E	F	G
A	11	19	17	18
B	16	14	18	15
C	15	16	19	13



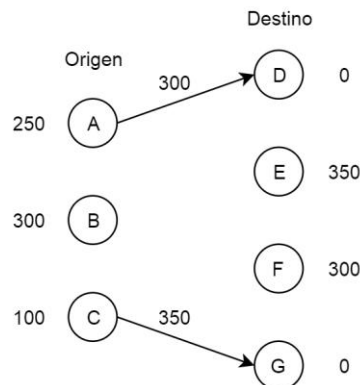
Un posible cromosoma para este problema sería el siguiente:

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

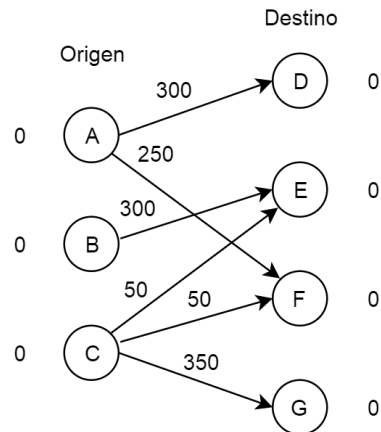
En este cromosoma, el valor de prioridad más alto (7) ocurre en el nodo destino D, lo cual significa que se debe atender primero, para lo cual se selecciona el nodo origen que tenga el menor costo para llegar a D. Observando la matriz de costos, el nodo origen A tiene el menor costo (11), por lo cual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

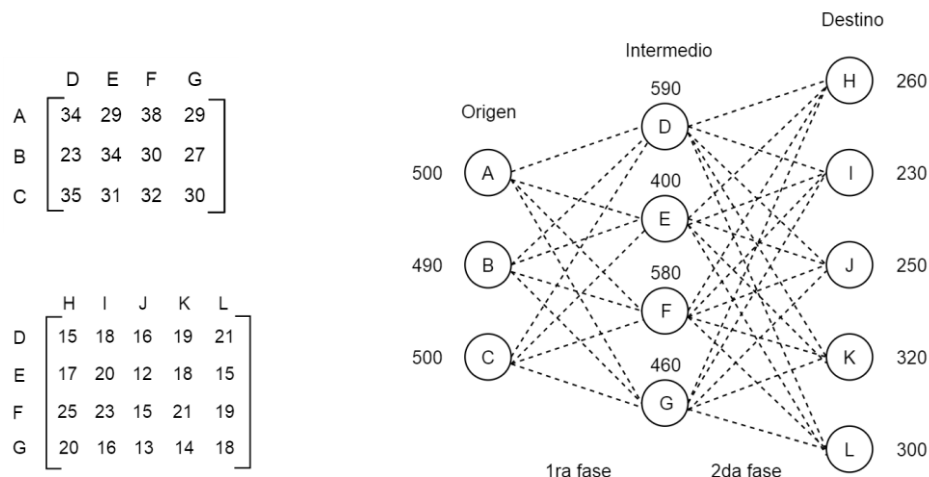
1. **DecodificarCromosoma (*cromosoma*)**
2. Para cada nodo en el cromosoma hacer:
3.     *nodo* = escoger el siguiente nodo de mayor prioridad (*cromosoma*)
4.     Si *nodo* es un nodo de destino entonces:
5.         *origen* = seleccionar el nodo origen de menor costo
6.         *bienesAtransportar* =  $\min(\text{oferta}_{\text{origen}}, \text{demanda}_{\text{nodo}})$
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 las restricciones de capacidad y disponibilidad de vehículos definidas previamente en la formulación, basta someter la variable *bienesAtransportar* a la cantidad de vehículos disponibles en el nodo de origen. De esta manera, el límite

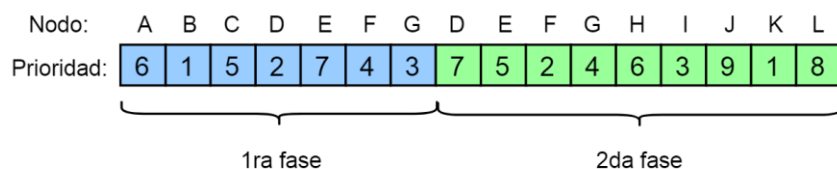
superior de la variable *bienesAtransportar* será la capacidad máxima (tanto en peso como en volumen) de los vehículos disponibles en el nodo origen.

Para considerar la diferente prioridad de los bienes de ayuda humanitaria, la estrategia es ejecutar el algoritmo *DecodificarCromosoma* con los parámetros de oferta y demanda de los bienes de ayuda humanitaria de primera prioridad. Si después de ejecutar el algoritmo sobran vehículos disponibles y espacio en los vehículos parcialmente cargados, volver a ejecutar el algoritmo *DecodificarCromosoma* con los parámetros de oferta y demanda de los bienes de ayuda de segunda prioridad y así sucesivamente.

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

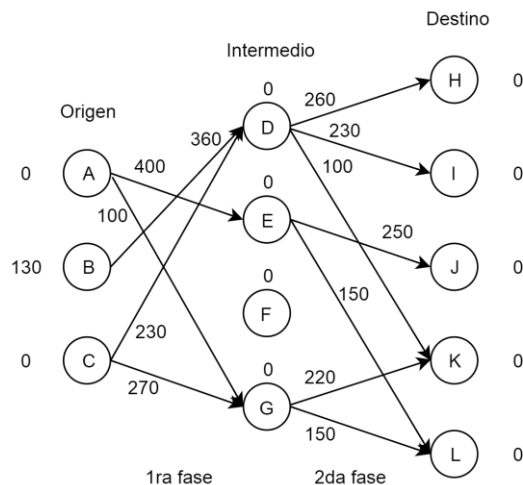


De la misma manera en que existen dos matrices para representar cada fase de distribución, el cromosoma puede ser extendido a:



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 el aspecto multiperíodo de la formulación del problema. Para este ejemplo, a partir del cromosoma mostrado, la aplicación del algoritmo de decodificación daría como resultado la siguiente asignación:



Con esta extensión, el cromosoma queda definido como un arreglo de tamaño  $O + 2 * M + D$ , donde  $O$  es el número de nodos de origen,  $M$  es el número de nodos intermedios,  $D$  es el número de nodos de destino, y  $N = O + 2 * M + D$ , donde  $N$  es la cantidad total de nodos en la red de distribución. El contenido del cromosoma serán números enteros que representen el nivel de prioridad de cada nodo:

```
Clase Cromosoma {
    Arreglo de enteros  $[O + 2 * M + D]$  PrioridadTransporte;
}
```

Se puede observar que el tamaño del cromosoma no será superior a  $2 * N$ , 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, D: tiempo límite, Iter: número de iteraciones)**
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 el tiempo **D** se agote o se llegue al límite de **Iter** iteraciones
15. Retornar  $\text{mejor\_solución}$

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 de demanda (o necesidad de ayuda humanitaria) (F1) y minimización de costo de transporte (F2):

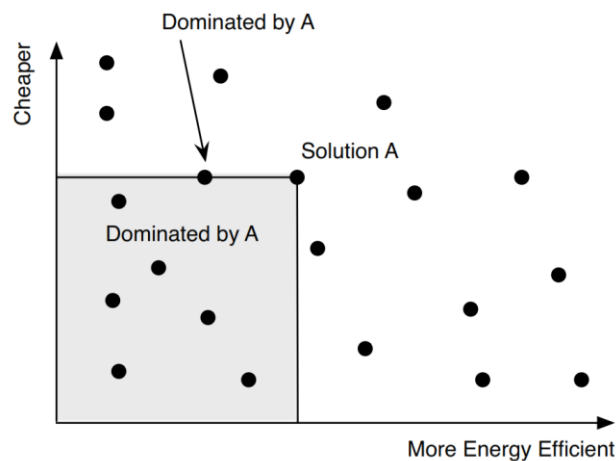
$$\sum_{k,i,t \geq 1} Prioridad_i * Demanda_{k,i,t}$$
$$\sum_{j,k,v,t} CostoTransp_v * Distancia_{j,k} * Veh_{j,k,v,t}$$



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 y en qué vehículos fueron transportados, lo cual permite obtener el valor de F2. De la misma manera, aplicando las actualizaciones de inventario y satisfacción de necesidad de ayuda definidas en el capítulo de formulación, se obtiene información de cuánta demanda fue atendida y cuándo fue atendida, lo cual permite obtener el valor de F1. Ambos valores representan la aptitud del cromosoma, por lo cual se representan como una tupla (F1, F2). En este caso los objetivos están parcialmente enfrentados, pues para tener un costo de transporte de 0 no habría movimiento de vehículos, lo cual significaría que la demanda no sería cubierta en lo absoluto. De similar manera, en casos en los que haya excesiva demanda, el costo para trasladar todos los bienes podría superar el presupuesto con el que se cuenta para movilizarlos. 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**:

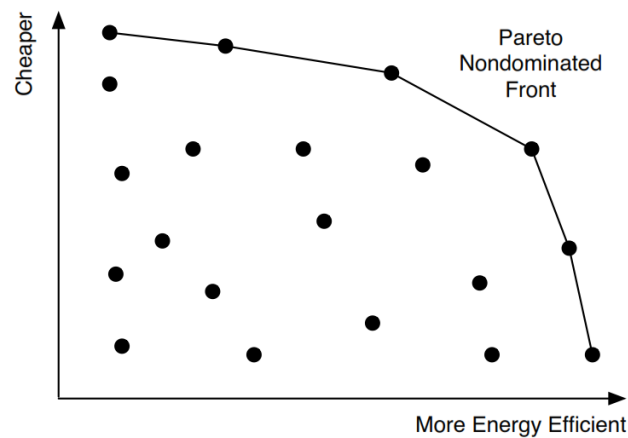


Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015).

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.

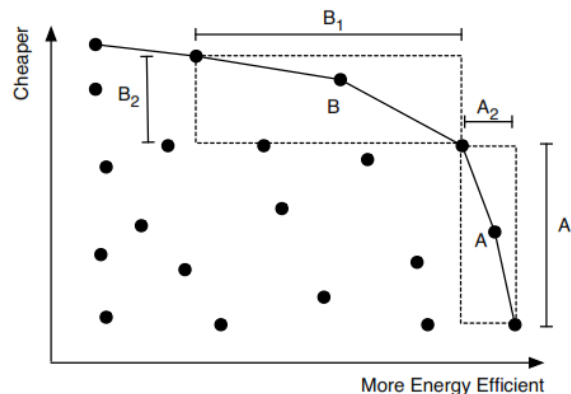
## 6.5 Selección

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).

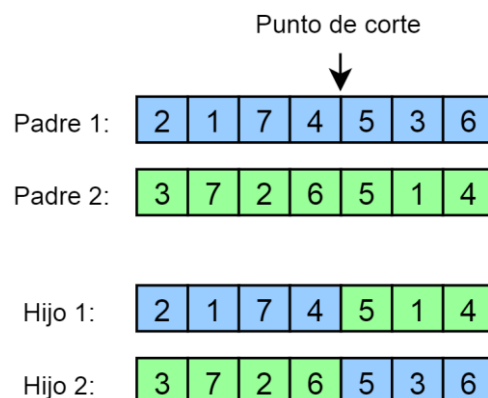
El algoritmo de selección por torneo es el siguiente:

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

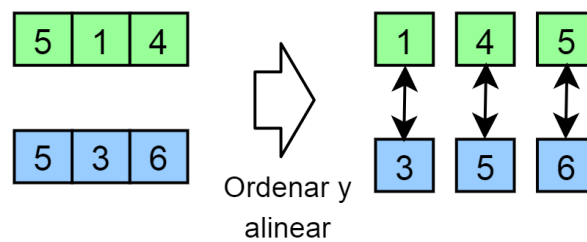
Para evaluar la aptitud se utilizará el concepto de aptitud por dominancia definido en la sección anterior.

## 6.6 Cruce

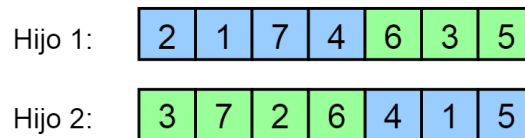
El operador de cruce permite juntar la información de dos individuos, preferiblemente los más aptos, con el fin de crear individuos que contengan las mejores características de ambos padres. Dado que el cromosoma representa permutaciones, es posible aplicar operadores de cruce específicos para permutaciones. 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 un remapeo 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:



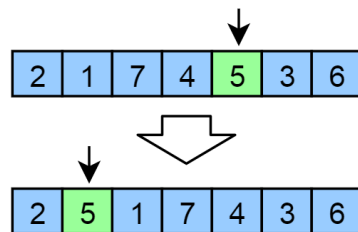
Esto queda reflejado en el siguiente algoritmo de cruce:

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

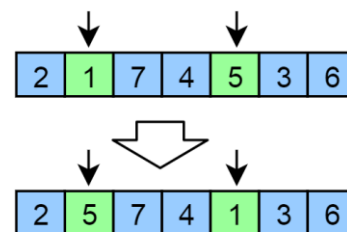
Para considerar múltiples períodos de distribución, el cromosoma se segmenta por cada período o fase según lo descrito en el capítulo anterior, y el cruce se realiza por cada período separado, debido a que cada fase representa una permutación independiente.

## 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. Debido a que el cromosoma definido 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.



Mutación por inserción



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.

## 6.8 Criterio de parada

Existirán dos criterios de parada. El primero será el número de iteraciones del algoritmo genético, mientras que el segundo será un tiempo límite. Si alguna de las condiciones se cumple, esto es, si se acaban todas las iteraciones del algoritmo genético o si se llega al tiempo límite, se detendrá la ejecución del algoritmo y se retornarán las mejores soluciones.

## **Capítulo 7. Experimentación Numérica**

### **7.1 Introducción**

Una vez obtenidos los resultados del algoritmo genético y la optimización del programa lineal entero mixto mediante el solver GLPK, es momento de comparar dichos resultados para validar las hipótesis de investigación del presente proyecto de fin de carrera.

### **7.2 Configuración de experimentación**

Se ejecutará cada método con 3 conjuntos de datos diferentes. El primero será un conjunto de datos pequeño generado aleatoriamente, el segundo será un conjunto de datos que represente un escenario real, a partir del análisis del caso de heladas y friaje en el Perú en el año 2017, y el tercero será un conjunto de datos grande generado aleatoriamente, para las pruebas de estrés de los algoritmos.

Como la optimización es multiobjetivo, en el caso del algoritmo genético, se obtendrá como resultado el frente de Pareto conteniendo un conjunto diverso de soluciones no dominadas. Para poder compararlo con el programa lineal, dado que este solo cuenta con un único objetivo combinando los dos objetivos principales, el programa lineal se ejecutará 3 veces variando el peso de cada objetivo: en el primer caso se dará mayor peso al primer objetivo, en el segundo caso se dará mayor peso al segundo objetivo y en el tercer caso, se dará el mismo peso a ambos objetivos. Una vez obtenidos estos resultados, se harán pruebas de hipótesis por cada objetivo por separado.

Finalmente, se realizarán 20 corridas de cada configuración. A partir de estas corridas se evaluará el tiempo de ejecución y la optimalidad de las soluciones. Para todas las pruebas se asumirá un p-valor de 0.05 para rechazar las hipótesis nulas.

#### **7.2.1 Tiempo de ejecución**

Las hipótesis estadísticas para la comparación del tiempo de ejecución son las siguientes:

- H0: el algoritmo genético tiene un tiempo de ejecución mayor o igual al programa lineal entero mixto
- H1: el algoritmo genético tiene un menor tiempo de ejecución que el programa lineal entero mixto

El diseño experimental es a tres bloques, debido a que se está probando con tres conjuntos diferentes de datos. El procedimiento para validar las hipótesis es el siguiente:

#### **7.2.1.1 Verificar normalidad**

Primero se aplicará una prueba de normalidad para verificar que los tiempos de ejecución siguen una distribución normal. Para ello se utilizará la prueba de Shapiro-Wilk bajo las siguientes hipótesis:

- H0: los tiempos de ejecución siguen una distribución normal
- H1: los tiempos de ejecución no siguen una distribución normal

Si el p-valor de la prueba es mayor o igual a 0.05, no existe evidencia en los datos para rechazar H0. Esta prueba se realizará por cada tipo de algoritmo.

#### **7.2.1.2 Verificar igualdad de varianzas**

Una vez probada la normalidad de ambas distribuciones, se procederá a verificar si las varianzas de los tiempos de ejecución del algoritmo genético y el programa lineal son iguales o diferentes. Para ello se utilizará la prueba de F de Fisher bajo las siguientes hipótesis:

- H0: las varianzas de los tiempos de ejecución son iguales
- H1: las varianzas de los tiempos de ejecución son diferentes

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que las varianzas son diferentes. Dependiendo de los resultados de esta prueba se utilizarán variantes ligeramente diferentes de la siguiente prueba para comparación de medias.

#### **7.2.1.3 Verificar igualdad de medias**

Una vez verificada si las varianzas son iguales o diferentes, se procederá a verificar si las medias de los tiempos de ejecución del algoritmo genético y el programa lineal son iguales o diferentes. Debido a que la cantidad de corridas es igual a 20, se utilizará la prueba de t de Student para diferencia de medias con varianzas iguales o diferentes, dependiendo del resultado de la prueba anterior, a dos colas, bajo las siguientes hipótesis:

- H0: las medias de los tiempos de ejecución son iguales
- H1: las medias de los tiempos de ejecución son diferentes

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que las medias son diferentes.



#### **7.2.1.4 Verificar comparación de medias**

Una vez verificada si las medias son iguales o diferentes, se procederá a verificar si la media del tiempo de ejecución del algoritmo genético es menor al programa lineal. De manera similar a la prueba anterior, se utilizará la prueba de t de Student para diferencia de medias, esta vez a una cola, bajo las siguientes hipótesis:

- H0: la media del tiempo de ejecución del algoritmo genético es mayor al del programa lineal
- H1: la media del tiempo de ejecución del algoritmo genético es menor al del programa lineal

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que la media del tiempo de ejecución del algoritmo genético es menor al del programa lineal. Con estos últimos resultados, validaríamos la hipótesis estadística inicial.

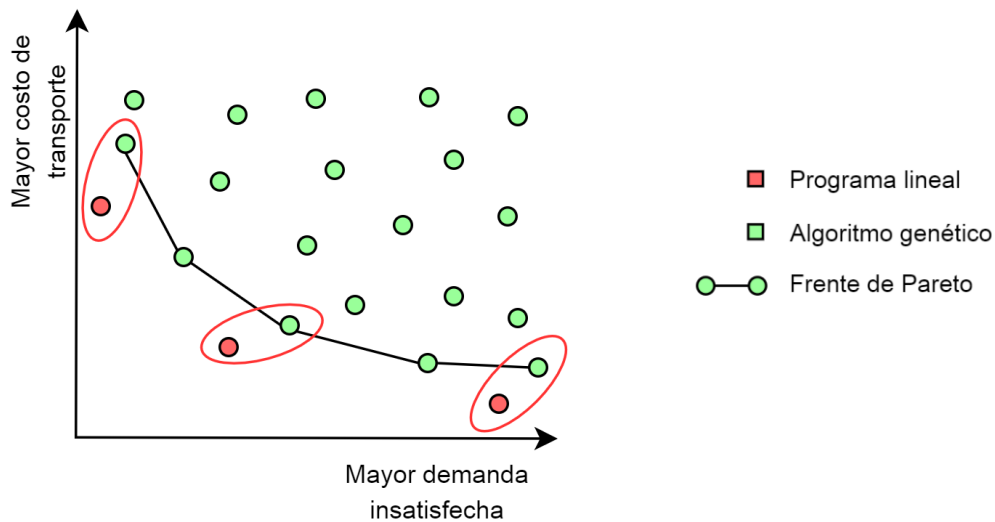
#### **7.2.2 Optimalidad**

Las hipótesis estadísticas para la comparación de los valores objetivo son las siguientes:

- H0: el algoritmo genético tiene un valor en el objetivo  $i$  menor o igual al programa lineal entero mixto
- H1: el algoritmo genético tiene un valor en el objetivo  $i$  mayor que el programa lineal entero mixto

Donde el objetivo  $i$  representa el costo de transporte y la demanda insatisfecha (uno a la vez). Estas pruebas se realizarán por cada uno de los dos objetivos por separado. El diseño experimental es a tres bloques, debido a que se está probando con tres conjuntos diferentes de datos.

Un punto importante es que el algoritmo genético dará como resultado el frente de Pareto, el cual es un conjunto de soluciones no dominadas bajo diferentes compromisos en ambos objetivos. Para comparar este resultado con los 3 resultados del programa lineal (con diferentes pesos para cada objetivo), cada uno de los 3 resultados del programa lineal se comparará con la solución del frente de Pareto más cercana en el espacio de funciones objetivos, como se describe en la siguiente imagen:



Con este procedimiento ya es posible construir un conjunto de resultados sobre los cuales se puedan aplicar las pruebas de hipótesis de manera similar al tiempo de ejecución:

#### 7.2.2.1 Verificar normalidad

Primero se aplicará una prueba de normalidad para verificar que los valores de cada objetivo siguen una distribución normal. Para ello se utilizará la prueba de Shapiro-Wilk bajo las siguientes hipótesis:

- H0: los valores del objetivo  $i$  siguen una distribución normal
- H1: los valores del objetivo  $i$  no siguen una distribución normal

Donde el objetivo  $i$  representa el costo de transporte y la demanda insatisfecha (uno a la vez). Si el p-valor de la prueba es mayor o igual a 0.05, no existe evidencia en los datos para rechazar H0. Esta prueba se realizará por cada tipo de algoritmo.

#### 7.2.2.2 Verificar igualdad de varianzas

Una vez probada la normalidad de ambas distribuciones, se procederá a verificar si las varianzas de los valores objetivo del algoritmo genético y el programa lineal son iguales o diferentes. Para ello se utilizará la prueba de F de Fisher bajo las siguientes hipótesis:

- H0: las varianzas de los valores del objetivo  $i$  son iguales
- H1: las varianzas de los valores del objetivo  $i$  son diferentes

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que las varianzas son diferentes. Dependiendo de los

resultados de esta prueba se utilizarán variantes ligeramente diferentes de la siguiente prueba para comparación de medias.

#### **7.2.2.3 Verificar igualdad de medias**

Una vez verificada si las varianzas son iguales o diferentes, se procederá a verificar si las medias de los valores objetivo del algoritmo genético y el programa lineal son iguales o diferentes. Debido a que la cantidad de corridas es igual a 20, se utilizará la prueba de t de Student para diferencia de medias con varianzas iguales o diferentes, dependiendo del resultado de la prueba anterior, a dos colas, bajo las siguientes hipótesis:

- H0: las medias de los valores del objetivo  $i$  son iguales
- H1: las medias de los valores del objetivo  $i$  son diferentes

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que las medias son diferentes.

#### **7.2.2.4 Verificar comparación de medias**

Una vez verificada si las medias son iguales o diferentes, se procederá a verificar si la media de los valores objetivo del algoritmo genético es menor al programa lineal. De manera similar a la prueba anterior, se utilizará la prueba de t de Student para diferencia de medias, esta vez a una cola, bajo las siguientes hipótesis:

- H0: la media del valor del objetivo  $i$  del algoritmo genético es mayor al del programa lineal
- H1: la media del valor del objetivo  $i$  del algoritmo genético es menor al del programa lineal

Si el p-valor de la prueba es menor a 0.05, existe evidencia en los datos para rechazar H0, por tanto se aceptaría que la media del valor del objetivo del algoritmo genético es menor al del programa lineal. Con estos últimos resultados, validaríamos la hipótesis estadística inicial.

## **Capítulo 8. Conclusiones y trabajos futuros**

### **8.1 Conclusiones**

### **8.2 Trabajos futuros**

## 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>
- Foundation, P. S. (2018). Welcome to Python.org. Retrieved April 14, 2018, from <https://www.python.org/>
- Gen, M., Altiparmak, 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>
- GLPK. (2012). GLPK - GNU Project - Free Software Foundation (FSF). Retrieved April 14, 2018, from <https://www.gnu.org/software/glpk/>
- Google. (2018). Google Optimization Tools | Optimization | Google Developers. Retrieved April 14, 2018, from <https://developers.google.com/optimization/>
- 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. (2016). Microsoft Excel 2016, Spreadsheet Software, Excel Free Trial.

- Retrieved April 14, 2018, from <https://products.office.com/en/excel>
- 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](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](https://doi.org/10.2991/978-94-91216-74-9_2)
- Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization : algorithms and complexity*. Prentice Hall.
- 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](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>
- 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. Teresa, 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)

## **Anexos**

Los anexos deben ser referenciados desde el documento. Por ejemplo, debe existir un párrafo donde se diga que determinada información puede ser vista en el Anexo X.

Los anexos pueden numerarse con letras o número de acuerdo a su preferencia.