**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, Mayo de 2018

Resumen

Tema FCI

# Tabla de Contenido

[Resumen 2](#_Toc511607947)

[Tema FCI 3](#_Toc511607948)

[Tabla de Contenido 4](#_Toc511607949)

[Índice de Figuras **Error! Bookmark not defined.**](#_Toc511607950)

[Índice de Tablas **Error! Bookmark not defined.**](#_Toc511607951)

[Capítulo 1. Generalidades 11](#_Toc511607952)

[1.1 Problemática 11](#_Toc511607953)

[1.2 Objetivos 14](#_Toc511607954)

[1.2.1 Objetivo general 14](#_Toc511607955)

[1.2.2 Objetivos específicos 14](#_Toc511607956)

[1.2.3 Resultados esperados 15](#_Toc511607957)

[1.2.4 Mapeo de objetivos, resultados y verificación 16](#_Toc511607958)

[1.3 Herramientas y Métodos 19](#_Toc511607959)

[1.3.1 Python 19](#_Toc511607960)

[1.3.2 Jupyter Lab 19](#_Toc511607961)

[1.3.3 Visual Studio Code 19](#_Toc511607962)

[1.3.4 AMPL 20](#_Toc511607963)

[1.3.5 AMPL IDE 20](#_Toc511607964)

[1.3.6 GNU Linear Programming Kit 20](#_Toc511607965)

[1.3.7 Microsoft Excel 20](#_Toc511607966)

[1.3.8 Kanban 21](#_Toc511607967)

[1.3.9 Prueba F de Fisher 21](#_Toc511607968)

[1.3.10 Prueba t de Student 21](#_Toc511607969)

[1.3.11 Prueba Kolmogorov-Smirnov 21](#_Toc511607970)

[1.3.12 Prueba de Shapiro-Wilk 22](#_Toc511607971)

[1.4 Viabilidad 22](#_Toc511607972)

[1.4.1 Viabilidad Técnica 22](#_Toc511607973)

[1.4.2 Viabilidad Temporal 22](#_Toc511607974)

[1.4.3 Viabilidad Económica 23](#_Toc511607975)

[1.4.4 Conclusión 23](#_Toc511607976)

[1.5 Alcance, Limitaciones y Riesgos 24](#_Toc511607977)

[1.5.1 Alcance 24](#_Toc511607978)

[1.5.2 Limitaciones 24](#_Toc511607979)

[1.5.3 Riesgos 25](#_Toc511607980)

[Capítulo 2. Marco Conceptual 26](#_Toc511607981)

[2.1 Introducción 26](#_Toc511607982)

[2.2 Peligros y desastres 26](#_Toc511607983)

[2.3 Gestión del desastre y logística humanitaria 26](#_Toc511607984)

[2.4 Logística humanitaria en el Perú 28](#_Toc511607985)

[2.5 Optimización Combinatoria 28](#_Toc511607986)

[2.6 Optimización de múltiples objetivos 29](#_Toc511607987)

[2.7 Problema de transporte 29](#_Toc511607988)

[2.8 Problema de transbordo 30](#_Toc511607989)

[2.9 Métodos para la solución de problemas de optimización 31](#_Toc511607990)

[2.9.1 Métodos exactos 31](#_Toc511607991)

[2.9.2 Métodos heurísticos 31](#_Toc511607992)

[2.9.3 Métodos metaheurísticos 32](#_Toc511607993)

[2.9.3.1 Algoritmos Genéticos 33](#_Toc511607994)

[Capítulo 3. Estado del Arte 34](#_Toc511607995)

[3.1 Introducción 34](#_Toc511607996)

[3.2 Investigaciones Académicas 34](#_Toc511607997)

[3.2.1 Transporte en operaciones de respuesta a desastres (*Transportation in disaster response operations*) 34](#_Toc511607998)

[3.2.2 Enrutamiento para esfuerzos de ayuda humanitaria (*Routing for relief efforts*) 34](#_Toc511607999)

[3.2.3 Modelos para enrutamiento de ayuda humanitaria: Equidad, eficiencia y eficacia (*Models for relief routing: Equity, efficiency and efficacy*) 35](#_Toc511608000)

[3.2.4 Óptima reubicación de stock bajo incertidumbre en operaciones humanitarias post-desastre (*Optimal stock relocation under uncertainty in post-disaster humanitarian operations*) 36](#_Toc511608001)

[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*) 36](#_Toc511608002)

[3.2.6 Metaheurística bi-objetivo para la planificación de operaciones de ayuda humanitaria en desastres (*Bi-objective metaheuristic for disaster relief operation planning*) 37](#_Toc511608003)

[3.2.7 Modelo de optimización multi-criterio para la distribución de ayuda humanitaria (*A multi-criteria optimization model for humanitarian aid distribution*) 37](#_Toc511608004)

[3.3 Productos comerciales 38](#_Toc511608005)

[3.3.1 SimpliRoute 38](#_Toc511608006)

[3.3.2 SAP Transportation Management VSR Optimization 39](#_Toc511608007)

[3.3.3 Google Optimization Tools 40](#_Toc511608008)

[3.4 Revisión y discusión 40](#_Toc511608009)

[3.5 Conclusiones 41](#_Toc511608010)

[Capítulo 4. Definición de la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria 42](#_Toc511608011)

[4.1 Introducción 42](#_Toc511608012)

[4.2 Resultado 42](#_Toc511608013)

[4.2.1 Parámetros del problema de optimización 42](#_Toc511608014)

[4.2.1.1 Red de distribución 42](#_Toc511608015)

[4.2.1.2 Horizonte de planificación de la distribución 42](#_Toc511608016)

[4.2.1.3 Oferta y demanda inicial en los nodos de la red de distribución 42](#_Toc511608017)

[4.2.1.4 Vehículos para el transporte de bienes 43](#_Toc511608018)

[4.2.1.5 Volumen y peso de los bienes 43](#_Toc511608019)

[4.2.2 Variables de decisión 44](#_Toc511608020)

[4.2.2.1 Cantidad de vehículos que se movilizan 44](#_Toc511608021)

[4.2.2.2 Cantidad de bienes que se transportan 44](#_Toc511608022)

[4.2.3 Variables auxiliares 44](#_Toc511608023)

[4.2.3.1 Oferta y demanda a lo largo del tiempo 44](#_Toc511608024)

[4.2.3.2 Consumo de bienes 44](#_Toc511608025)

[4.2.3.3 Vehículos disponibles 45](#_Toc511608026)

[4.2.4 Función objetivo 45](#_Toc511608027)

[4.2.4.1 Minimizar demanda actual 45](#_Toc511608028)

[4.2.4.2 Minimizar costo de transporte 45](#_Toc511608029)

[4.2.5 Restricciones 46](#_Toc511608030)

[4.2.5.1 Evitar enviar más bienes de los que se dispone 46](#_Toc511608031)

[4.2.5.2 Evitar usar más vehículos de los que se dispone 46](#_Toc511608032)

[4.2.5.3 Evitar sobrepasar la capacidad de los vehículos 47](#_Toc511608033)

[4.2.5.4 Balance en el inventario 47](#_Toc511608034)

[4.2.5.5 Balance en la cantidad de vehículos disponibles 48](#_Toc511608035)

[4.2.5.6 Determinación del consumo 48](#_Toc511608036)

[4.2.5.7 Actualización de la demanda 49](#_Toc511608037)

[4.2.5.8 Condiciones iniciales del problema 49](#_Toc511608038)

[4.2.6 Rango de existencia 49](#_Toc511608039)

[4.3 Relación con el objetivo 50](#_Toc511608040)

[4.4 Validación de la construcción 50](#_Toc511608041)

[Capítulo 5. Definición de las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria 55](#_Toc511608042)

[5.1 Introducción 55](#_Toc511608043)

[5.2 Resultado 55](#_Toc511608044)

[5.2.1 Estructuras de datos para los parámetros iniciales y variables auxiliares del problema 55](#_Toc511608045)

[5.2.2 Variables de decisión y cromosoma 57](#_Toc511608046)

[5.2.3 Cromosoma 58](#_Toc511608047)

[5.3 Relación con el objetivo 62](#_Toc511608048)

[Capítulo 6. Diseño del algoritmo genético para el problema de optimización de distribución de ayuda humanitaria 63](#_Toc511608049)

[6.1 Introducción 63](#_Toc511608050)

[6.2 Esquema general del algoritmo genético 63](#_Toc511608051)

[6.3 Construcción de la población inicial 63](#_Toc511608052)

[6.3.1 Generación de cromosomas 64](#_Toc511608053)

[6.4 Evaluación de la aptitud 64](#_Toc511608054)

[6.4.1 Frente de Pareto 65](#_Toc511608055)

[6.5 Selección 66](#_Toc511608056)

[6.5.1 Selección por torneo 67](#_Toc511608057)

[6.6 Cruce 68](#_Toc511608058)

[6.7 Mutación 70](#_Toc511608059)

[6.8 Criterio de parada 70](#_Toc511608060)

[Capítulo 7. Experimentación Numérica 71](#_Toc511608061)

[7.1 Introducción 71](#_Toc511608062)

[7.2 Configuración de experimentación 71](#_Toc511608063)

[7.2.1 Tiempo de ejecución 71](#_Toc511608064)

[7.2.1.1 Verificar normalidad 72](#_Toc511608065)

[7.2.1.2 Verificar igualdad de varianzas 72](#_Toc511608066)

[7.2.1.3 Verificar igualdad de medias 72](#_Toc511608067)

[7.2.1.4 Verificar comparación de medias 73](#_Toc511608068)

[7.2.2 Optimalidad 73](#_Toc511608069)

[7.2.2.1 Verificar normalidad 74](#_Toc511608070)

[7.2.2.2 Verificar igualdad de varianzas 74](#_Toc511608071)

[7.2.2.3 Verificar igualdad de medias 75](#_Toc511608072)

[7.2.2.4 Verificar comparación de medias 75](#_Toc511608073)

[Capítulo 8. Conclusiones y trabajos futuros 76](#_Toc511608074)

[8.1 Conclusiones 76](#_Toc511608075)

[8.2 Trabajos futuros 76](#_Toc511608076)

[Referencias 77](#_Toc511608077)

[Anexos a](#_Toc511608078)

# Índice de Figuras

[Ilustración 1. Fenomenología más frecuente en Perú (INDECI, 2008). 11](#_Toc511608088)

[Ilustración 2. Comparación entre resultados de programación entera mixta y heurística (Campbell et al., 2008). 35](#_Toc511608089)

[Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010). 36](#_Toc511608090)

[Ilustración 4. Matriz de recompensa para cada objetivo de optimización (Vitoriano, Ortuño, Tirado, Montero, et al., 2011). 38](#_Toc511608091)

[Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018). 39](#_Toc511608092)

[Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018). 40](#_Toc511608093)

[Ilustración 7. Región de soluciones dominadas por la solución A en un espacio de 2 objetivos (Luke, 2015) 65](#_Toc511608094)

[Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015). 66](#_Toc511608095)

[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](#_Toc511608096)

**Índice de Tablas**

[Tabla 1. Cronograma del proyecto 23](#_Toc511608097)

[Tabla 2. Matriz de riesgos 25](#_Toc511608098)

# Generalidades

## 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).



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 Chincha 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.

## Objetivos

### Objetivo general

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

### Objetivos específicos

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

### Resultados esperados

1. Función objetivo, restricciones, variables de decisión y parámetros definidos para el problema de distribución de ayuda humanitaria (O1).
2. Estructuras de datos del algoritmo genético propuesto (O2).
3. Pseudocódigo del algoritmo genético conteniendo la función evaluación de aptitud y los operadores (O3).
4. Algoritmo genético implementado y contenido en un componente reutilizable (O4).
5. Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver (O5).
6. Componente de software que contenga el programa lineal y ejecute el solver (O5)
7. Desarrollo de un generador de datos de prueba para los algoritmos de optimización (O6).
8. Análisis de datos de un caso real para generar un caso de prueba basada en un escenario real (O6).
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).
10. Instancias de prueba de diferente tamaño generadas aleatoriamente (O6).
11. Informe de evaluación y comparación de tiempo de ejecución y optimalidad de los algoritmos de optimización mediante experimentación numérica (O6).
12. Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el programa lineal (O7).

### Mapeo de objetivos, resultados y verificación

|  |  |  |
| --- | --- | --- |
| **Objetivo (O1):** Definir la formulación de la función objetivo, restricciones, variables de decisión y parámetros del problema de distribución de ayuda humanitaria | | |
| **Resultado (R1)** | **Meta física** | **Medio de verificación** |
| Función objetivo, restricciones, variables de decisión y parámetros definidos | Documento | * Definición de función objetivo, restricciones, variables de decisión y parámetros en el documento |
| **Objetivo (O2):** Definir las estructuras que soporten los datos y el algoritmo para la resolución del problema de distribución de ayuda humanitaria | | |
| **Resultado (R2)** | **Meta física** | **Medio de verificación** |
| Estructuras de datos definidas | Documento | * Definición de estructuras de datos en el documento |
| **Objetivo (O3):** Definir y diseñar la función de aptitud y operadores del algoritmo genético | | |
| **Resultado (R3)** | **Meta física** | **Medio de verificación** |
| Pseudocódigo del algoritmo genético conteniendo la función evaluación de aptitud y los operadores | Documento | * Definición del pseudocódigo de la función de evaluación de aptitud y los operadores del algoritmo genético en el documento |
| **Objetivo (O4):** Implementar el algoritmo genético para la resolución del problema de distribución de ayuda humanitaria | | |
| **Resultado (R4)** | **Meta física** | **Medio de verificación** |
| Algoritmo genético implementado y contenido en un componente reutilizable | Software | * Código del algoritmo genético * Interfaz en consola para ejecutar el algoritmo genético |
| **Objetivo (O5):** Diseñar e implementar un programa lineal a partir de la formulación del problema de optimización | | |
| **Resultado (R5)** | **Meta física** | **Medio de verificación** |
| Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver | Software | * Programa lineal formulado e implementado en un lenguaje de modelado para su optimización mediante un solver |
| **Resultado (R6)** | **Meta física** | * **Medio de verificación** |
| Componente de software que contenga el programa lineal y ejecute el solver | Software | * Código del componente * Interfaz en consola para ejecutar el solver |
| **Objetivo (O6):** Diseñar y desarrollar la experimentación numérica para comparar el desempeño del algoritmo genético con el método de programación lineal | | |
| **Resultado (R7)** | **Meta física** | **Medio de verificación** |
| Análisis y desarrollo de un generador de datos de prueba para los algoritmos de optimización | Software | * Código del generador de datos * Interfaz en consola para generar datos |
| **Resultado (R8)** | **Meta física** | **Medio de verificación** |
| Análisis de datos de un caso real para generar un caso de prueba basada en un escenario real | Documento | * Análisis de los datos utilizados de un caso real describiendo dimensiones del problema de un escenario real |
| **Resultado (R9)** | **Meta física** | **Medio de verificación** |
| Instancia de prueba basada en un escenario real conteniendo información sobre la red distribución, así como la oferta, demanda y tipos de bienes y vehículos a transportar | Dataset | * Archivo con datos describiendo los datos de entrada de un escenario real del problema de optimización |
| **Resultado (R10)** | **Meta física** | **Medio de verificación** |
| Instancias de prueba de diferente tamaño generadas aleatoriamente | Dataset | * Archivo con datos describiendo datos de entrada de diferente tamaño para los algoritmos de optimización |
| **Resultado (R11)** | **Meta física** | **Medio de verificación** |
| Informe de evaluación y comparación de 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 |
| **Objetivo (O7):** Desarrollar la interfaz de usuario para la ejecución y visualización de los algoritmos de optimización | | |
| **Resultado (R12)** | **Meta física** | **Medio de verificación** |
| Interfaz de usuario implementada y conteniendo los módulos del algoritmo genético y el programa lineal | Software | * Código de la interfaz de usuario * Interfaz gráfica para ejecutar los algoritmos de optimización |

## Herramientas y Métodos

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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.

### 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 avance 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.

### 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.

### 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.

### 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.

### 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.

## Viabilidad

### 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 desarrollo 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.

### 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 |

### 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.

### 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.

## Alcance, Limitaciones y Riesgos

### 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.

### 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.

### Riesgos

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

Tabla 2. Matriz de riesgos

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

# Marco Conceptual

## 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.

## 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

## 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 inter-gubernamentales, 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.

## 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.

## Optimización Combinatoria

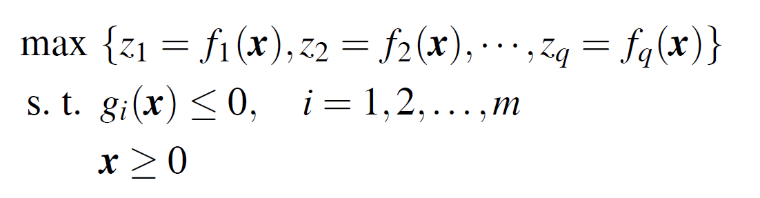
La Optimización Combinatoria tiene como objetivo encontrar un objeto óptimo en una colección finita de objetos (Papadimitriou & Steiglitz, 1982). Típicamente, dicha colección posee una representación concisa, como una red de transporte, por ejemplo, mientras que el número de objetos es enorme (todas 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.

## 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:



Donde x es el conjunto de variables de decisión, f(x) es la función objetivo y g(x) son funciones que representan las restricciones del problema, sin pérdida de generalidad, un problema de optimización puede ser formulado como:



## 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:

Donde:

* La función objetivo es el costo total de distribución
* xij es el número de unidades a ser transportadas desde el origen i al destino j
* cij es el costo de transportar unidades desde el origen i al destino j
* si es el nivel de oferta en el nodo i
* di es el nivel de demanda en el nodo i

## 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:

Donde:

* La función objetivo es el costo total de distribución
* xij es el número de unidades a ser transportadas desde el origen i al destino j
* cij es el costo de transportar unidades desde el origen i al destino j
* uij es la capacidad máxima de unidades que se pueden transportar desde i a j
* bi es el flujo neto generado en el nodo i
  + bi > 0, si el nodo i es un nodo de oferta
  + bi < 0, si el nodo i es un nodo de demanda
  + bi = 0, si el nodo i es un nodo de transbordo

## Métodos para la solución de problemas de optimización

### 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

### 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, los 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.

### 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.

### 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

# Estado del Arte

## 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.

## Investigaciones Académicas

### 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.

### 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).

### 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.

### 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:

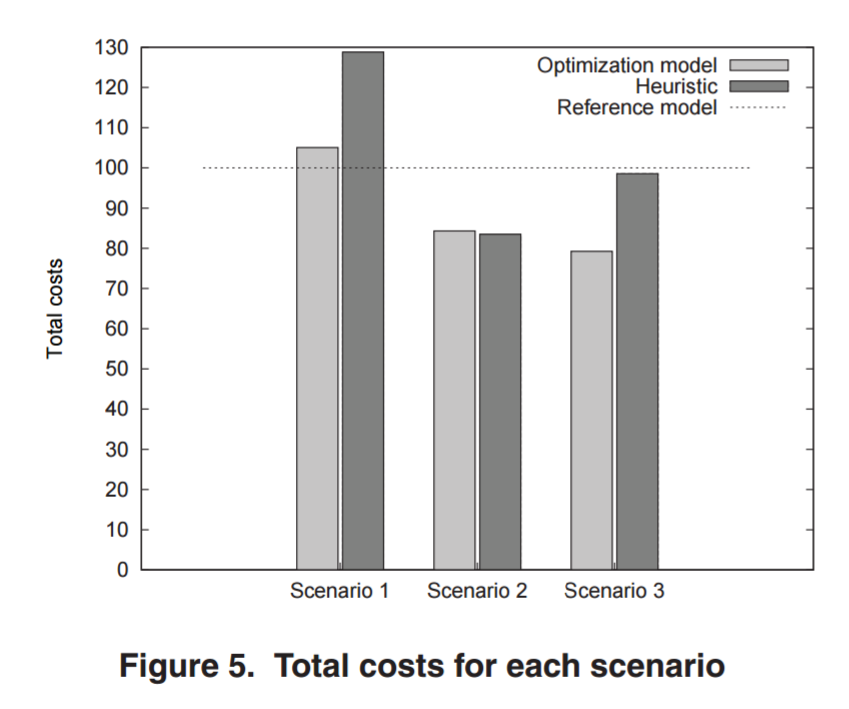


Ilustración 3. Comparación de costos entre el método exacto y la heurística (Blecken et al., 2010).

### 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.

### 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.

### 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.

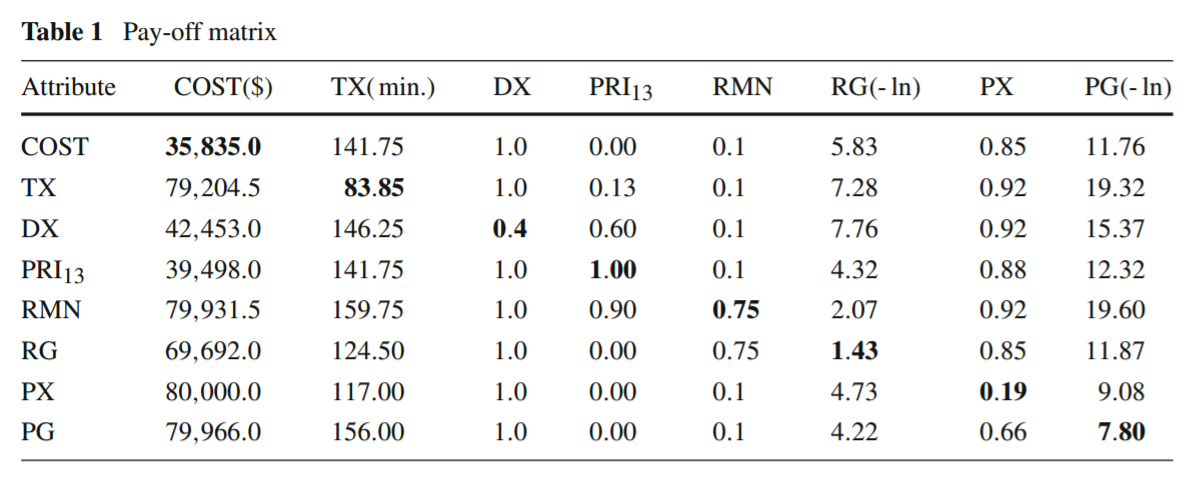


Ilustración 4. Matriz de recompensa para cada objetivo de optimización (Vitoriano, Ortuño, Tirado, Montero, et al., 2011).

## Productos comerciales

### 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.

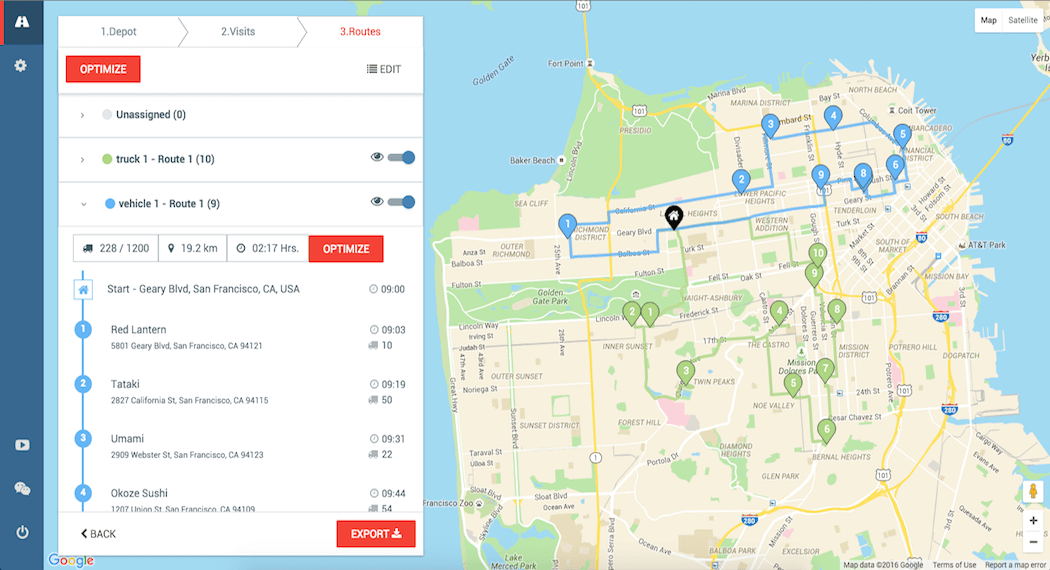


Ilustración 5. SimpliRoute, ejemplo de interfaz (SimpliRoute, 2018).

### 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).

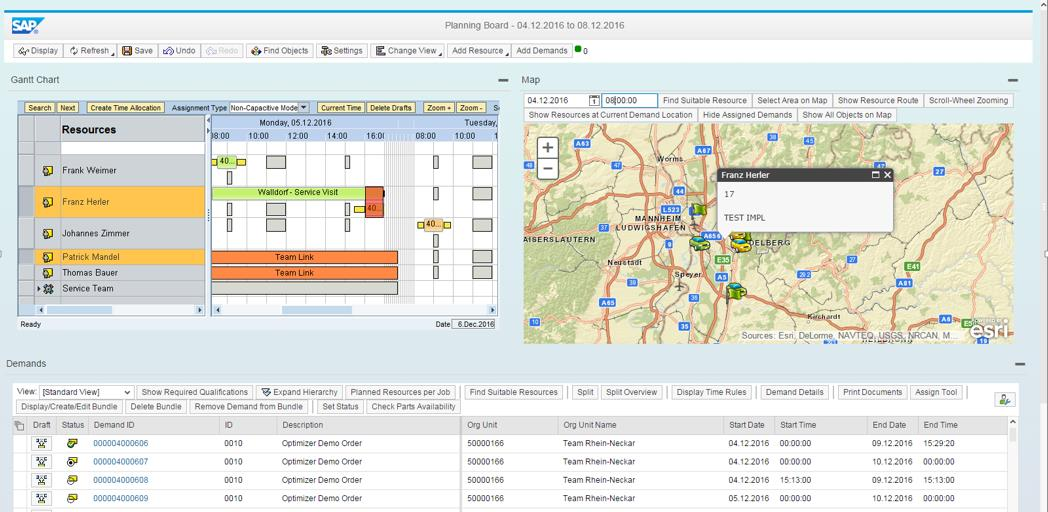


Ilustración 6. SAP, ejemplo de interfaz (SAP, 2018).

### 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.

## 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.

## 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.

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

## 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

## Resultado

### Parámetros del problema de optimización

### 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.

### Costo de transporte

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

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

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

Para considerar bienes de diferentes tipos, estos parámetros se pueden expandir a ***Inventarioki*** y ***Demandaki***, que indican la oferta y demanda de bienes de tipo i en el nodo k.

### Capacidad de almacenes

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

### Variables de decisión

### Cantidad de bienes que se transportan

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

### Función objetivo

### Minimizar costo de transporte

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

### Minimizar inequidad en la distribución

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

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

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

Para lograr solucionar este hecho, es posible usar el promedio de los cuadrados de los porcentajes de satisfacción. Debido a que la función cuadrática crece de forma más rápida que la función identidad, esta función puede penalizar de manera más drástica a soluciones altamente no equitativas. Para el mismo ejemplo planteado anteriormente, la primera solución que realiza un reparto equitativo tendría un promedio de cuadrados de 0.0625, y llevándolo a la escala original tomando la raíz cuadrada, resulta en un promedio de 25% (). Por otro lado, la solución que no realiza un reparto equitativo tiene un promedio de cuadrados de 0.25, y de la misma manera, llevándolo a la escala original tomando la raíz cuadrada, resulta en un promedio de 50% (), el cual es un valor mayor a la solución óptima. De hecho, se puede notar que cualquier otra combinación de distribución obtendrá un promedio mayor o igual a 25%. De esta manera, con este tipo de promedio, sí es posible distinguir la diferencia entre un reparto equitativo y uno no equitativo.

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

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

### Restricciones

### Balance en los puntos de abastecimiento

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

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

### Balance en los puntos de demanda

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

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

### Balance en los puntos de transbordo

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

### Evitar sobrepasar la capacidad de los almacenes

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

### Rango de existencia

Dado que todas las variables y parámetros a excepción de los parámetros referentes a costo de transporte representan cantidades discretas, el rango de existencia es el conjunto de los números enteros no negativos: . El rango de existencia de los parámetros de costo de transporte es el conjunto de los números reales no negativos:

## Relación con el objetivo

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

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

## 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.

## Resultado

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

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

* Parámetros
  + **K**: número de nodos en la red de distribución
  + **I**: número de tipos de bienes (suministros) de ayuda humanitaria
  + ***Inventarioki***: inventario total de bienes de tipo i en el nodo k
  + ***Demandaki***: demanda total de bienes de tipo i en el nodo k
  + ***Capacidadk***: capacidad total del nodo de distribución k
  + ***CostoTranspjk***: costo de transporte unitario entre los nodos j y k

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

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

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

Clase ***Nodo*** {

Entero ***idNodo***;

Arreglo de enteros [K, I] ***Demanda***;

Arreglo de enteros [K, I] ***Inventario***;

Arreglo de enteros [K] ***Capacidad***;

}

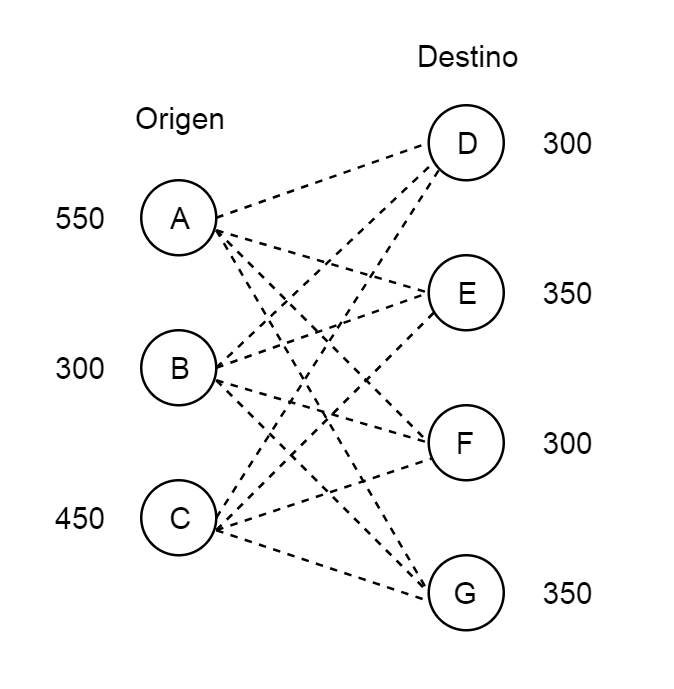
### Variable de decisión y cromosoma

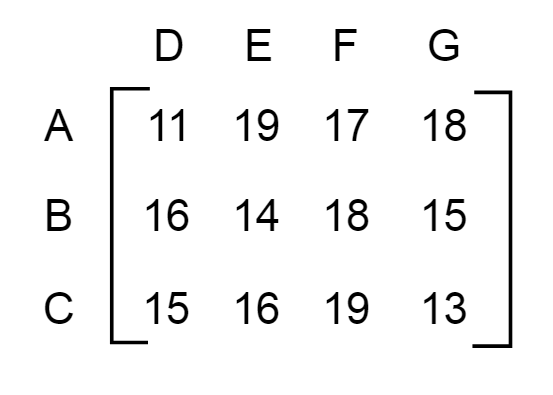
La variable de decisión que se formuló en el capítulo anterior es la siguiente:

* ***Transpjki***, que indica la cantidad de bienes (suministros de ayuda humanitaria) de tipo i que se transportan desde el nodo j hacia k

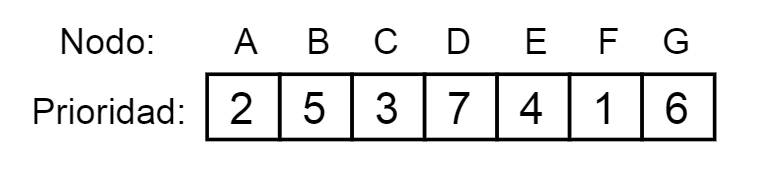
Esta variable de decisión es la que genera un plan de distribución y describe una posible solución para el problema de optimización. De acuerdo con primeras investigaciones en el tema de algoritmos genéticos aplicados al problema de transporte (Vignaux & Michalewicz, 1991), una manera inicial de definir el cromosoma es hacer que el cromosoma represente directamente dicha variable de decisión. Sin embargo, decidir representar la matriz de transporte directamente en el cromosoma hace complicado el uso de operadores genéticos como mutación y especialmente cruce, siendo una de las razones principales el hecho de que las redes de transporte no suelen están 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, Altiparmak, & 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.

### Cromosoma

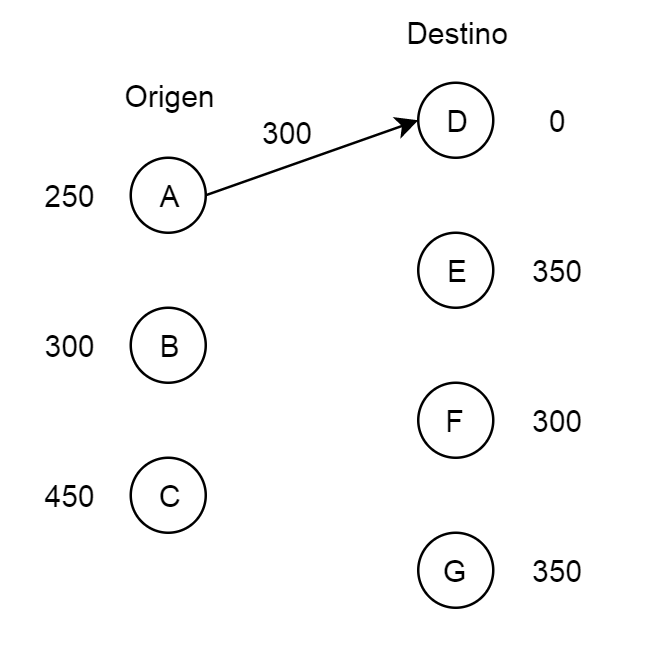
De acuerdo a lo mencionado en la sección anterior, el cromosoma consiste en una asignación de prioridades a los nodos de la red de transporte. Por ejemplo, sea la siguiente red de distribución, donde cada círculo representa un nodo de oferta (izquierda) o demanda (derecha), junto con la cantidad de oferta y demanda expresado en valores numéricos (en el contexto de ayuda humanitaria, la oferta son los bienes de ayuda disponibles en almacenes y la demanda es la necesidad de ayuda en puntos de desastre). La matriz de la izquierda representa la matriz de costos de la red de transporte, donde el elemento de la fila i y la columna j de la matriz indica el costo para llegar desde i hasta j:



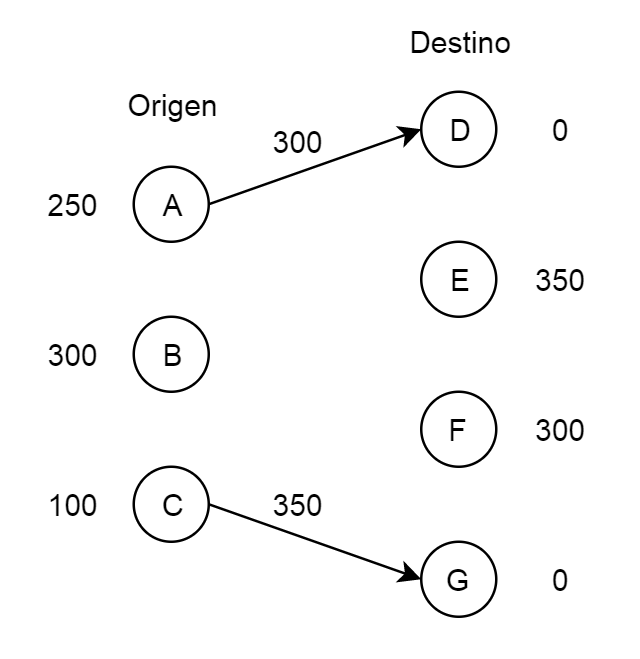
Un posible cromosoma para este problema sería el siguiente:



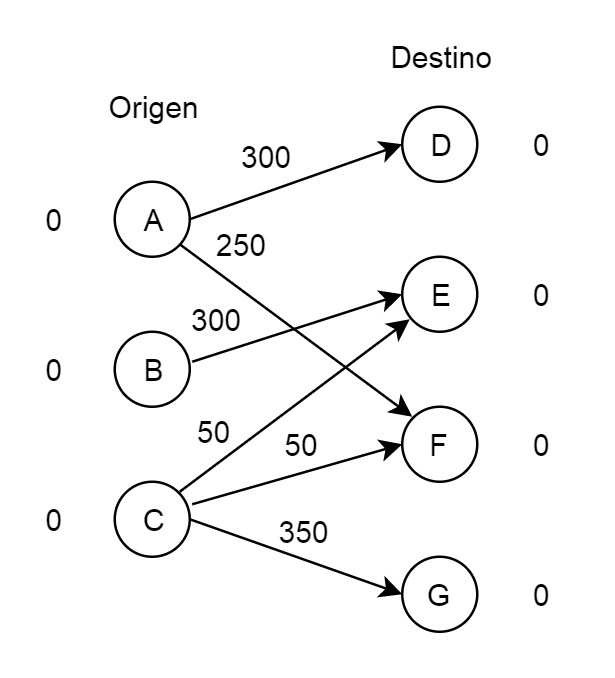
En este cromosoma, el valor de prioridad más alto (7) ocurre en el nodo destino D, lo cual significa que se debe atender primero, para lo cual se selecciona el nodo origen que tenga el menor costo para llegar a D. Observando la matriz de costos, el nodo origen A tiene el menor costo (11), por lo cual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

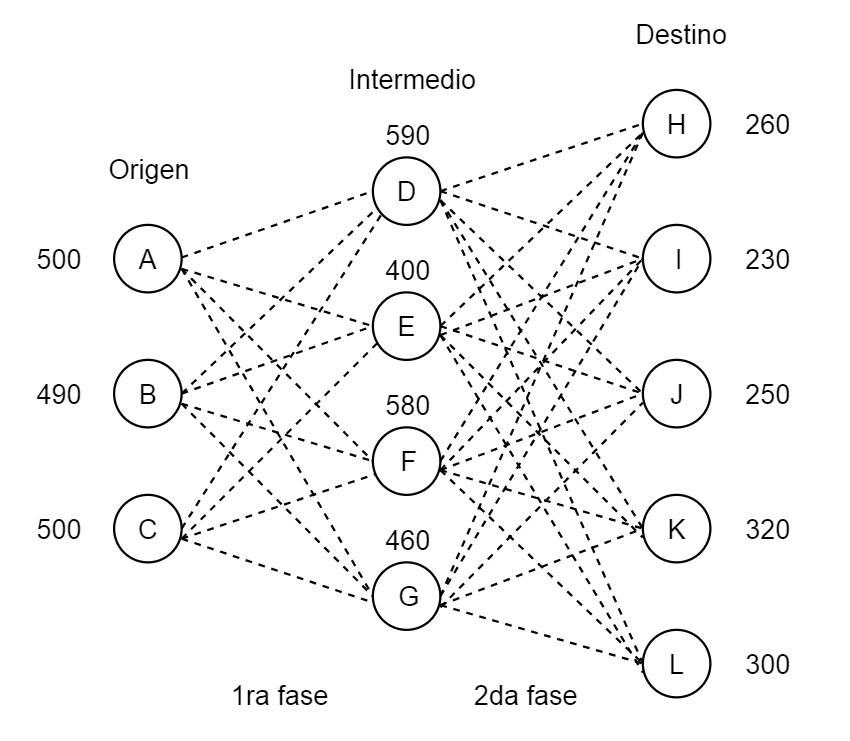
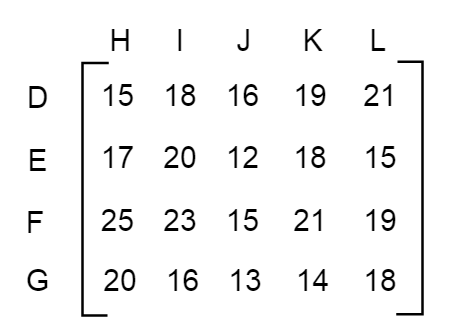
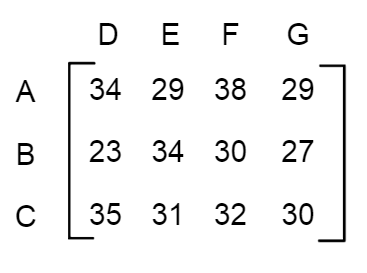
A partir del ejemplo previo, se define el siguiente algoritmo de decodificación de cromosoma:

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 (ofertaorigen, demandanodo)
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 (ofertanodo, demandadestino)
11. ActualizarOfertaYDemanda(*nodo*, *destino*, *bienesATransportar*)

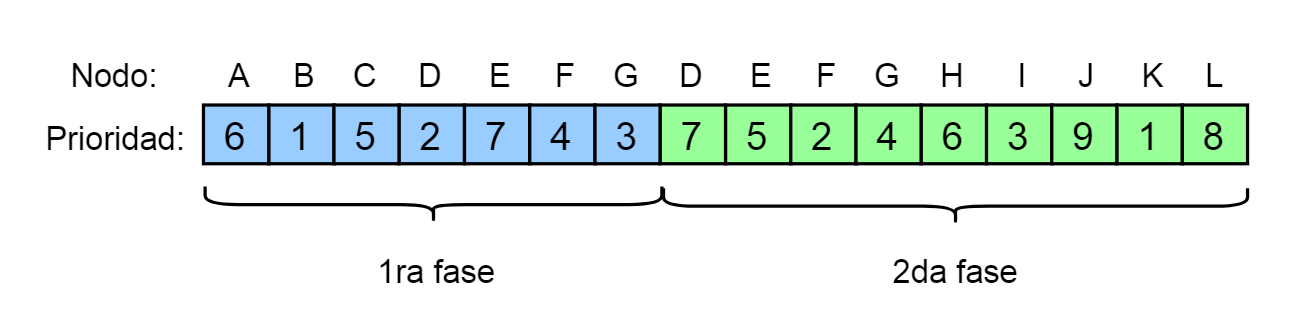
Para considerar las restricciones de capacidad y disponibilidad de vehículos definidas previamente en la formulación, basta someter la variable *bienesAtransportar* a la cantidad de vehículos disponibles en el nodo de origen. De esta manera, el límite superior de la variable *bienesAtransportar* será 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 diversos tipos de redes de distribución. Para este ejemplo, a partir del cromosoma mostrado, la aplicación del algoritmo de decodificación daría como resultado la siguiente asignación:



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.

## 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.

# Diseño del algoritmo genético para el problema de optimización de distribución de ayuda humanitaria

## 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.

## Esquema general del algoritmo genético

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

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

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

## 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

### 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.

## Evaluación de la aptitud

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

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

### 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.

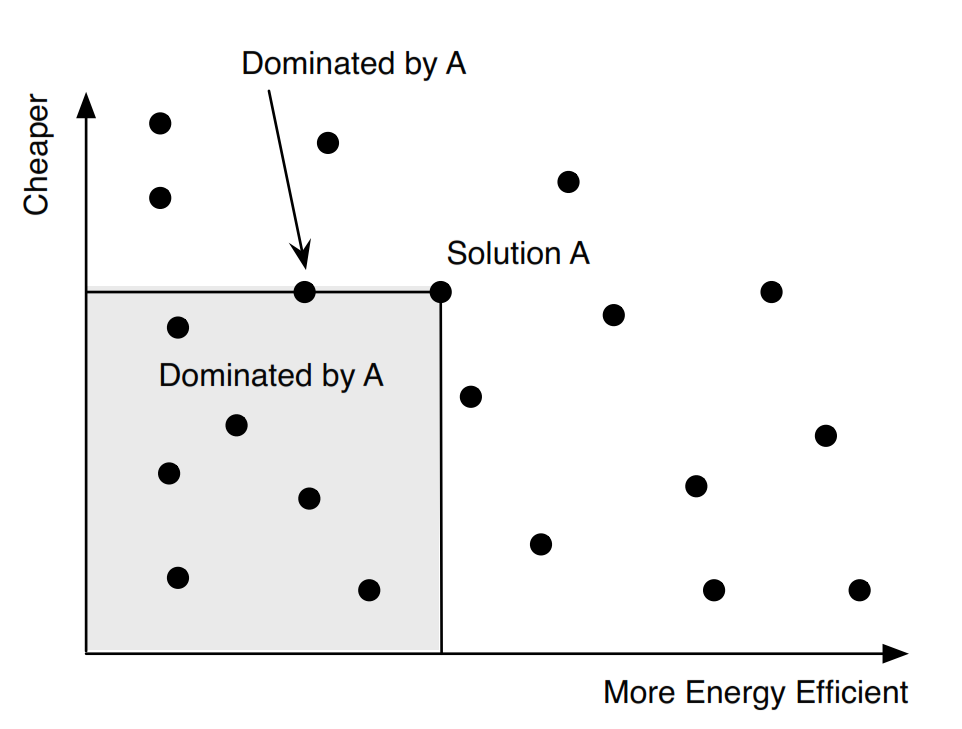


Ilustración 7. Región de soluciones dominadas por la solución A en un espacio de 2 objetivos (Luke, 2015)

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**:

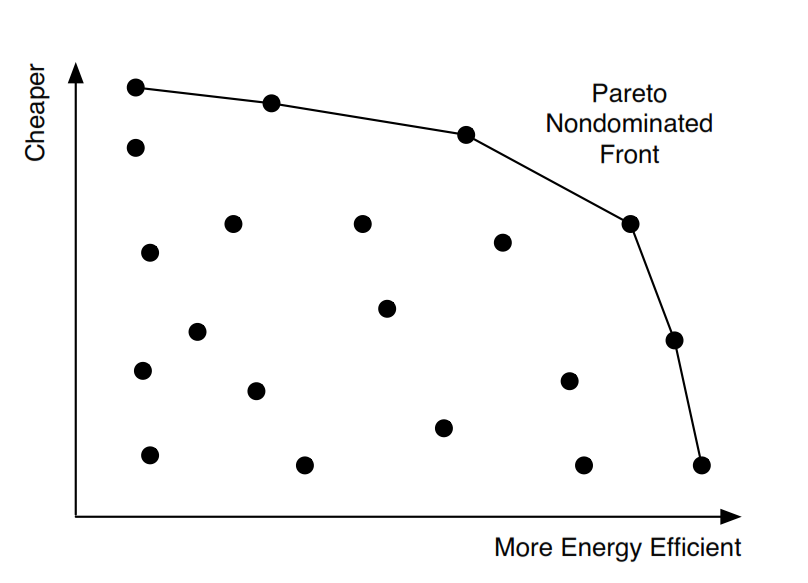


Ilustración 8. Frente de Pareto de soluciones no dominadas (Luke, 2015).

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.

## 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 esDominado entonces
10. Rango[p] = i
11. frentePareto.añadirElemento (p)
12. P’ = P’ – 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.

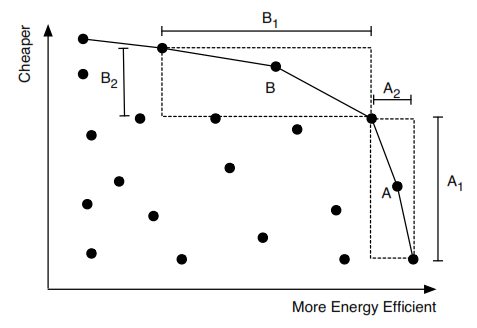


Ilustración 9. Dispersión de las soluciones A y B. B está más alejado de sus vecinos, por lo cual tendría un valor más alto de dispersión (Luke, 2015).

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**.

### Selección por torneo

La selección por torneo es un método de selección en algoritmos genéticos que se caracteriza por su invarianza a la escala del valor de la aptitud de los individuos. Esto la hace ideal para diferentes tipos de objetivos y en especial cuando el algoritmo genético se encuentra cerca de la convergencia, donde los valores de aptitud serían cercanos entre sí y no se distinguirían mucho (Luke, 2015).

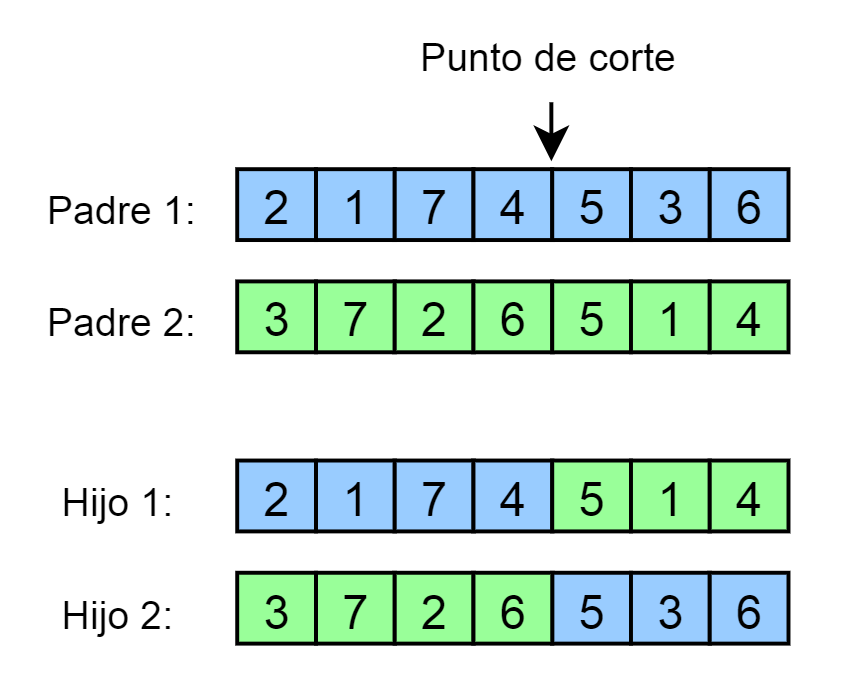
El algoritmo de selección por torneo es el siguiente:

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

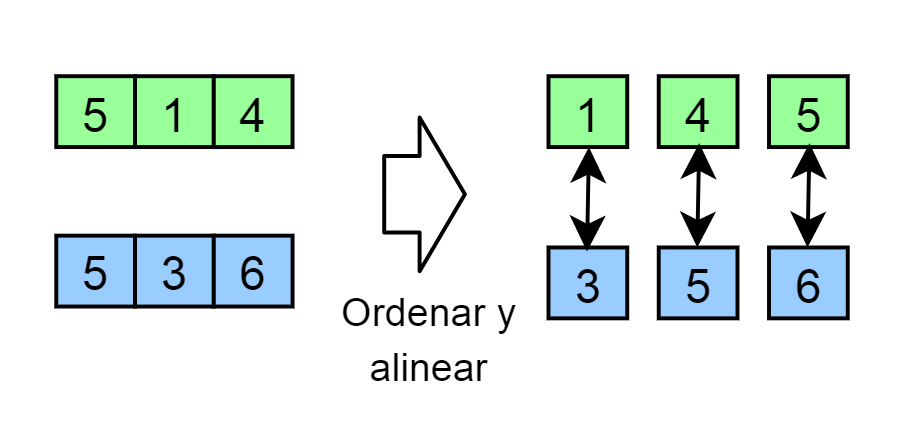
Para evaluar la aptitud se utilizará el concepto de aptitud por dominancia definido en la sección anterior.

## Cruce

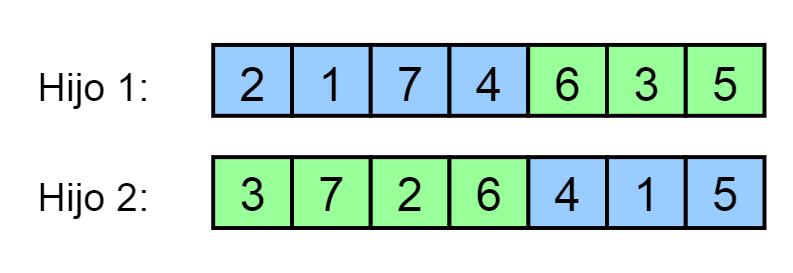
El operador de cruce permite juntar la información de dos individuos, preferiblemente los más aptos, con el fin de crear individuos que contengan las mejores característicos de ambos padres. Dado que el cromosoma representa permutaciones, es posible aplicar operadores de cruce específicos para permutaciones. 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 una reasignación entre los valores que fueron intercambiados, para ello, primero se ordenan las secciones que fueron intercambiadas para luego alinearlas, con lo cual se obtiene el mapeo:



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

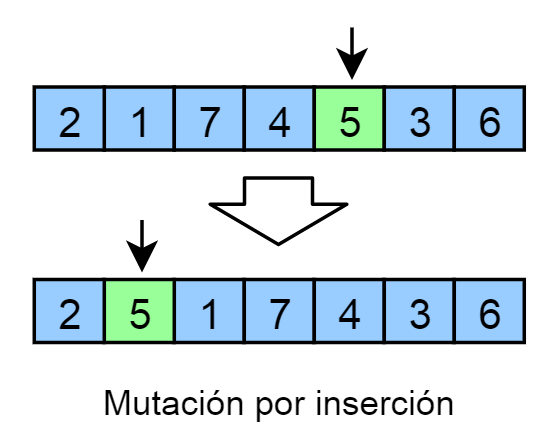
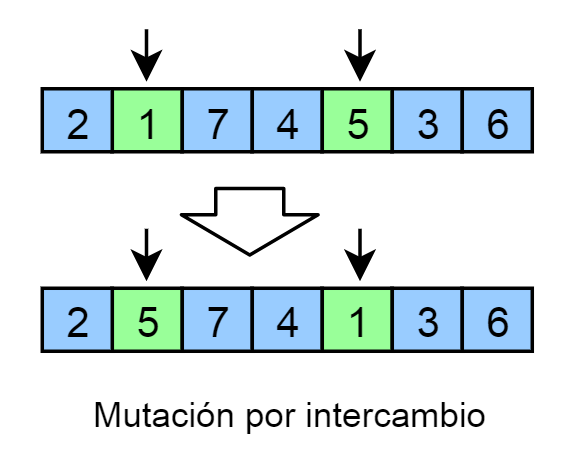


Esto queda reflejado en el siguiente algoritmo de cruce:

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

## 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.

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.

## Criterio de parada

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

# Diseño del modelo matemático de programación no lineal para el problema de optimización de distribución de ayuda humanitaria

## Introducción

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

## Formulación del modelo matemático

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

### Conjuntos

### Parámetros

### Variables de decisión

### Funciones objetivo

### Restricciones

### Rango de existencia

## Algoritmo de optimización exacto (*solver*)

El algoritmo (*solver*) que se utilizará para resolver el programa no lineal de forma exacta es Couenne. Couenne (*Convex Over and Under ENvelopes for Nonlinear Estimation*) es un algoritmo de tipo ramificación y poda (*branch & bound*) para resolver problemas de programación no lineal entera mixta (MINLP) de forma exacta. Para utilizarlo, se debe implementar la formulación matemática en un lenguaje de modelado como AMPL (*A Mathematical Programming Language*). Un ejemplo del lenguaje AMPL es el que se muestra a continuación:

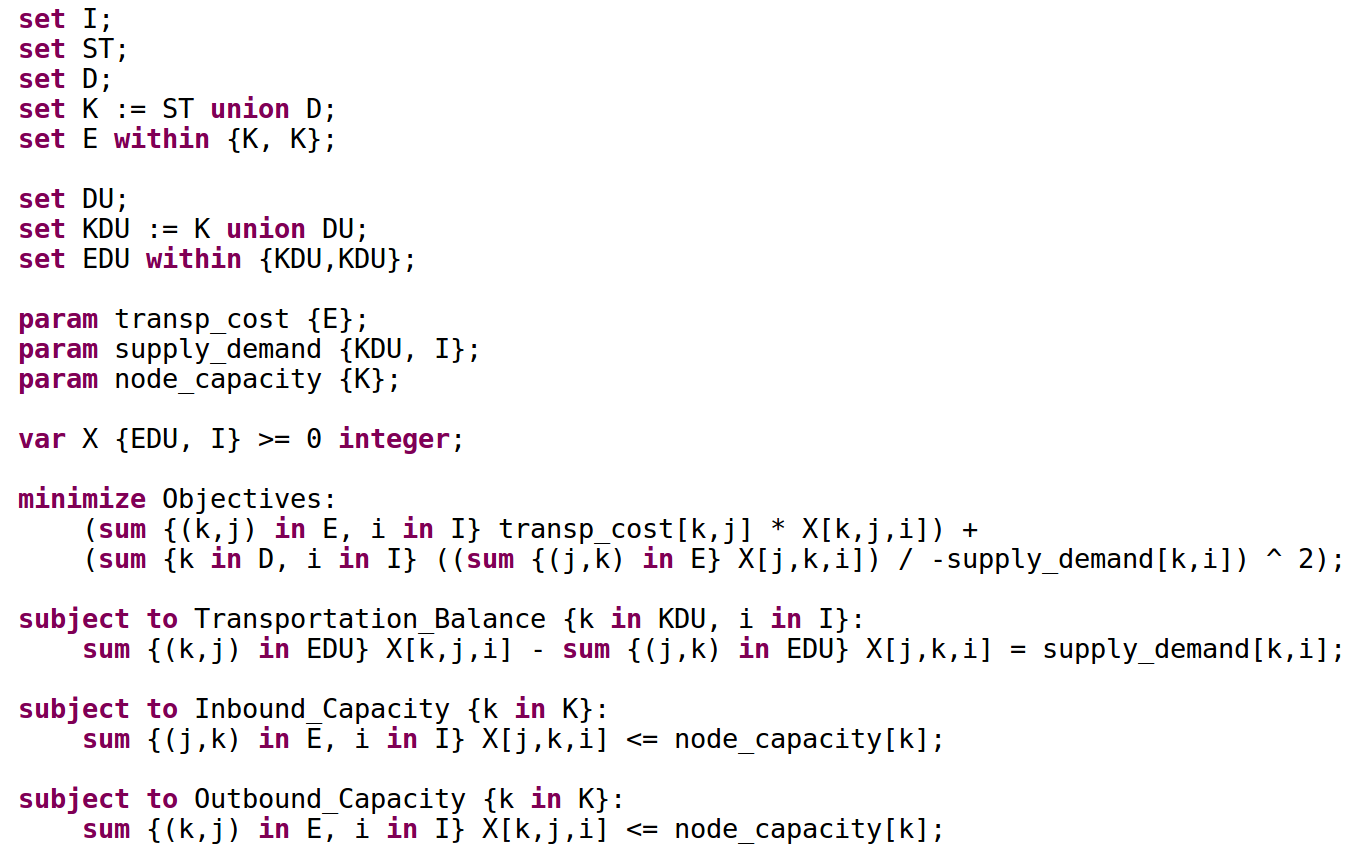


Ilustración 10: Lenguaje de modelado AMPL

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

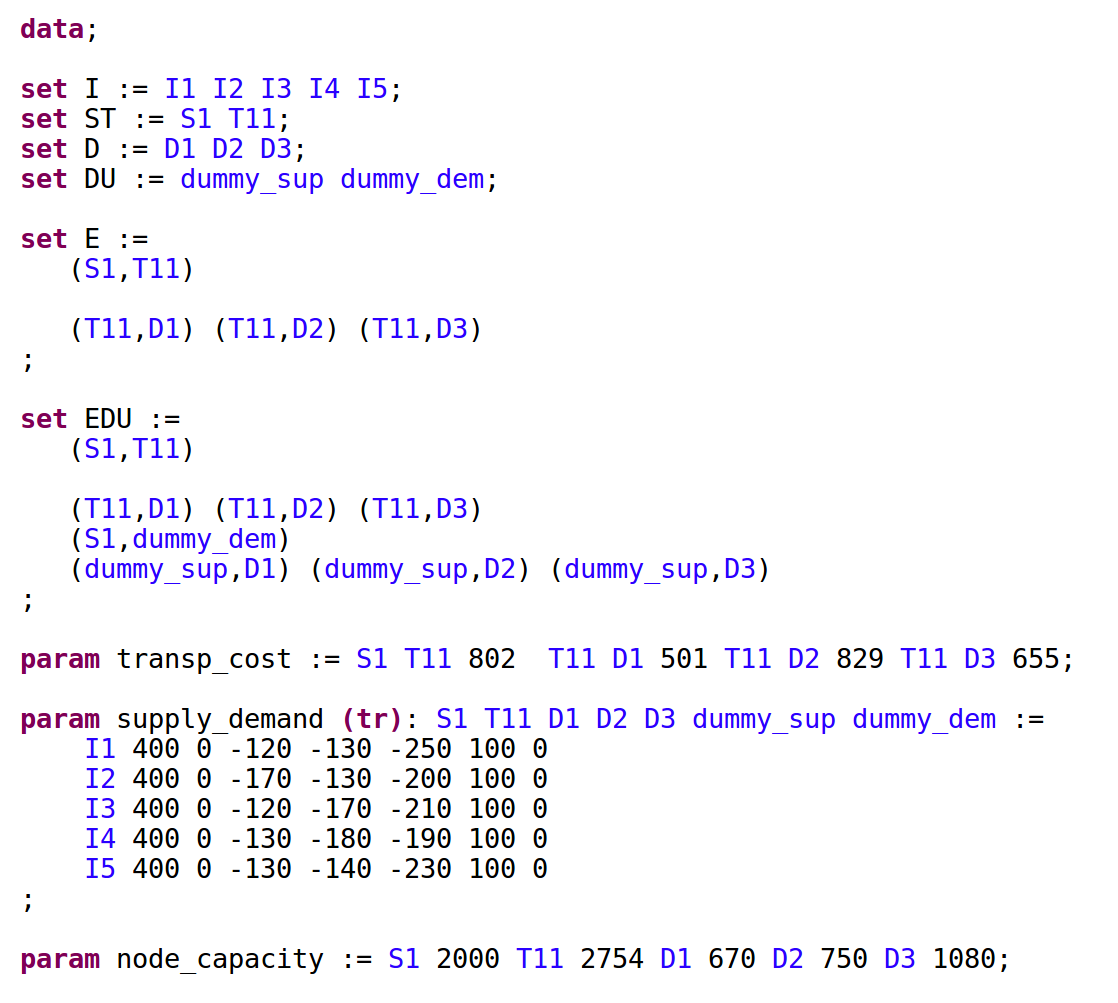


Ilustración 11: Ingreso de datos en lenguaje AMPL

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



Ilustración 12: Ejecución y reporte de resultados de Couenne

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

# Experimentación Numérica

## Introducción

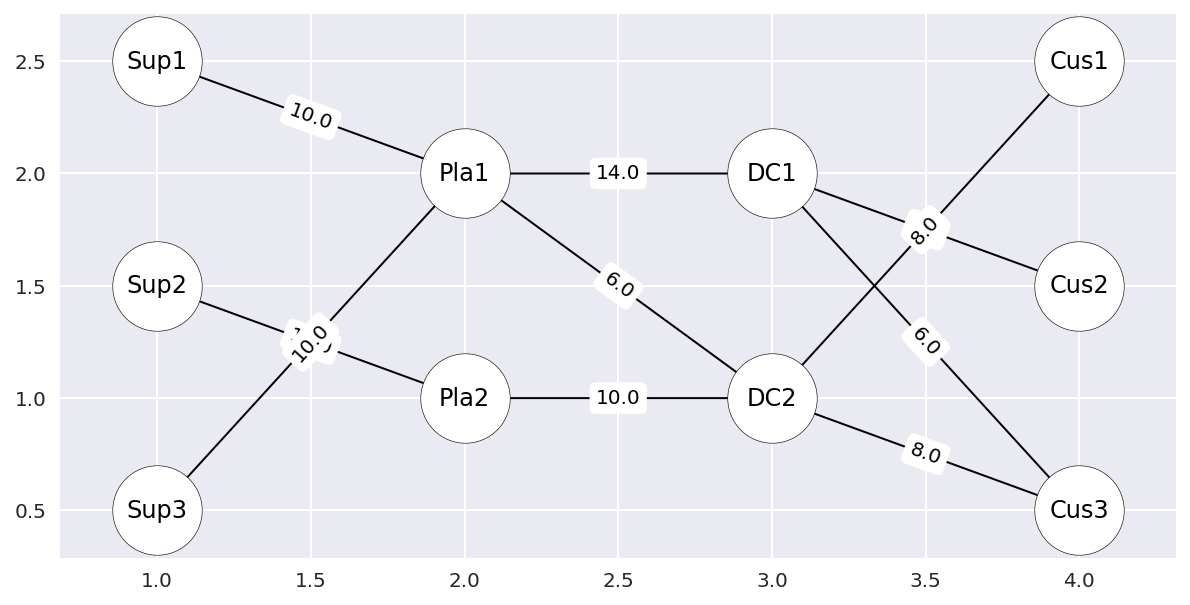
Una vez obtenidos los resultados del algoritmo genético y la optimización del programa no lineal entero mixto mediante el algoritmo Couenne (*solver*), es momento de comparar dichos resultados para validar las hipótesis de investigación del presente proyecto de fin de carrera.

## Configuración de la experimentación

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

### Generación de conjuntos de datos aleatorios

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



El primer nivel se asigna para los almacenes, el último nivel para los puntos de demanda, y los niveles intermedios para los puntos de transbordo. Una vez determinados los puntos de abastecimiento y demanda, se asigna una cantidad aleatoria de bienes disponibles en los almacenes y bienes requeridos en los puntos de demanda tomando muestras de la distribución de Poisson, la cual es una distribución discreta comúnmente usada para simular cantidades discretas (Montgomery, 2011).

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

### Toma de datos

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

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

## Resultados de la experimentación

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

### Tiempo de ejecución

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

Para visualizar mejor los resultados se presentan los siguientes gráficos:

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

Esta información nos es de utilidad para determinar a partir de qué punto conviene cambiar de método de optimización, considerando su uso en un sistema de apoyo a la toma de decisiones. Para poder evaluar el punto a partir del cual el algoritmo genético es más rápido considerando que sus tiempos de ejecución son variables, se pueden utilizar los p-valores obtenidos de aplicar pruebas de hipótesis para la comparación de medias de los tiempos de ejecución sobre las muestras de las corridas del algoritmo. Como un p-valor decrece cuando la diferencia de medias es menor, el punto de cruce entre los tiempos de ejecución quedará determinado por el menor p-valor. Primero se realizará una prueba de normalidad para verificar que los tiempos de ejecución del algoritmo genético siguen una distribución normal, y luego se utilizará una prueba de comparación de medias para obtener los p-valores.

### 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.

### 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.

En los siguientes gráficos se muestran los p-valores para cada número de tipos de bienes a distribuir.

Se puede notar que los menores p-valores corresponden a los puntos de cruce vistos en los gráficos comparativos mostrados anteriormente.

### Valor objetivo

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:

# Conclusiones y trabajos futuros

## Conclusiones

## 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

Ortuño, M. T., Cristóbal, P., Ferrer, J. M., Martín-Campo, F. J., Muñoz, S., Tirado, G., & Vitoriano, B. (2013). Decision Aid Models and Systems for Humanitarian Logistics. A Survey. *Atlantis Computational Intelligence Systems*, *7*. https://doi.org/10.2991/978-94-91216-74-9\_2

Papadimitriou, C. H., & Steiglitz, K. (1982). *Combinatorial optimization : algorithms and complexity*. Prentice Hall.

Razali, N. M., & Wah, Y. B. (2011). Power comparisons of Shapiro-Wilk, Kolmogorov-Smirnov, Lilliefors and Anderson-Darling tests - Semantic Scholar. Retrieved April 14, 2018, from https://www.semanticscholar.org/paper/Power-comparisons-of-Shapiro-Wilk%2C-Lilliefors-and-Razali-Wah/dcdc0a0be7d65257c4e6a9117f69e246fb227423

SAP. (2018). VSR Optimization. Retrieved from https://help.sap.com/doc/saphelp\_tm80/8.0/en-US/4a/5a56e958b14204b9d9a65fed58ea6f/content.htm?no\_cache=true

Schrijver, A. (2003). *Combinatorial optimization : polyhedra and efficiency*. Springer.

Shapiro, S. S., & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, *52*(3–4), 591–611. https://doi.org/10.1093/biomet/52.3-4.591

Sheng, S., Dechen, Z., & Xiaofei, X. (2006). Genetic Algorithm for the Transportation Problem with Discontinuous Piecewise Linear Cost Function. *IJCSNS International Journal of Computer Science and Network Security*, *6*(7A). Retrieved from https://pdfs.semanticscholar.org/e591/b7cc76501ed2b371008d286ced7a2a4ded37.pdf

SimpliRoute. (2018). SimpliRoute. Retrieved April 14, 2018, from https://www.simpliroute.com/

Toth, P., & Vigo, D. (2002). *The vehicle routing problem*. Society for Industrial and Applied Mathematics. Retrieved from https://dl.acm.org/citation.cfm?id=505847

Tzeng, G.-H., Cheng, H.-J., & Huang, T. D. (2007). Multi-objective optimal planning for designing relief delivery systems. *Transportation Research Part E 43*. https://doi.org/10.1016/j.tre.2006.10.012

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

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.