

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



---

**Maestría en Ciencias  
en Ciencias de la Computación**

---

**Optimización multiobjetivo para la planificación de transporte  
público aplicando técnicas metaheurísticas**

Tesis  
para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:

**David Peña Morales**

Ensenada, Baja California, México  
2017

Tesis defendida por  
**David Peña Morales**

y aprobada por el siguiente Comité

---

**Dr. Andrey Chernykh**  
Director de tesis

**Dr. Carlos Alberto Brizuela Rodríguez**

**Dr. Jesús Favela Vara**

**M.C. Jorge Enrique Preciado Velasco**



---

**Dr. Jesús Favela Vara**  
Coordinador del Posgrado en Ciencias de la  
Computación

---

**Dra. Rufina Hernández Martínez**  
Directora de Estudios de Posgrado

*David Peña Morales © 2017*

*Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis.*



Resumen de la tesis que presenta **David Peña Morales** como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

### **Optimización multiobjetivo para la planificación de transporte público aplicando técnicas metaheurísticas**

Resumen aprobado por:

---

**Dr. Andrey Chernykh**  
**Director de tesis**

El transporte público es un tema importante para el desarrollo de las ciudades, dado que afecta directamente el bienestar de las personas que deben trasladarse por trabajo, estudio, asistencia médica o entretenimiento. La calidad del servicio debe garantizar una experiencia que cumpla con criterios como disponibilidad, comodidad, seguridad y tiempos de espera cortos. Por otro lado, las empresas proveedoras del servicio buscan reducir los costos de operación (combustible, mantenimiento o pago al conductor); por lo que, una asignación adecuada de la flota para cierta demanda de pasajeros reduce la circulación de vehículos semivacíos haciendo más rentable la operación, asimismo favorece con la descongestión de las vías y el impacto ambiental, al disminuir el consumo innecesario de combustible. El diseño de horarios y la calendarización de los vehículos forman parte de un conjunto de tareas importantes para la planificación de transporte público; ya que ambos dependen del estudio y optimización de la frecuencia de circulación, por consiguiente, es importante para evitar sobrecostos y mejorar la calidad del servicio percibida por el usuario; lo cual es esencial a la hora de proponer al transporte público como una opción atractiva frente al transporte privado. En esta tesis se propone un algoritmo evolutivo multiobjetivo para la creación de horarios utilizando vehículos de diferentes tipos (diferente capacidad y costo de funcionamiento), considerando las restricciones de las entidades gubernamentales para los sistemas de transporte público en el contexto de ciudades inteligentes. Los objetivos en conflicto abordados son: la minimización del costo total de circulación, que afecta al operador, y la minimización del número de pasajeros que no pueden trasladarse en un periodo de tiempo, lo cual supone una reducción en la congestión de los vehículos y menor tiempo de espera. Un conjunto de soluciones no-dominadas representa diferentes asignaciones de vehículos a un horario, que satisface un conjunto de viajes para una ruta definida, en este documento se presenta el análisis experimental realizado, asimismo la comparación entre los resultados obtenidos y aquellos generados por el algoritmo evolutivo multiobjetivo NSGA-II, en donde se evidencia un rendimiento competitivo del algoritmo propuesto en términos de convergencia y diversidad.

**Palabras clave:** algoritmos evolutivos, ciudades inteligentes, metaheurísticas, optimización multiobjetivo, transporte público.

Abstract of the thesis presented by **David Peña Morales** as a partial requirement to obtain the Master of Science degree in Computer Science

### **Multiobjective optimization for public transport planning using metaheuristic**

Abstract approved by:

---

Dr. Andrey Chernykh  
Thesis Director

Public transport is an important topic for the cities development, because it affects directly the people's welfare when they move for working, studying, health care or entertainment. The quality of service must guarantee a good experience in terms such as comfort, safety, availability and shorter waiting time. On the other hand, the transport companies look for reducing the operational cost (e.g. fuel, maintenance or driver). Therefore, for a given passenger demand an appropriate fleet assignment can reduce the flux of empty vehicles, fuel consumption and the associated costs, helping to reduce traffic jams, negative environmental impact and operation costs. Timetabling and vehicle scheduling are important activities in public transport planning. The timetables depend on the study and optimization in vehicles' frequency; hence, they are important to avoid additional costs and improve the quality of service perceived by users, which is essential to making public transport more attractive, and to increase its use compared to private transport methods. This paper presents a heuristic based on MOCeII (a cellular genetic algorithm for multiobjective optimization) to create bus timetables using multiple vehicle types (i.e. different capacity and operating cost), considering restrictions of government agencies for public transport systems in the context of smart cities, also known as Intelligent Transport System (ITS). The conflicting objectives tackled in this work are the minimization of the overall cost, that affects the operator's business and the minimization of the number of passengers that cannot move in a time-period, which suppose a reduction in vehicles overload and less waiting time. A set of non-dominated solutions represents different assignments of vehicles covering a given set of trips in a defined route, this document includes the experimental analysis and the comparison between the results and the non-dominated sets provided by a multiobjective evolutionary algorithm known as NSGA-II, this study shows a competitive performance of the proposed algorithm in terms of convergence and diversity.

**Keywords: evolutionary algorithms, metaheuristics, multiobjective optimization, public transportation, smart cities.**

Dedicatoria

*A Colombia,  
Robinson y Alexander*

*“Education is the passport to the future, for tomorrow  
belongs to those who prepare for it today”*

—Malcolm X

## Agradecimientos

A mis padres y hermano, por su apoyo incondicional desde la distancia y ser una motivación para alcanzar grandes logros.

Al Dr. Andrey Chernykh, por su apoyo y acompañamiento fundamental para elaborar el trabajo de tesis.

A los miembros de mi comité de tesis, por sus consejos, indicaciones y colaboración en este trabajo.

A mis amigos que estuvieron apoyándome, tolerándome y aconsejándome durante las largas sesiones de estudio.

Al Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE), en especial a todo el Departamento de Ciencias de la Computación (DCC) que me ha brindado un gran apoyo en mi estancia en la institución.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de maestría.

# Tabla de contenido

Resumen .....	ii
Abstract .....	iii
Dedicatoria .....	iv
Agradecimientos .....	v
Lista de figuras .....	viii
Lista de tablas .....	xi
<b>Capítulo 1. Introducción .....</b>	<b>1</b>
<b>1.1 Motivación.....</b>	<b>1</b>
<b>1.2 Antecedentes.....</b>	<b>3</b>
1.2.1 Heurísticas Tempranas .....	5
1.2.2 Métodos Matemáticos. ....	6
1.2.3 Métodos Metaheurísticos .....	6
<b>1.3 Objetivos .....</b>	<b>11</b>
1.3.1 Objetivo general .....	11
1.3.2 Objetivos específicos .....	11
<b>1.4 Organización de la Tesis .....</b>	<b>11</b>
<b>Capítulo 2. Transporte público.....</b>	<b>13</b>
<b>2.1 Introducción .....</b>	<b>13</b>
<b>2.2 Creación de horarios o calendarios.....</b>	<b>15</b>
<b>2.3 Conceptos básicos.....</b>	<b>16</b>
<b>2.4 Creación de horarios o calendarios con vehículos de diferentes tipos.....</b>	<b>18</b>
<b>2.5 Trabajos relacionados .....</b>	<b>18</b>
<b>2.6 Definición del problema.....</b>	<b>20</b>
2.6.1 Planteamiento del problema .....	20
2.6.2 Definición formal del problema .....	21
<b>Capítulo 3. Optimización multiobjetivo (OMO) .....</b>	<b>23</b>
<b>3.1 Introducción .....</b>	<b>23</b>
<b>3.2 Conceptos básicos de OMO.....</b>	<b>23</b>
3.2.1 Vector ideal.....	26
3.2.2 Conceptos de dominancia y Pareto .....	27
<b>3.3 Métodos de solución.....</b>	<b>31</b>
3.3.1 Clasificación de los métodos para POMO.....	32
3.3.2 Método del criterio global (método sin preferencia) .....	33
3.3.3 Ordenamiento lexicográfico (método <i>a priori</i> ).....	34
3.3.4 Restricción $-\epsilon$ (método <i>a posteriori</i> ) .....	34
3.3.5 Optimización evolutiva multiobjetivo (método <i>a posteriori</i> ) .....	35
<b>Capítulo 4. Algoritmos evolutivos para problemas de OMO.....</b>	<b>37</b>
<b>4.1 Introducción .....</b>	<b>37</b>
<b>4.2 Conceptos básicos de Algoritmos evolutivos (AE).....</b>	<b>38</b>
4.2.1 Componentes de un AE .....	40
4.2.2 Representación del individuo .....	42
4.2.3 Población .....	43



4.2.4	Función para evaluación de aptitud (función de calidad).....	44
4.2.5	Mecanismo para la selección de padres.....	44
4.2.6	Recombinación .....	44
4.2.7	Mutación .....	46
4.2.8	Mecanismo de selección de los sobrevivientes (Reemplazo).....	47
<b>4.3</b>	<b>AE con enfoque multiobjetivo (AEMO).....</b>	<b>47</b>
4.3.1	Clasificación de los AEMO.....	48
4.3.2	Nondominated Sorting Genetic Algorithm II (NSGA-II) .....	49
4.3.3	MultiObjective Cellular genetic algorithm (MOCeII) .....	51
<b>4.4</b>	<b>AEMO propuesto .....</b>	<b>53</b>
4.4.1	Representación (codificación) .....	53
4.4.2	Funciones objetivo y evaluación de aptitud .....	55
4.4.3	Operadores genéticos.....	56
<b>Capítulo 5.</b>	<b>Resultados experimentales .....</b>	<b>58</b>
<b>5.1</b>	<b>Indicadores de calidad (métricas de rendimiento).....</b>	<b>59</b>
5.1.1	Espaciado y propagación .....	59
5.1.2	Hipervolumen .....	61
5.1.3	Cubrimiento entre dos conjuntos.....	62
5.1.4	Indicador $-\epsilon$ .....	63
<b>5.2</b>	<b>Diseño de experimentos .....</b>	<b>65</b>
<b>5.3</b>	<b>Resultados experimentales .....</b>	<b>68</b>
<b>5.4</b>	<b>Análisis de resultados y discusión.....</b>	<b>73</b>
<b>Capítulo 6.</b>	<b>Conclusiones y trabajo futuro .....</b>	<b>76</b>
<b>Literatura citada.....</b>		<b>78</b>

## Lista de figuras

<b>Figura 1.</b> Relaciones importantes entre entidades comprendidas en ciudades inteligentes con los principales retos involucrados.....	2
<b>Figura 2.</b> Línea de tiempo de la evolución en el ámbito de la investigación de operaciones sobre el estudio de problemas de optimización asociados a sistemas de transporte. ....	4
<b>Figura 3.</b> Portada y una página de “ <i>Bus Scheduling Manual</i> ” Agosto de 1947 (Ceder, 2007). ....	4
<b>Figura 4.</b> Trazado de la ruta y sus respectivas paradas definidas al momento del diseño topológico de R (izquierda). Tipos vehículos disponibles, calendario propuesto que asigna cada vehículo (i) a un viaje (tk) y su comportamiento para una demanda de pasajeros obtenida en un periodo de tiempo j (derecha).....	22
<b>Figura 5.</b> Representación gráfica del espacio de decisión ( $\Omega$ ) y su correspondiente espacio objetivo ( $\Lambda$ ). ....	25
<b>Figura 6.</b> Soluciones óptimas de Pareto (frente de Pareto) en el espacio objetivo ( $\Lambda$ ), para los cuatro casos posibles con dos funciones objetivo (f1 y f2). a). Minimización-Minimización. b). Minimización-Maximización. c). Maximización-Minimización. d). Maximización-Maximización.....	29
<b>Figura 7.</b> Vector objetivo ideal ( $z^*$ ) y ficticio ( $z^{**}$ ), frente de Pareto (PF) y dos posibles conjuntos de soluciones P1, y P4 (izquierda). Asimismo, tres vectores objetivo diferentes (P1, P2, P3) y su respectivo PF (derecha), para dos POMOs de minimización con dos funciones objetivo f1 y f2. ....	30
<b>Figura 8.</b> Componentes principales de un genotipo que define los rasgos de un organismo, manifestados, según el ambiente al que está expuesto, como: colores, formas, y demás características visibles del individuo (fenotipo). ....	39
<b>Figura 9.</b> Ejemplo de una representación binaria para un conjunto de individuos con diferentes características. Analogía entre los conceptos computacionales y su contraparte biológica, estructura de los datos y terminología. ....	39
<b>Figura 10.</b> Ciclo evolutivo tradicional para un AG con población panmíctica.....	41
<b>Figura 11.</b> “ <i>One-point crossover</i> ” (arriba) con un $\rho = 3$ , “ <i>n-point crossover</i> ” (medio) para un $n = 3$ , $\rho = 1,3,5$ y cruzamiento uniforme (abajo) para un arreglo de números aleatorios = 0.1, 0.7, 0.8, 0.2, 0.9, 0.8, 0.2 y $\rho = 0.5$ . ....	45
<b>Figura 12.</b> Mutación “ <i>swap</i> ” (arriba) para un $\kappa = 2,3,7$ , “ <i>random resetting</i> ” (abajo) para un cromosoma de con representación binaria.....	46

<b>Figura 13.</b> Cuadro esquemático de los pasos que ejecuta NSGA-II para producir una nueva generación $P_i + 1$ . .....	50
<b>Figura 14.</b> Cálculo de la distancia de apiñamiento (“ <i>crowding distance</i> ”), para los valores extremos del frente se asigna una distancia infinita. ....	51
<b>Figura 15.</b> Diferentes tipos de poblaciones utilizadas en algoritmos evolutivos a) Panmíctica, b) Islas conectadas, c) Vecindarios superpuestos. ....	51
<b>Figura 16.</b> Vecindarios diferentes, a) cuatro, b) ocho, c) doce, para dos individuos $I_1 = (x, y)$ e $I_2 = (x-1, y-1)$ . ....	52
<b>Figura 17.</b> Ciclo de reproducción para un algoritmo genético celular con cuatro vecinos por individuo y un conjunto no dominado de cuatro individuos. ....	52
<b>Figura 18.</b> Representación para dos individuos vecinos y su respectivo comportamiento para un periodo de tiempo y una carga de pajaros definida. ....	54
<b>Figura 19.</b> Perfil de carga para una ruta con seis paradas si, $i = 1, \dots, 6$ , una carga máxima de 250 pasajeros en la tercera parada, una capacidad de vehículos total igual a 200 y un factor de carga $LF_2 = 75\%$ . $LQ_3 = 100$ indica que en la tercera parada se están perdiendo cien pasajeros ya que no pueden ser transportados por la flota asignada. ....	55
<b>Figura 20.</b> Método de recombinación implementado, modificación del tradicional “ <i>one-point crossover</i> ” .....	56
<b>Figura 21.</b> Método de mutación implementado, con un $k = 3$ (cantidad de genes que pueden ser modificados). ....	57
<b>Figura 22.</b> Ejemplo de la representación de una solución (cromosoma) para un caso con tres tipos de vehículos diferentes y un total de seis viajes en tres periodos de tiempo. Etapas de reproducción para un algoritmo celular asíncrono. ....	57
<b>Figura 23.</b> Distancias entre las soluciones de un P. ....	60
<b>Figura 24.</b> Distancias entre las soluciones de un P, incluidas las distancias entre los extremos del frente de Pareto y las soluciones más cercanas. ....	61
<b>Figura 25.</b> El hipervolumen encerrado por las soluciones que pertenecen al conjunto de soluciones no dominado $P_1$ y el punto W (izquierda). Comparación entre el hipervolumen construido por $P_1$ y $P_2$ (derecha). ....	62
<b>Figura 26.</b> Ejemplo de la interacción de dos conjuntos de soluciones $P_1$ y $P_2$ y sus posibles relaciones de dominancia. ....	63
<b>Figura 27.</b> Cada una de las áreas marcadas representa el espacio que es $\epsilon$ -dominado al calcular la métrica así: $I_{\epsilon P_1, P_2} = 0.88$ , $I_{\epsilon P_1, P_3} = 1$ , $I_{\epsilon P_1, P_f} = 4$	

.Multiplicando a la solución P1 los valores obtenidos para $I_e$ se obtienen las áreas que representan cada uno de los tres posibles casos de $\epsilon$ . .....	64
<b>Figura 28.</b> Mapa digital de la ciudad de Los Ángeles, California EEUU con las paradas de la ruta 217 resaltadas en sentido sur-norte y norte-sur (Esidian, 2017). .....	66
<b>Figura 29.</b> Carga de prueba utilizada para la ejecución de los algoritmos, representa el promedio de la demanda de pasajeros por cada una de las 59 paradas en intervalos de tiempo de una hora (19 intervalos en total) para la ruta 217 en LA, carga máxima de pasajeros (481) en la parada Fairfax/ Rosewood durante las 17:00 a las 18:00 horas (hora pico). .....	66
<b>Figura 30.</b> Mapa de calor para el rendimiento promedio (IHV) de los resultados del algoritmo genético celular basado en MOCeII en su versión asíncrona. ....	68
<b>Figura 31.</b> Degradación del promedio de IHV para las 342 combinaciones de parámetros para el algoritmo genético celular ( $P_r$ y $P_m$ ). .....	69
<b>Figura 32.</b> Impacto de la variación de la probabilidad de recombinación con un mismo valor de $P_m$ . .....	69
<b>Figura 33.</b> Rendimiento del hipervolumen en 30 ejecuciones independientes del algoritmo genético celular para 100 generaciones, es decir, evaluar la malla toroidal completa, individuo por individuo. ....	70
<b>Figura 34.</b> Mapa de calor para el rendimiento promedio (IHV) de los resultados del algoritmo genético NSGA-II. ....	71
<b>Figura 35.</b> Conjunto de soluciones obtenido con el algoritmo genético celular, correspondiente a un $IHV = 0.4398$ en comparación a la población inicial para una ejecución con $P_r = 0,95$ y $P_m = 0,3$ . .....	71
<b>Figura 36.</b> Mejor aproximación al frente de Pareto obtenida para MOCeII y NSGAII, correspondiente a un $IHV= 0.4398$ y $0.3866$ respectivamente (izquierda). Mejor aproximación al frente de Pareto obtenida, mediante la unión de las 30 soluciones encontradas en cada ejecución de los algoritmos (derecha). ....	72
<b>Figura 37.</b> Comparación entre un frente de Pareto artificial y dos conjuntos de soluciones no dominadas. ....	74
<b>Figura 38.</b> Ejemplo de tres calendarios diferentes (fenotipo) al interpretar tres cromosomas ubicados en el espacio objetivo (izquierda). Comparación de los tres horarios encontrados versus el costo de operación de una asignación vehicular observada en la ruta 217 en Los Ángeles, California EEUU. ....	75

## Lista de tablas

<b>Tabla 1.</b> Relaciones de dominancia entre dos soluciones (puntos en el espacio objetivo) y entre dos conjuntos de aproximación al frente de Pareto. ....	30
<b>Tabla 2.</b> Resumen general de las métricas binarias <b>Indicador-<math>\epsilon</math></b> y <b>Cubrimiento entre dos conjuntos</b> en términos de las relaciones de dominancia. Los espacios vacíos indican que no hay un método de comparación que permita definir que es completamente compatible con esta relación de dominancia. ....	65
<b>Tabla 3.</b> Parámetros de configuración iniciales para el algoritmo genético celular. ....	67
<b>Tabla 4.</b> Parámetros de configuración iniciales para el algoritmo genético NSGA-II. ....	70
<b>Tabla 5.</b> Métricas para POMO como indicadores de calidad calculadas para AGC y NSGA-II. ....	72
<b>Tabla 6.</b> Comparación de resultados aplicando la métrica $ITSCX', X''$ donde $X'$ y $X''$ representan el algoritmo en la fila y en la columna respectivamente. ....	72
<b>Tabla 7.</b> Comparación de resultados aplicando la métrica $I \in A, B$ donde A y B representan el algoritmo en la fila y en la columna respectivamente. ....	73

## Capítulo 1. Introducción

---

En un mundo cada vez más interconectado y globalizado, en 2014 más de la mitad de los habitantes (54%) se encuentra en zonas metropolitanas, a diferencia del 30% estimado en 1950. Este abrupto crecimiento implica cambios importantes tanto en el tamaño como en la densidad de población, este efecto se acentuará en las próximas décadas, con un crecimiento estimado del 70% de la población mundial viviendo en las ciudades en el 2050 (United Nations, 2014), lo cual conlleva a un incremento de la demanda de las infraestructuras que interactúan directamente con las personas que suelen moverse a las áreas urbanas en busca de oportunidades y una calidad de vida mejor, lo que implica diversos problemas como la congestión, aumentando la utilización de una cantidad limitada de recursos naturales y otros bienes o servicios como: la energía, el agua, la educación, los servicios de salud o el transporte.

### 1.1 Motivación

Actualmente, las principales dificultades para las ciudades en términos de movilidad urbana suelen relacionarse con la incapacidad de los sistemas de transporte público para satisfacer las necesidades de un número creciente de usuarios y aunque cada ciudad tiene problemáticas adicionales diferentes, las autoridades o entes responsables de la movilidad comparten retos comunes como (Houghton et al., 2011):

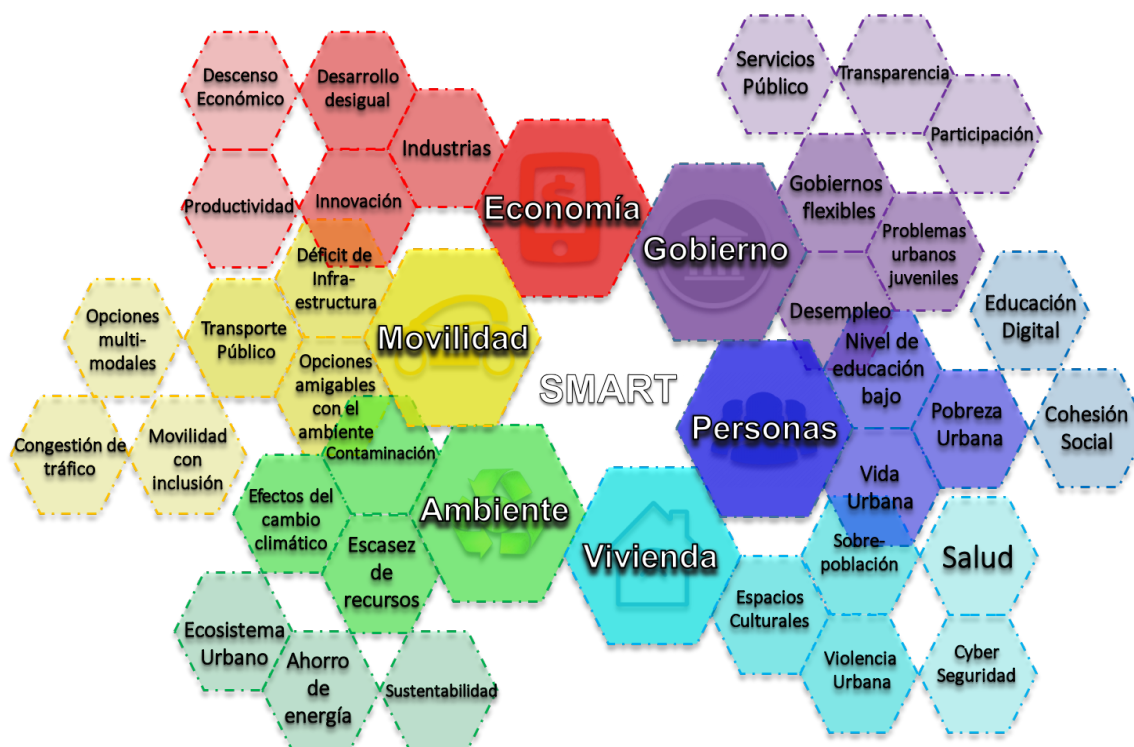
- Reducir la congestión vehicular mejorando el flujo del tráfico, concibiendo un ambiente sustentable y más limpio.
- Incentivar el uso del transporte público y otras opciones de desplazamiento más ecológicas como la bicicleta.

Adicionalmente, es importante impulsar una reducción en el uso del vehículo particular, el cual en promedio transporta menos de dos personas, ocupando el mismo espacio para trasladar a cinco pasajeros; generando más polución y agravando el efecto invernadero, este uso desmedido del espacio público aumenta el tráfico, perjudicando notablemente el tiempo desplazamiento de los ciudadanos.

Ante este panorama, se hace necesario el desarrollo de metodologías para el diseño y planificación de los sistemas de transporte público que brinden un servicio de calidad a los ciudadanos.

*"Una ciudad avanzada no es en la que los pobres pueden moverse en carro, sino una en la que incluso los ricos utilizan el transporte público"* (Peñalosa, 2013).

Las ciudades inteligentes ("Smart Cities") proponen la incorporación de tecnologías para la cohesión de los aspectos que involucran problemas de tipo social, institucional e infraestructura en un ecosistema urbano (Albino et al., 2015). Áreas tan diversas como la administración ciudadana, educación, servicios de salud, seguridad pública, vivienda, energía, transporte y logística, pueden mejorarse, interconectarse y tornarse más eficientes gracias a la incorporación de tecnología (Washburn y Sindhu, 2010). Las ciudades inteligentes permiten reducir costos, hacer un uso responsable de los recursos y fomentar la participación activa de los ciudadanos en los procesos de toma de decisiones, con el fin de alcanzar una ciudad sustentable e inclusiva (ver **Figura 1**).



**Figura 1.** Relaciones importantes entre entidades comprendidas en ciudades inteligentes con los principales retos involucrados.

Para el caso puntual del transporte, se introduce el concepto de “*Smart Mobility*” o sistema de transporte inteligente (**ITS**: por sus siglas en inglés “*Intelligent Transportation System*”) en el que se integran tecnologías de la información y la comunicación (TIC), aplicadas al sector del transporte (Alam et al., 2016).

Los avances tecnológicos, en especial en ciencias computacionales y sistemas de comunicación, permiten recolectar un gran volumen de datos del transporte y la movilidad de las personas en las ciudades, de los cuales se puede extraer información relevante, la cual compromete a los ITS con los usuarios, organizaciones gubernamentales y operadores de diferentes servicios a contar con nuevas herramientas y aplicaciones producto del arduo trabajo de la comunidad científica.

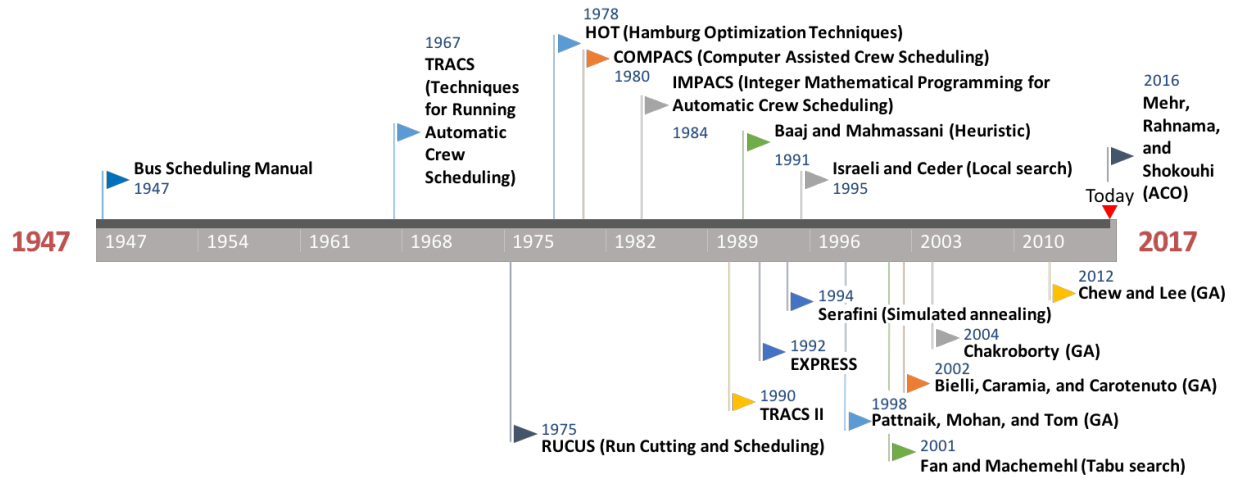
Los objetivos de los ITS, generalmente se formulan, mediante la parametrización de metas establecidas por entes interesados (Gordon, 2016):

- Mejorar la seguridad de los sistemas de transporte.
- Incrementar la eficiencia operativa y capacidad de los sistemas de transporte.
- Reducir el consumo de energía y el impacto ambiental.
- Mejorar la productividad económica tanto de usuarios como de operadores de los sistemas de transporte.
- Mejorar la movilidad individual, la convivencia y comodidad para los pasajeros en los sistemas de transporte
- Crear un entorno que permita el desarrollo y despliegue de nuevas tecnologías de ITS.

## 1.2 Antecedentes

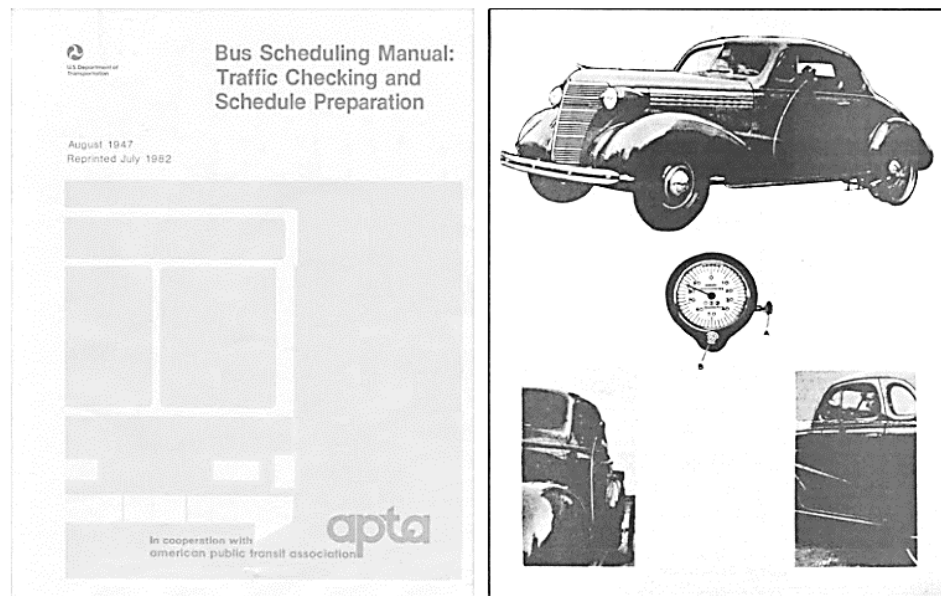
Diferentes modelos y algoritmos han sido desarrollados para resolver problemas relacionados con el transporte, en especial transporte público urbano. La mayoría de estos trabajos se basan en modelos planteados hace algunas décadas, los cuales son abordados con técnicas de inteligencia computacional, buscando aproximar mejor las posibles soluciones, debido a la complejidad computacional que pueden presentar algunos sub-problemas como la creación de horarios o la calendarización de los vehículos. En la **Figura 2** se muestra una línea de tiempo de los esfuerzos que se están realizando en este campo de investigación.





**Figura 2.** Línea de tiempo de la evolución en el ámbito de la investigación de operaciones sobre el estudio de problemas de optimización asociados a sistemas de transporte.

Se han propuesto diferentes algoritmos para la solución a las problemáticas de planificación de rutas, calendarización de conductores, asignación de vehículos (tamaño de la flota requerida) y asignación de frecuencias de despacho para viajes en una ruta específica. Este conjunto de problemáticas se puede englobar en el problema de planificación del transporte público, el cual a través de la historia ha tenido una evolución que surge desde la década del 40 (ver **Figura 3**) en diferentes lugares (Ceder, 2007), en donde manuales para la calendarización de buses hacían una aparición en un sistema que requería mayor detalle para una comprensión completa de los problemas asociados .



**Figura 3.** Portada y una página de "Bus Scheduling Manual" Agosto de 1947 (Ceder, 2007).

En los años 70 aparecen nuevas ideas basadas en la intuición, pero aún no se contaba con un modelo concreto, en la década de los 80 aparecen formulaciones para algunas funciones objetivo (p.ej. reducir el tiempo de espera o minimizar el costo de operación) y algunos parámetros para el cubrimiento de la demanda; luego a inicios de los 90 aparecen nuevos enfoques ligados a la exploración del espacio de soluciones y la utilización de modelos estocásticos (Ramírez et al., 2015).

Para realizar la planificación de un sistema de transporte público es necesario solucionar todos los problemas relacionados, algunos autores (Ceder, 2016; Ibarra-Rojas et al., 2015) proponen una serie de pasos o fases de la siguiente manera:

- Diseño de recorridos para las rutas.
- Definición de frecuencias de los vehículos.
- Determinación de la tabla de horarios, hora de salida y hora de llegada.
- Asignación de vehículos para cubrir los viajes.
- Asignación de los conductores y demás personal requerido para la operación.

Todas las etapas deben ser tratadas simultáneamente para asegurar mejores resultados, sin embargo, debido a la complejidad de desarrollar un sistema completo, se definen una serie de sub-problemas, los cuales se desarrollan de manera individual y secuencial, no obstante, comparten características que pueden ser comparadas y examinadas en detalle.

### 1.2.1 Heurísticas Tempranas

Llamaremos heurísticas tempranas a las primeras aproximaciones al problema, las cuales no tenían algún modelo matemático desarrollado y se basaban en buscar una solución inicial e ir refinándola al máximo por exploración con algún método no riguroso.

El sistema TRACS (“Techniques for Running Automatic Crew Scheduling”) fue desarrollado en la Universidad de Leeds en 1967, como uno de los primeros esquemas propuestos el cual se enfocaba en generar la mejor solución inicial posible, debido a que no consideraba la posibilidad de hacer modificaciones con buen rendimiento usando las técnicas disponibles como el tanteo; TRACS generaba el calendario inicial para la asignación de conductores mediante la combinación de distribuciones que cubren

las horas de mayor demanda, generando pequeños bloques que podían ser eliminados ya que se superponían con la actividad de otro conductor (J. Li, 2002). En los 70's surge el sistema RUCUS ("Run Cutting and Scheduling"), en Hamburgo se propone el HOT ("Hamburg Optimization Techniques") los cuales generaban alguna posible solución inicial la cual se explotaba con una técnica de búsqueda local buscando mejorarla. En la década de los ochenta se incorpora un paquete de programación conocido como BUSMAN, el cual contenía un algoritmo de nombre COMPACS ("Computer Assisted Crew Scheduling") el cual utilizaba las soluciones iniciales generadas por TRACS para luego modificarlas en busca de mejoras mediante la interacción entre la solución inicial y las posibles variaciones (Chamberlain y Wren, 1992).

### 1.2.2 Métodos Matemáticos.

El problema de asignación de autobuses o calendarización de conductores puede modelarse como un cubrimiento de conjuntos ("*set covering*") (Chvatal, 1979) o partición de conjuntos ("*set partitioning*") (Bianco et al., 1994), modelos matemáticos de grafos ampliamente conocidos.

A finales de los años 70 se desarrolló IMPACS ("*Integer Mathematical Programming for Automatic Crew Scheduling*") instalado en Londres en 1984. Luego apareció TRACS II como una nueva generación de software para programación de conductores basado en programación matemática que garantiza un calendario óptimo (Fores et al., 1999). En 1992 se presentó EXPRESS un sistema desarrollado para Nueva Zelanda fundamentado en el modelo de partición de conjuntos (J. Li, 2002).

Dado que la cantidad de combinaciones es muy alta y estos problemas son de tipo NP-difícil (Ceder, 2007), se propusieron implementaciones en combinación con heurísticas para reducir su dimensionalidad y poder encontrar soluciones factibles en tiempos razonables.

### 1.2.3 Métodos Metaheurísticos

Varios investigadores utilizan diferentes técnicas de inteligencia computacional combinadas con métodos propios de la investigación de operaciones para tratar de encontrar soluciones óptimas a los problemas relacionados con transporte como la planificación del transporte público. Esto se hace necesario en vista de que la programación de frecuencias para despacho de vehículos o la asignación de conductores son problemas combinatorios, donde las variables de decisión deben definirse dentro de un espacio discreto

y el número de alternativas es equivalente a todas las combinaciones posibles de todos los valores de las variables involucradas.

Se han realizado numerosas propuestas de métodos de optimización para resolverlos, los cuales pueden clasificarse en exactos y heurísticos. Los exactos garantizan encontrar el óptimo global, pero hay muy pocas aplicaciones en la literatura, debido a su ineficiencia por la alta dimensionalidad que debe manejarse. Los heurísticos o metaheurísticos solo permiten aproximarse a la solución óptima (óptimo global), pero son más eficientes y flexibles con los recursos necesarios para su ejecución. Los métodos heurísticos siguen en general una de las dos estrategias básicas: “divide y vencerás” y mejora iterativa. Los primeros, fraccionan apropiadamente el problema en sub-problemas de tamaño más “manejable” y luego enlazan las soluciones encontradas para cada uno de ellos. Los algoritmos de mejora iterativa inician con una posible solución al problema la cual de manera continua es modificada hasta que se mejore la función objetivo. Esta búsqueda puede estancarse en un valor óptimo local por lo que es recomendable hacer el proceso varias veces utilizando diferentes configuraciones generadas aleatoriamente, guardando los mejores resultados. Adicional a la complejidad antes mencionada, hay diversos problemas con múltiples objetivos que pueden estar en conflicto lo cual aumenta el costo y la dificultad computacional a la hora de intentar resolverlos. En el caso del transporte público, como se pudo observar en la **Figura 3**, se han implementado diferentes métodos heurísticos, metaheurísticos y exactos; algunos de ellos basados en el aporte realizado por Israeli, Ceder, Baaj y Mahmassani.

Baaj y Mahmassani (1995) proponen varias restricciones al problema de planificación de transporte con parámetros que tienen en cuenta diversos aspectos importantes como la frecuencia de los vehículos, el tamaño de la flota y el factor de carga de los pasajeros. El modelo está definido como:

$$\text{Minimizar} \quad [C_1 \sum_{i=1}^n \sum_{j=1}^n d_{ij} t_{ij} + C_2 \sum_{k=1}^n f_k t_k]$$

$$\text{Sujeto a} \quad f_k \geq f_{min} \quad \forall k \in R \quad (1)$$

$$LF_k = \frac{Q_k^{max}}{f_k \alpha} \leq LF_{max} \quad \forall k \in R \quad (2)$$

$$\sum_{k \in R} N_k = \sum_{k \in R} f_k t_k \leq FS \quad (3)$$

Donde:

- $n$  Cantidad de nodos de la red.
- $d_{ij}$  Demanda entre los nodos  $i$  y  $j$ .
- $t_{ij}$  Tiempo de viaje entre  $i$  y  $j$ .

$f_k$	Frecuencia de los vehículos que operan la ruta $k$ .
$f_{min}$	Frecuencia mínima permitida para cualquier ruta.
$t_k$	Tiempo total de viaje de la ruta $k$ .
$Q_k^{max}$	Máximo flujo por arco de la ruta $k$ .
$\alpha$	Capacidad de pasajeros en los vehículos.
$LF_k$	Factor de carga en la ruta.
$LF_{max}$	Factor de carga máximo permitido.
$N_k$	Cantidad de buses operando la ruta $k$ .
$FS$	Tamaño de la flota disponible.
$R$	Conjunto de rutas.
$C_1$ y $C_2$	Pesos relativos para la función a minimizar.

Para este modelo se propone una solución heurística basado en la combinación de rutas, en el cual la población inicial se genera a partir de identificar los caminos más cortos entre los nodos de alta demanda.

Por otra parte, se propusieron algoritmos de búsqueda local, para un modelo de optimización multiobjetivo, en donde se pretende minimizar el tamaño de la flota y reducir el costo que representa la cantidad de pasajeros por hora, tiempo de espera de los pasajeros entre cada parada y tiempo de viaje en el que el bus se encuentra vacío (Israeli y Ceder, 1995).

$$\text{Minimizar: } f_1 = \left[ C_1 \sum_{i,j \in N} PH_{ij} + C_2 \sum_{i,j \in N} WH_{ij} + C_3 \sum_{r \in R} EH_r \right] \quad (4)$$

$$\text{y } f_2 = FS \quad (5)$$

Donde:

$PH_{ij}$	Pasajeros/hora entre los nodos $i$ y $j$ .
$WH_{ij}$	Tiempo de espera de los pasajeros entre los nodos $i$ y $j$ .
$EH_{ij}$	Espacio vacío por hora en la ruta $r$ .
$FS$	Tamaño de la flota disponible.
$R$	Conjunto de rutas.
$C_1, C_2$ y $C_3$	Pesos relativos para la función $f_1$ a minimizar.

El modelo se resuelve en tres momentos:

- Se generan varios conjuntos de soluciones no dominados, mediante la solución al problema asociado de cubrimiento de conjuntos.
- Se determina las frecuencias de despacho de cada vehículo.
- Para la exploración de soluciones, se implementa un método de búsqueda local, finalmente se evalúan y seleccionan las mejores soluciones del conjunto Pareto-óptimo.

La técnica conocida como “recocido simulado” se caracteriza por alterar una solución factible seleccionada aleatoriamente. Esta perturbación producirá una nueva alternativa que sustituirá a la anterior. El reemplazo será aceptado o no en función de alguna regla de decisión. El proceso se repite iterativamente hasta que se considere que se llegó al óptimo o con una aproximación aceptable (Kirkpatrick et al. , 1983). El método se basa en el "recocido simulado" realizado en mecánica estadística, La técnica consiste en calentar y luego enfriar el material como el acero o las cerámicas, variando sus características, el calor promueve un cambio en la energía de los átomos alcanzando los valores iniciales (mínimo local del problema), luego el enfriamiento controlado pretende alcanzar los valores cuyo rendimiento tenga menor energía que los valores iniciales (mínimo global del problema). Serafini (1994) utiliza algoritmos basados en “recocido simulado” y sugiere su uso en problemas con múltiples objetivos; empleando reglas de agregación ponderada de valores de los objetivos, diferente a los algoritmos clásicos de recocido simulado para un solo objetivo. Costa et al. (2016), presentan un algoritmo en el campo de los trenes de alta velocidad, inicializado de manera aleatoria y utilizando técnicas clásicas de enfriamiento para mejorar la solución y escapar de óptimos locales.

Algoritmos basados en “Búsqueda Tabú”, un método introducido por Glover en 1989 como una técnica iterativa que escapa de óptimos locales aceptando soluciones no tan buenas, moviéndose paso a paso hacia una solución lo más cercana posible a un óptimo global (Glover et al., 2007). La búsqueda tabú, al contrario de otros métodos que tratan de escapar de los óptimos locales aleatoriamente, utiliza una exploración inteligente basada en formas sistemáticas, esta metaheurística está basada en la premisa de que para clasificar un procedimiento de solución como “inteligente”, es necesario que éste incorpore memoria y exploración adaptativa. El sistema HACS (Shen y Kwan, 2001) aplica búsqueda tabú al problema de la asignación de conductores. Un método de búsqueda tabú fue aplicado para el diseño de redes de transporte con demanda variable basado en el trabajo de Baaj y Mahmassani (1995) con enfoque no lineal y multiobjetivo (Fan y Machemehl, 2008).

Los métodos conocidos como *optimización por colonias de hormigas* se inspiraron en los estudios sobre el comportamiento de las colonias de hormigas y como logran determinar caminos cortos hacia fuentes de alimentos a través de la búsqueda semi-aleatoria de un grupo de hormigas que depositan feromonas (Dorigo et al., 2006). La feromona depositada constituye la información que permite la coordinación de los esfuerzos del enjambre en busca de comida, una hormiga está más inclinada a seguir una trayectoria cuanto más fuerte sea el rastro de feromona depositada en ella. ¿Cómo ayuda esto a encontrar el camino más corto a la comida? Pues resulta que la hormiga que da con la comida primero (presumiblemente porque encontró un camino más o menos corto hacia el alimento) también es la primera en depositar el doble de feromona sobre una trayectoria al seguir ésta en su camino de vuelta a la colonia. Otras hormigas preferirán ligeramente esta trayectoria en comparación con otras que solo cuentan con la mitad de feromona en ellas, y al hacerlo intensifican el rastro de feromona haciendo la ruta aún más atractiva para nuevas hormigas. Como al principio el rastro de feromona no es muy fuerte, las primeras hormigas que siguen el camino hacia la comida pueden desviarse y encontrar atajos, con lo que se perfecciona la solución propuesta. Eventualmente la mayor parte de hormigas siguen el camino hasta agotar la fuente de alimento (Quintero, 2005). Mehr et al. (2016) implementan este tipo de metaheurística para resolver al problema de diseño de líneas para trenes ligeros o buses de tránsito rápido en la ciudad de Mashhad en Irán.

Una técnica muy utilizada son los algoritmos genéticos (**AG**), inspirados en los conceptos de la genética y la selección natural. Se han realizado varios aportes por diferentes investigadores para el diseño de redes de transporte (topología y trayectos de las rutas) y la creación de horarios o calendarios, desde mediados de los años 90, por ejemplo Pattnaik et al. (1998) se centran en minimizar el costo asociado tanto para el operador como para el usuario, con un AG, que consta de dos fases: primero se genera una población inicial de candidatos a ser soluciones factibles al problema, luego se hacen torneos típicos en AG como ejemplo de selección natural para poder mejorar los individuos, finalizando con un selecto grupo de posibles soluciones con un alto desempeño. Bielli et al. (2002) proponen un AG con una función objetivo que llaman *ff*, la cual se caracteriza por tener una metodología de cómputo que genera nuevas poblaciones, evaluadas de manera iterativa con enfoques multiobjetivo, con el fin de diseñar la mejor infraestructura de buses que satisfaga una demanda y oferta para determinados casos. Chakroborty (2003, 2004) menciona la alta efectividad de los algoritmos genéticos para solucionar el problema de diseño de redes de transporte urbano (UTNDP por sus siglas en inglés: "*Urban Transit Network Design Problem*"). Chew y Lee (2012) desarrollan un marco de trabajo ("**framework**") usando AG cuya finalidad es reducir el costo para los pasajeros, resolviendo el problema de diseño de rutas en tránsito urbano. Utilizan una estructura basada en permutación de enteros y operaciones clásicas de cruzamiento y mutación para dicha representación.

## 1.3 Objetivos

### 1.3.1 Objetivo general

Diseñar e implementar un algoritmo para la planificación táctica en sistemas de transporte público utilizando vehículos de diferentes tipos en el contexto de ciudades inteligentes; aplicando metaheurísticas para optimización combinatoria multiobjetivo, apoyando en la reducción de los costos operativos y la disminución de la pérdida de la calidad del servicio.

### 1.3.2 Objetivos específicos

- Analizar información relevante al transporte público (p.ej. comportamiento de la demanda de pasajeros), que permita detectar la problemática a desarrollar.
- Modelar el problema con un enfoque de optimización combinatoria (función(es) de costo y representación del individuo), aplicando métodos estocásticos o metaheurísticas (p.ej. algoritmos genéticos).
- Diseñar e implementar un algoritmo que entregue una solución al problema propuesto con enfoque multiobjetivo.
- Encontrar los parámetros y las restricciones adecuadas que permitan al algoritmo entregar una solución(es) al problema capaz de minimizar el costo de operación, disminuyendo la pérdida la calidad del servicio, considerando las normas establecidas por las autoridades competentes.

## 1.4 Organización de la Tesis

Este trabajo está dividido en capítulos que desglosan la investigación desarrollada. En el Capítulo 2 se presentan algunos detalles e información relevante al transporte público y su planificación, así como una serie de conceptos básicos que se utilizarán a lo largo del documento, por último, se presenta la definición formal del problema planteado. En el Capítulo 3 se introducen los conceptos teóricos pertinentes a optimización, particularmente con múltiples objetivos y algunos métodos de solución para este tipo de



problemas. En el Capítulo 4 se hace énfasis a uno de los posibles métodos de solución para problemas con objetivos en conflicto conocido como algoritmos evolutivos, detallando su funcionamiento, características y componentes principales, finalizando con la presentación del algoritmo elegido para resolver el problema antes planteado. En el Capítulo 5 se presentan los resultados experimentales obtenidos y la metodología que se utilizó para obtenerlos, describiendo cada una de las métricas utilizadas como indicadores de calidad. Por último, se presentan algunas conclusiones y posible trabajo futuro en el área.

## Capítulo 2. Transporte público

---

### 2.1 Introducción

Es posible dividir un problema de planificación de transporte público (**PTP**) en cuatro etapas principales, ejecutadas habitualmente de manera secuencial, iniciando por la definición de la ruta, la creación de los horarios, la calendarización de los vehículos y por último la calendarización del personal a cargo del funcionamiento del sistema (Ceder, 2007, 2016). Por otra parte algunos autores sugieren hacer una división más específica fraccionando el problema de manera que incluyen decisiones tácticas, estratégicas y operacionales, manteniendo el concepto de dividir el problema en sub-problemas, en algunos casos con complejidad computacional NP-difícil, lo cual comprueba la dificultad de abordar un problema de PTP como un todo (Ceder, 2016; Hassold y Ceder, 2014; Ibarra-Rojas et al., 2015).

De forma genérica las fases necesarias para abordar un problema de PTP son:

- A. **Diseño de la ruta o red (planificación estratégica):** en esta fase se define la topología y estructura de las rutas, además del punto de partida y llegada, se definen la cantidad necesaria de paradas y la distancia entre las mismas, para lo que es necesario conocer la demanda de pasajeros. Esta fase está a cargo de las entidades reguladoras, buscando minimizar algunas funciones objetivo como el costo de operación, costos para los pasajeros y tiempos de viaje. Este tipo de procesos estratégicos tienen un horizonte temporal a largo plazo dado la complejidad de la infraestructura y demás implicaciones a la hora de entregar las líneas o rutas con sus respectivas paradas.
- B. **Determinación de la frecuencia (planificación táctica):** se busca determinar la frecuencia mínima de operación por periodos de tiempo basado en el estudio de la demanda de pasajeros de una ruta específica. Es importante tener en cuenta detalles como las horas pico y las horas valle, ya que marcan una serie de patrones característicos que enriquecen la planificación. Es posible que las entidades reguladoras a cargo realicen esta fase apoyados con los proveedores del servicio con el fin de reducir los costos de operación, los tiempos de viaje, los tiempos de espera priorizando un servicio de calidad para los usuarios. Debido a los cambios que se producen en cuanto al clima, festividades (diferentes en cada lugar), eventos, cambios en la demanda o algún acontecimiento, este tipo de procesos se definen a mediano plazo, monitoreando su funcionamiento para realizar los cambios pertinentes.

- C. **Construcción de horarios (planificación táctica):** en esta etapa se plasma la frecuencia deseada en una tabla de horarios que definen la hora de partida y de llegada de los vehículos, para cada parada y las terminales involucradas en cada ruta. Para resolver esta etapa del proceso los operadores del servicio, basados en las restricciones del ente regulador, definen el horario de cada vehículo de tal manera que satisfaga cierta demanda de pasajeros, maximizando el confort (menos tiempo de espera, menos sobrecarga o más seguridad). Usualmente el operador apoyado en la capacidad del vehículo y los patrones de la demanda construye los horarios a mediano y corto plazo debido a la volatilidad de la operación, donde averías en los vehículos, algún fallo en la red vial o imprevistos hacen necesario tomar medidas rápidas y precisas.
- D. **Asignación de la flota o calendarización de los vehículos (planificación operacional):** en esta fase los operadores del servicio buscan en gran medida minimizar el costo de operación, reutilizando al máximo los vehículos, cumpliendo los viajes estipulados en la fase anterior lo que implica cumplir con la frecuencia que exige la entidad reguladora que garantiza un servicio de calidad para el usuario. Al igual que en la construcción de horarios, esta fase tiene un horizonte temporal de mediano a corto plazo.
- E. **Asignación de los conductores o calendarización de los conductores (planificación operacional):** esta fase puede o no incluir al resto del personal necesario para la operación exitosa de un sistema de transporte público, sin embargo, en esencia se busca distribuir de manera que el costo total sea reducido al personal encargado a cada área, puntualmente los conductores son un ente importante para el buen funcionamiento del sistema, dado su alto nivel de interacción con el usuario final o pasajero. Lo anterior sin desconocer las implicaciones legales de cualquier trabajador, resaltando la importancia de no sobrecargar a los conductores y poner en riesgo no solo la operación del sistema de transporte sino las vidas e integridad de los pasajeros. Es una fase proyectada a mediano y corto plazo y es ejecutada exclusivamente por los operadores.

En este trabajo nos enfocaremos en la creación de los horarios de los vehículos (planificación táctica descrita en la sección 2.1.C), basado en un análisis de la frecuencia para un problema que incluye vehículos con diferentes capacidades, en este capítulo abordaremos con más detalle las dos etapas que se pretenden abordar (determinación de frecuencias y creación de horarios), definiendo algunos conceptos básicos en el campo del transporte público, posteriormente se introducirá las implicaciones de trabajar con la particularidad de contar con vehículos de diferentes tipos, finalizando con la definición detallada del problema que se desea resolver.

## 2.2 Creación de horarios o calendarios

La planeación o planificación táctica, es la encargada de la asignación o distribución de los recursos necesarios proyectados en la planificación estratégica, donde se definen las metas que se desean alcanzar. Debe ser un proceso constante y continuo que debe realizar el operador del servicio. Esta tarea se realiza a corto o mediano plazo, puesto que, al tomar un horizonte temporal más largo, se consideraría parte de la planificación estratégica (como el diseño del trazado de la ruta). Las decisiones tácticas tienen una implicación directa sobre el futuro de la operación, ya que establecen detalles sobre el comportamiento del sistema, tiempo de respuesta, efectividad y eficiencia; por lo tanto, deben ser establecidas mediante un proceso iterativo y flexible que pueda aceptar correcciones y ajustes durante su ejecución.

Para el caso específico del transporte público, las decisiones tácticas tienen una implicación directa sobre el incremento de la calidad del servicio y la reducción de los costos operacionales. Es necesario conocer el diseño topológico de la ruta y una matriz de demanda origen-destino o por paradas. En este orden de ideas, las problemáticas que se pueden abordar en esta etapa de la planificación de un sistema de transporte público son (Ibarra-Rojas et al., 2015):

- La frecuencia y flota (cantidad y tipo de vehículos) asignados para cubrir diferentes rutas en ciertos periodos de tiempo.
- La construcción de los horarios, que incluyan la frecuencia mínima de operación y demás restricciones que las entidades gubernamentales disponen para un funcionamiento con cierto nivel de calidad.
- Los posibles diseños de estrategias operacionales pueden ser considerados, en pro de aumentar el rendimiento de la planificación.

Por lo tanto, el diseño de tablas de horario debe satisfacer la demanda al mismo tiempo que cumple con restricciones como las regulaciones en temas de seguridad o la frecuencia de operación, contando con una serie de elementos como parámetros de entrada adicionales como: la ruta, tipo y capacidad de los vehículos, paradas, perfil de la demanda, factor de carga (cantidad deseada de pasajeros a bordo del vehículo) y tiempos de duración de la ruta.

Ceder (2007, 2016) propone los siguientes objetivos, relevantes a la hora de diseñar horarios como parte de la planificación táctica de un sistema de transporte:

- Evaluar las posibles tablas de horarios en términos de recursos requeridos.

- Mejorar el ajuste entre los tiempos de partida para cada viaje con la demanda de pasajeros, minimizando los recursos necesarios simultáneamente.
- Incluir excepciones para viajes que no dependen de datos de demanda de pasajeros, pero que tienen otras razones conocidas por el planificador.
- Proponer diferentes métodos de cálculo de la frecuencia de operación y para la creación de horarios.

Entonces es posible estructurar las dos fases, determinación de la frecuencia y creación de horarios, en una sola actividad conocida como el **desarrollo o construcción de horarios**.

## 2.3 Conceptos básicos

Algunos conceptos importantes pueden ser necesarios para comprender mejor el problema que se pretende abordar, algunos de estos son:

**Red de rutas (líneas):** una zona urbana con una demanda de pasajeros específica necesita una red o conjunto de rutas (paradas y terminales), esta red puede representarse como un grafo  $N(V, A)$ , donde los nodos  $V$  representan puntos o zonas geográficas y mientras que un conjunto de aristas  $(i, j) \in A$  representan un conjunto de posibles métodos de transporte que conecta el nodo  $i$  con el nodo  $j$ . Es posible diseñar el trazado de la ruta como un conjunto de aristas que conecten dos puntos de interés (terminal de salida y llegada), por otro lado, las paradas de la ruta pueden ser los lugares con una demanda potencialmente alta o que conecte con otros lugares de manera que se busque centralizar el lugar donde se desea ubicar la parada, esto debido a que entre menos paradas es posible que el desplazamiento sea más rápido.

**Frecuencia de operación:** la frecuencia está definida como el número de veces que se repite un evento periódico por unidad de tiempo, para el caso del transporte público, define la cantidad de vehículos que deben circular en cada lapso. La idea principal es poder garantizar el acomodamiento del mayor número de pasajeros a bordo en condiciones adecuadas en términos de confort y seguridad. De forma general y basado en la cantidad máxima de pasajeros en algún punto de la ruta  $R$  (hora pico), la frecuencia de operación ( $f_j$ ) está dada por:

$$f_j = \frac{P_j^{max}}{LF_j CAP_i}, i = 1, \dots, n \quad (6)$$

$$P_j^{max} = \max_{s \in R^s} P_j^s \quad (7)$$

Para cada periodo de tiempo  $j$  (p.ej. una hora) se calcula una frecuencia en base a la carga máxima  $P_j^{max}$ , que puede ser el máximo flujo de pasajeros en alguna parada  $s$  del conjunto de paradas  $R^S$ , en el tiempo  $j$ .  $CAP_i$  es la capacidad máxima de un vehículo de tipo  $i$ , es decir, la cantidad de sillas disponibles o la cantidad de pasajeros que pueden ser transportados, la define el fabricante en base a los criterios para los que fueron diseñados los  $n$  tipos de vehículos diferentes.  $LF_j$  (factor de carga) donde  $0 < LF_j \leq 1$ , representa proporción de la capacidad máxima de un vehículo permisible para la operación en el periodo de tiempo  $j$ .

**Frecuencia mínima de operación:** es común que por políticas gubernamentales se defina una frecuencia mínima de operación  $f_j^{min}$  de los vehículos durante el día, esté tipo de medidas afectan directamente a la definición de la frecuencia  $f_j$ , estableciendo la fórmula de la siguiente manera:

$$f_j = \max_{j=1, \dots, q} \left[ \frac{\bar{P}_j}{LF_j CAP_i}, f_j^{min} \right] \quad (8)$$

$\bar{P}_j$  es el promedio de pasajeros a lo largo de todas las paradas de  $R^S$  durante el periodo  $j$ .

**Intervalos:** denota los intervalos de tiempo entre cada viaje,  $H$  es el inverso de la frecuencia (cantidad de vehículos por unidad de tiempo).

$$H_j = \frac{1}{f_j} \quad (9)$$

**Tiempo de espera:** está asociado al tiempo que invierten los pasajeros en una parada  $s \in R^S$  esperando la llegada de un vehículo que pueda transportarlo a su destino. Aunque este concepto depende significativamente de los patrones de llegada de los pasajeros en cada parada, al conocerse los tiempos de salida y llegada de los buses a cada parada, el tiempo de espera estaría relacionado con el momento en el que el pasajero llega a la parada (antes, después o al tiempo), de manera que se puede dividir en tres grupos, los pasajeros que llegan a una parada de manera que coincidan con el bus, tienen una probabilidad  $\rho_c$ , los que llegan a la parada a la hora óptima  $(1 - \rho_c)\rho_o$  y los que llegan de manera aleatoria  $(1 - \rho_c)(1 - \rho_o)$  (Jolliffe y Hutchinson, 1975; Luethi et al., 2006).

**Demanda de pasajeros:** es uno de los conceptos cruciales para la planificación del transporte público, debido a que interactúa activamente en cada una de las fases, se puede dividir en tres niveles: regional, grupos de rutas y ruta específica. La demanda a nivel regional es útil para la primera fase (planificación estratégica) a la hora de diseñar la topología de las rutas (trayectos y paradas). Por grupos de rutas es útil para sincronizar horarios y reducir el tiempo de desplazamiento con viajes que incluyen el uso de

transferencias. Por último y el concepto más relevante para este trabajo es la demanda de pasajeros por ruta específica, estos datos pueden ser adquiridos de diferentes maneras como: verificadores en puntos geográficos, verificadores a bordo de los viajes o boletos electrónicos.

## 2.4 Creación de horarios o calendarios con vehículos de diferentes tipos

La planificación del transporte público considera el balance entre el costo operacional y la calidad del servicio para los pasajeros, por otra parte, es importante considerar el impacto ambiental y pensar en posibles mejoras que no solo brinden un buen servicio, sino que sean amigables con el ambiente. Utilizar vehículos de gran envergadura significa una gran cantidad de energía, por el peso y tamaño del vehículo además del gran número de pasajeros que puede trasladar, esto implica que para minimizar el consumo de energía por pasajero-kilometro trasladado, solo se puede utilizar este tipo de vehículos con una alta demanda de pasajeros y con un factor de carga que garantice la calidad del servicio. En periodos en los que la demanda de pasajeros es baja (p.ej. horas valle) es recomendable utilizar vehículos pequeños, ya que tienen un menor consumo de energía y tienen la posibilidad de utilizar energías renovables como baterías eléctricas, siendo beneficioso para el ambiente, reduciendo costos y manteniendo la calidad del servicio (Potter, 2003).

Los horarios están diseñados en base a la frecuencia de operación mínima requerida para satisfacer un nivel de servicio, posteriormente a la hora de calendarizar los vehículos necesarios se considera el costo de operación y posibles combinaciones entre diferentes tipos de vehículos, diferentes autores han abordado el problema de creación de horarios con diferentes enfoques y estrategias para el cálculo de la frecuencia de operación.

## 2.5 Trabajos relacionados

Uno de los primeros aportes al área específica de la creación de horarios basados en la determinación previa de la frecuencia en 1971, propone minimizar el tiempo de espera de los pasajeros, comparando un conjunto  $n$  de vehículos (buses, trenes, etc.) que tienen alguna hora de salida asignada  $t_j, j = 1, \dots, n$  con una función suavizada de llegada de pasajeros, presentando como resultados que la frecuencia óptima para vehículos de gran tamaño varía con el tiempo, aproximadamente como la raíz cuadrada de la tasa de

llegada de los pasajeros (Newell, 1971). Furth y Wilson (1981) propusieron un modelo que maximiza el beneficio social completo; el modelo se compone de los beneficios de los pasajeros, como la disminución del tiempo de espera. El modelo incorpora restricciones relacionadas con el tamaño de la flota, intervalos de operación permitidos (máximo caso permitido como inverso a la frecuencia mínima antes descrita) y el presupuesto total. El modelo de Furth y Wilson fue extendido por Koutsopoulos y Wilson (1985), quienes simplificaron el problema, dividiéndolo en intervalos de operación equidistantes, resolviéndolo mediante programación lineal. Proponen una formulación basada en las implicaciones sociales, dividida en tres componentes: los costos de funcionamiento, los tiempos de espera y los inconvenientes del hacinamiento de los pasajeros. Ceder recopila y define cuatro métodos diferentes para el cálculo de la frecuencia que dependen del perfil de carga de la demanda de pasajeros y las restricciones que estipulen los entes reguladores, presentando cómo obtener horarios óptimos al seleccionar la carga máxima de pasajeros como punto de referencia (Ceder, 1984, 2007, 2016).

Hadas y Shnaiderman (2012) abordan la minimización del costo total basado en el espacio vacío (p.ej. los asientos desocupados) y la demanda que no es atendida. Los autores definieron las distribuciones de probabilidad para los tiempos de viaje y la demanda de pasajeros con herramientas de geolocalización. Basándose en esta información, definieron un enfoque de optimización analítica que determina las frecuencias y el tamaño de los vehículos. La implementación con el enfoque propuesto muestra que la reducción de costos más significativa se obtiene en casos con bajo nivel de servicio. Li et al. (2013) consideran parámetros estocásticos tales como demanda, tiempos de espera, tiempos de embarque o descenso y tiempos de viaje. Los autores definen un método de optimización estocástica híbrido para encontrar la frecuencia que minimiza el tiempo de espera para los pasajeros y maximiza las ganancias del operador. Li et al. desarrollan un AG para resolver el problema y comparan su enfoque con los modelos tradicionales (analíticos) para el ajuste de la frecuencia propuestos por Newell (1971) y Ceder (1984b). Los autores afirman que los intervalos obtenidos son generalmente más grandes que los que usan el enfoque de Newell y más cortos que los que usan Ceder. Además, mencionan que estos intervalos moderados proporcionan un mejor equilibrio entre los costos operativos del bus y la satisfacción de los pasajeros.

Verbas y Mahmassani amplían el modelo presentado por Furth y Wilson (1981) considerando la variación de la demanda a lo largo del tiempo y la ruta lineal. El problema se formula con programación no lineal para minimizar la suma de los pasajeros y el tiempo de espera en todas las paradas, rutas e intervalos de tiempo, sujeto a restricciones tales como: presupuesto, tamaño de la flota y factores de carga (Verbas & Mahmassani, 2013; Verbas & Mahmassani, 2015).



Para el caso con enfoque multiobjetivo, Kwan y Chang (2008) presentan una formulación para el problema de diseño de horarios con dos objetivos en conflicto: minimizar el costo de la cantidad de transferencias necesarias y minimizar el costo causado por las desviaciones de un cronograma inicial. Los autores implementan un algoritmo genético clásico para resolver problemas con múltiples objetivos conocido como NSGA II (Deb et al., 2002), combinado con otros métodos (p.ej. búsqueda local) para resolver la formulación del problema. Hassold y Ceder (2014), estudian el problema de la creación de horarios con el objetivo de minimizar el tiempo de espera para los pasajeros (como un indicador de calidad del servicio) y una penalización basada en el espacio sin ocupar en los vehículos (como eficiencia operativa). La idea principal del estudio es poder combinar diferentes tipos de vehículos basados en la idea de Potter (2003) buscando evitar sobrecargas mejorando la utilización de los vehículos. Los autores implementan una heurística basada en grafos con la que combinan diferentes horarios en busca del conjunto óptimo de Pareto. Los resultados numéricos para un caso de estudio en Nueva Zelanda muestran un ahorro significativo en los tiempos de espera de los pasajeros, pero también una carga de pasajeros aceptable en todos los vehículos.

## 2.6 Definición del problema

### 2.6.1 Planteamiento del problema

El problema que se aborda en el presente trabajo es: la creación de horarios utilizando vehículos de diferentes tipos (capacidad y costo de operación) mediante el cálculo de la frecuencia de operación basado en la demanda de pasajeros (MVTTP por sus siglas en inglés: “*Multiobjective Vehicle-Type Timetabling Problem*”). Se propone entonces una formulación como un problema de optimización con dos objetivos en conflicto: reducir el costo de operación para los proveedores del servicio y minimizar el número de pasajeros que no pueden ser transportados en cada parada como indicador de calidad de servicio. MVTTP modela entonces un escenario realista donde un conjunto de vehículos de diferentes tipos (p.ej. buses articulados, minibuses, colectivos, buses estándar, buses de dos niveles, etc.) es asignado para cubrir un conjunto de viajes de una ruta específica.

Minimizar la función asociada al costo de operación  $f_2$ , adicionalmente contribuye a la reducción del impacto ambiental o contaminación al reducir el uso de combustibles fósiles como la gasolina o gasóleo (en motores diésel), mejora el flujo del tráfico al reducir la flota necesaria para cubrir un conjunto de viajes o asignar vehículos pequeños que utilizan menos espacio en las vías.

La función para la demanda de pasajeros no satisfecha ( $f_2$ ) tiene efectos adicionales en el retraso para abordar un vehículo (tiempo de espera) y el confort asociado al factor de carga y la capacidad de los vehículos, ya que se busca asignar un conjunto de vehículos que presenten la menor sobrecarga posible.

## 2.6.2 Definición formal del problema

La formulación formal para el problema MVTP es: Dado un superconjunto de vehículos con vehículos de diferentes tipos  $B = \{b_1, \dots, b_n\}$ , en el que  $b_i$  es la cantidad de vehículos de tipo  $i$ ,  $n$  la cantidad de diferentes tipos de vehículos  $\sum_{i=1}^n b_i$  es la flota disponible.  $T$  es un conjunto de  $m$  viajes  $T = \{t_1, \dots, t_m\}$  de una ruta específica  $R$ .

MVTP se define a partir de dos funciones objetivo  $f_1$  y  $f_2$  así:

$$\text{Minimizar} \quad f_1 = \sum_{i=1}^n \omega_i \quad (10)$$

$$\text{y} \quad f_2 = \sum_{j=1}^{Lp} \sum_{s \in R} LQ_j^s \quad (11)$$

$$\text{sujeto a:} \quad c_i = c_i^{bus} + c_i^{gas} + c_i^{driver} \quad (12)$$

$$\omega_i = c_i m_i \quad (13)$$

$$f_j \geq f_j^{min} \quad (14)$$

$$LF_j = \frac{P_j^{max}}{CAP_i \times f_j} \leq LF_j^{max}, i = 1, \dots, n \quad (15)$$

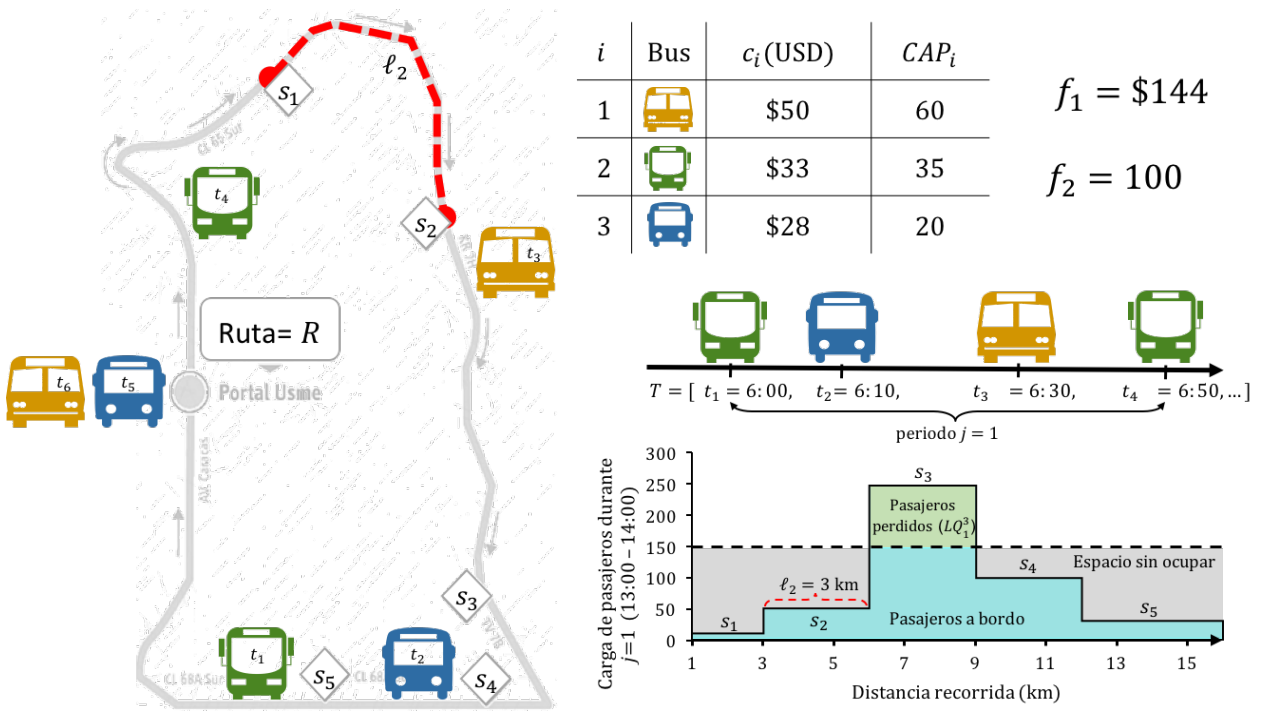
$$LQ_j^s = \max \left( P_j^s - \sum_{i \in M_j} LF_j \times CAP_i, 0 \right) \quad (16)$$

Donde:

$c_i^{gas}$	Costo del combustible para cada tipo de vehículo.
$c_i^{driver}$	Costo asociado al conductor para cada tipo de vehículo.
$c_i^{bus}$	Costo asociado a la operación del vehículo (p.ej. mantenimiento).
$c_i$	Costo total involucrado en un viaje para un vehículo de tipo $i$ .
$m_i$	Cantidad de vehículos de tipo $i$ necesarios para cubrir los viajes de $T$ .
$\omega_i$	Costo total al utilizar $m_i$ vehículos de tipo $i$ .
$P_j^{max}$	Cantidad máxima de pasajeros en cualquier parada $s$ en el lapso $j$ .

- $P_j^s$  Cantidad de pasajeros en una parada  $s$  en el lapso  $j$ .
- $f_j$  Frecuencia de operación en el lapso  $j$ .
- $f_j^{min}$  Frecuencia mínima de operación en el lapso  $j$ .
- $LF_j$  Factor de carga en el lapso  $j$ .
- $LF_j^{max}$  Factor de carga máximo en el lapso  $j$ .
- $M_j$  Conjunto de vehículos utilizados en el lapso  $j$ .
- $CAP_i$  Capacidad de los vehículos de tipo  $i$ .
- $LQ_j^s$  Demanda de pasajeros en la parada  $s$  que sobrepasa la capacidad de los vehículos  $i \in M_j$  en el lapso  $j$ .
- $Lp$  Último periodo de tiempo.

En la **Figura 4** se muestra un caso específico con tres diferentes vehículos, (bus, buseta y microbús) con capacidad y costo de operación diferentes. Se presenta una ruta ( $R$ ), con cinco paradas ( $s$ ), la cual debe ser cubierta por un conjunto de viajes ( $T$ ), utilizando una asignación de vehículos a cada viaje  $t_k$ , es posible trasladar a un grupo de pasajeros. Sin embargo, debido a la elección de vehículos realizada se tiene una pérdida de usuarios en la tercera parada, ya que la demanda en este punto ( $P_j^s$ ) excede la capacidad de los vehículos elegidos para ese periodo de tiempo ( $j$ ).



**Figura 4.** Trazado de la ruta y sus respectivas paradas definidas al momento del diseño topológico de  $R$  (izquierda). Tipos vehículos disponibles, calendario propuesto que asigna cada vehículo ( $i$ ) a un viaje ( $t_k$ ) y su comportamiento para una demanda de pasajeros obtenida en un periodo de tiempo  $j$  (derecha).

## Capítulo 3. Optimización multiobjetivo (OMO)

---

### 3.1 Introducción

Diversos problemas asociados a sistemas reales en biología, medicina, economía, industria, ingeniería y otras disciplinas exhiben un comportamiento en los que dos o más de sus objetivos están en conflicto de manera natural, por esta razón encontrar una solución de calidad, en un lapso de tiempo aceptable, ha sido y sigue siendo objeto de estudio por parte de los investigadores. En la investigación de operaciones y otras áreas afines, se han propuesto diferentes técnicas para abordar los problemas de decisión con múltiples objetivos o múltiples criterios, debido a la complejidad de los mismos se requieren diferentes enfoques que provean mejores resultados.

Los problemas asociados a optimización multiobjetivo (OMO), se componen de dos o más objetivos que entran en conflicto casi de manera evidente, la tarea es encontrar el conjunto de soluciones que satisfagan las restricciones y optimicen las funciones objetivo, las cuales deben satisfacerse con un grado de aceptabilidad definido desde el punto de vista del diseño, a diferencia de un problema con un solo objetivo en el que solo se busca una solución que optimice la única función objetivo y cumpla con las restricciones del problema (Coello et al., 2007; Deb et al., 2016).

En este capítulo, se presenta una breve introducción a OMO y parte de la terminología importante que se utilizará a lo largo del documento. Brevemente se presentarán las relaciones de dominancia y se discutirán algunos métodos de solución, incluido el uso de algoritmos evolutivos.

### 3.2 Conceptos básicos de OMO

Un problema de optimización multiobjetivo (POMO) se compone de un número  $k$  de funciones objetivo, las cuales deben ser minimizadas o maximizadas simultáneamente y al igual que en los problemas de único objetivo, cualquier solución factible (incluida la solución óptima), debe satisfacer ciertas restricciones definidas para cada caso particular. Con la finalidad de comprender los detalles de un POMO y las necesidades específicas que se pretenden abordar con los métodos de solución propuestos, iniciaremos la discusión partiendo de un problema con un objetivo definido por Coello et al. (2007) de la siguiente forma:

De manera general un problema de optimización mono-objetivo está definido como una minimización (o maximización) de una función  $f(\mathbf{x})$  sujeto a  $g_i(\mathbf{x}) \leq 0, i = \{1, \dots, m\}$ , y  $h_j(\mathbf{x}) = 0, j = \{1, \dots, p\}, \mathbf{x} \in \Omega$ . Una solución que minimice (o maximice) la función escalar  $f(\mathbf{x})$  donde  $\mathbf{x}$  es un vector de variables de decisión dimensión  $N$ ,  $\mathbf{x} = (x_1, \dots, x_N)$ , para algún universo  $\Omega$ .

Se puede observar que  $g_i(\mathbf{x}) \leq 0$  y  $h_j(\mathbf{x}) = 0$  representan las restricciones que se deben satisfacer mientras se optimiza  $f(\mathbf{x})$ . El universo  $\Omega$  contiene todas las posibles soluciones que pueden ser utilizadas para satisfacer la función y sus restricciones.

Por lo tanto, un método que permita encontrar la solución que provee un valor considerado como óptimo global (que puede no ser única) para un problema con una sola función objetivo, se conoce como optimización global, y se puede definir para un caso de minimización (sin perder generalidad) como: Dada una función  $f: \phi \subseteq \mathbb{R}^n \rightarrow \mathbb{R}, \phi \neq \emptyset$ , para un  $x \in \phi$  el valor de  $f^* \triangleq f(\mathbf{x}^*) > -\infty$  se conoce como óptimo global si y solo si  $\forall \mathbf{x} \in \phi: f(\mathbf{x}^*) \leq f(\mathbf{x})$ . Donde  $\mathbf{x}^*$  es la solución mínima global,  $f$  es la función objetivo y el conjunto  $\phi$  es la región factible de  $\mathbf{x}$  (Coello et al., 2007). Por otra parte, en los POMO, es posible encontrar un conjunto de soluciones en algunos casos infinito que, al ser evaluados, producen unos vectores cuyos componentes representan los valores de  $\Omega$  en el espacio asociado a las funciones objetivo. Osyczka (1985) define un POMO como:

*“Encontrar un vector de variables de decisión que satisfaga las restricciones y optimice una función vectorial cuyos elementos representan las funciones objetivo. Estas funciones son una descripción matemática de criterios de rendimiento los cuales usualmente están en conflicto unos con otros.”*

Entonces, podemos definir (matemáticamente) de manera general un POMO de la siguiente manera:

Minimizar/maximizar

$$f_k(\mathbf{x}), \quad k = 1, \dots, n \quad (17)$$

sujeto a:

$$g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m \quad (18)$$

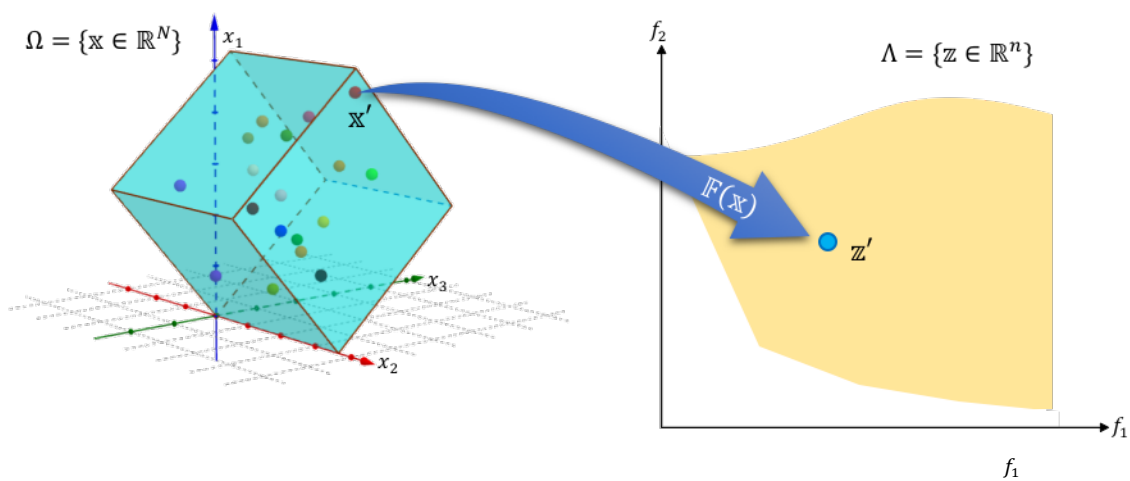
$$h_j(\mathbf{x}) = 0, \quad j = 1, \dots, p \quad (19)$$

$$\mathbf{x} \in \Omega \quad (20)$$

Donde una solución  $\mathbf{x}$  es una variable de decisión vectorial de dimensión  $N$ ,  $\mathbf{x} = [x_1, \dots, x_N]^T$ , en algún universo  $\Omega$  (**espacio de decisión**). Por lo general, para un POMO, existen varias soluciones con diferentes

niveles de compensación para las  $n$  funciones objetivo, que satisfacen las  $m + p$  restricciones del problema y componen el conjunto **óptimo de Pareto**, debido a que son aquellas que presentan el mejor rendimiento posible. En la mayoría de los POMO, se presentan limitaciones producidas por el manejo de los recursos disponibles para su ejecución o situaciones particulares en el ambiente o entorno del mismo, una solución al problema debe cumplir con cada una de estas limitantes las cuales son conocidas como **restricciones** y pueden ser representadas matemáticamente como desigualdades,  $g_i(\mathbf{x}) \leq 0$ , o igualdades,  $h_j(\mathbf{x}) = 0$ . Por lo tanto, se cuenta con  $m$  restricciones de desigualdad y  $p$  restricciones de igualdad; si una solución  $\mathbf{x}$  satisface todas las  $m + p$  restricciones es conocida como **solución factible**, la cual forma parte de un conjunto o región factible  $\mathcal{S} \subseteq \Omega$  (**espacio de búsqueda**).

Los problemas con múltiples objetivos son aquellos en los que la meta es optimizar  $n$  funciones objetivo simultáneamente (Coello et al., 2007). Los objetivos del problema, también conocidos como criterios, deben ser minimizados o maximizados según lo defina cada problema, sin embargo, es común abordar problemas únicamente de minimización en cada uno de sus objetivos dada la posibilidad de modificar un problema con objetivos mixtos (minimización y maximización combinados) mediante el teorema de dualidad (S. S. Rao, 2009). El vector de  $n$  funciones objetivo,  $\mathbb{F}(\mathbf{x}) = [f_1(\mathbf{x}), \dots, f_n(\mathbf{x})]^T$ , define un espacio coordinado (**espacio objetivo**  $\Lambda$ ) en el que cada solución puede ser representada; de manera que, se cuenta con un espacio de variables de decisión con dimensión  $N$ , en el que cada eje coordinado corresponde a los componentes del vector  $\mathbf{x}$ , y el espacio objetivo de dimensión  $n$ , en el que cada eje coordinado corresponde a un componente vectorial  $f_k(\mathbf{x})$ . Entonces para cada solución  $\mathbf{x}$  en el espacio de decisión existe un punto en el espacio objetivo el cual es mapeado a través de las funciones objetivo, la **Figura 5** muestra los dos espacios definidos anteriormente y la relación entre ellos.



**Figura 5.** Representación gráfica del espacio de decisión ( $\Omega$ ) y su correspondiente espacio objetivo ( $\Lambda$ ).

Por ende, es posible dividir el espacio de búsqueda en dos zonas que no se superponen, una en donde se encuentran las soluciones óptimas de Pareto (región óptima) y otra con las soluciones restantes. Como se mencionó anteriormente, es común encontrar múltiples soluciones en la región óptima, lo cual impide elegir una única solución para el POMO cuando no se cuenta con información adicional sobre el problema, por lo que todas las soluciones óptimas de Pareto son igual de importantes. Por lo cual, es importante encontrar la mayor cantidad de soluciones Pareto óptimas como sea posible, lo que permite estructurar dos metas pertinentes en OMO, primero encontrar el conjunto de soluciones más cercano a la región óptima real del problema, conocida como **frente de Pareto** y que a su vez sea lo más “diverso” posible tanto en el espacio de decisión como en el espacio objetivo, la diversidad está asociada a la distancia euclidiana entre las soluciones.

### 3.2.1 Vector ideal

Una solución particular para un POMO conocida como el **vector ideal**, consiste en un vector de variables que optimiza cada uno de las  $n$  funciones objetivo:

$$\mathbb{x}^{*(k)} = [x_1^{*(k)}, x_2^{*(k)}, \dots, x_N^{*(k)}]^T \quad (21)$$

Por lo que tenemos que un  $\mathbb{x}^{*(k)} \in \Omega$  es el que optimiza la  $k$ -ésima función objetivo  $f_k(\mathbb{x})$ , es decir:

$$f_k(\mathbb{x}^{*(k)}) = \underset{\mathbb{x} \in \Omega}{\text{óptimo}} (f_k(\mathbb{x})) \quad (22)$$

Entonces el **vector objetivo ideal**  $\mathbb{z}^*$  es:

$$\mathbb{z}^* = \mathbb{F}^* = [f_1^*, f_2^*, \dots, f_n^*]^T \quad (23)$$

$f_k^*$  representa el valor óptimo de la  $k$ -ésima función objetivo, el punto en  $\mathbb{R}^n$  que se obtiene a partir de  $\mathbb{F}^*$  se considera la solución ideal para el problema propuesto. En términos generales, el vector ideal es una solución que no existe debido a que la solución óptima para un POMO debe ser diferente para cada función objetivo y la única manera de que un vector ideal sea una solución factible, es cuando las soluciones óptimas son iguales; en este caso, las funciones objetivo no estarían en conflicto.

Por otra parte, es posible definir un **vector objetivo ficticio**  $\mathbb{z}^{**}$  cuyos componentes son estrictamente menores (para un caso de minimización) que el vector objetivo ideal, es decir  $\mathbb{z}_k^{**} = \mathbb{z}_k^* - \beta_k$  con  $\beta_k > 0$  para todo  $k = 1, 2, \dots, n$ . Cuya utilidad radica en la implementación de algoritmos para solucionar POMO

que requieren una solución que sea estrictamente mejor que cualquiera posible en el espacio de búsqueda, diferente a la función de cota inferior del vector ideal, considerando la posibilidad de que alguna solución comparta por lo menos uno de sus componentes.

### 3.2.2 Conceptos de dominancia y Pareto

Algunos de los algoritmos utilizados en OMO utilizan el concepto de **dominancia ( $<$ ) o dominancia de Pareto**, permitiendo comparar dos soluciones obtenidas ( $u$  y  $v$ ) y concluir cuál de las dos domina al otro, lo que puede implicar cuál de los dos es mejor, podemos definirlo como: dados dos vectores  $u = (u_1, \dots, u_n)$  y  $v = (v_1, \dots, v_n)$ , donde  $u_k = f_k(\mathbb{x}')$  y  $v_k = f_k(\mathbb{x})$ , podemos decir que  $u$  domina  $v$ , expresado como  $u < v$ , si y solo si  $u$  es menor o igual a  $v$  (para un caso de minimización) pero  $u$  es estrictamente menor que  $v$  en al menos uno de sus componentes, i.e.,  $\forall k \in \{1, \dots, n\}, u_k \leq v_k \wedge \exists k \in \{1, \dots, n\}: u_k < v_k$ .

Debido a esto podemos plantear tres posibilidades a la hora de comparar dos soluciones:

- $u$  domina a  $v$  ( $u < v$ ).
- $u$  es dominado por  $v$  ( $v < u$ ).
- $u$  y  $v$  no son comparables, dado que uno no domina al otro y viceversa.

Lo anterior permite identificar una propiedad interesante de la dominancia, donde si una solución  $u$  no domina a una solución  $v$ , no implica que  $v$  domine a  $u$ . Deb et al., (2016) proponen algunas propiedades importantes del operador de dominancia:

- *Reflexividad*: la relación de dominancia es irreflexiva debido a que cualquier solución  $u$  no puede dominarse a sí misma.
- *Simetría*: la relación de dominancia es asimétrica debido a que si  $u$  domina a  $v$  no implica que  $v$  domine a  $u$ . Por lo que el caso contrario es cierto, si  $u$  domina a  $v$  entonces  $v$  no domina a  $u$ .
- *Transitividad*: la relación de dominancia es transitiva, lo cual significa que si  $u$  domina a  $v$  y  $v$  domina a  $w$ , entonces  $u$  domina a  $w$ .



Así entonces se facilita comprender el concepto de un “óptimo”, debido a que como se mencionó anteriormente lo que se busca como solución a un POMO, es un conjunto de soluciones óptimas que no sean dominadas por ninguna otra solución en el espacio de búsqueda.

Propuesto inicialmente por Francis Ysidro Edgeworth en su trabajo relacionado con las curvas de indiferencia en economía a finales del siglo XIX (Edgeworth, 1881) y extendido en 1896 por Vilfredo Pareto (Pareto, 1896), el término **óptimo de Edgeworth-Pareto**, más conocido como **óptimo de Pareto**, puede definirse formalmente como:

**Optimalidad de Pareto:** Una solución  $\mathbf{x} \in \Omega$  es óptima de Pareto con respecto a  $\Omega$  si y solo si no existe un  $\mathbf{x}' \in \Omega$  tal que  $\mathbb{F}(\mathbf{x}') < \mathbb{F}(\mathbf{x})$ , i.e.  $\mathbb{F}(\mathbf{x}') = (f_1(\mathbf{x}'), \dots, f_n(\mathbf{x}'))$  domine a  $\mathbb{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_n(\mathbf{x}))$ . Por lo que la frase “**óptimo de Pareto**” hace referencia al espacio de decisión.

Basado en la optimalidad de Pareto, Coello et al. (2007) definen algunos conceptos útiles para la solución de POMO como el conjunto óptimo de Pareto, también conocido como **conjunto de Pareto** (conjunto de soluciones no dominadas  $\mathcal{P}_S$  en el espacio de decisión) y el conjunto de soluciones no dominadas representadas a través de la evaluación de todas las posibles soluciones en  $\Omega$  mediante las funciones objetivo que al plasmarse en el espacio objetivo se conoce como **frente de Pareto** ( $\mathcal{P}_F$ ).

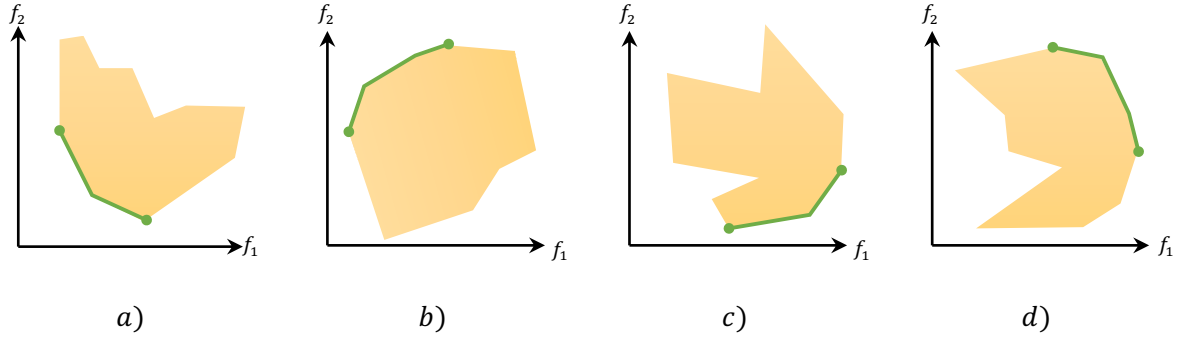
**Conjunto de Pareto:** Para un POMO dado,  $f(\mathbf{x})$ , el conjunto óptimo de Pareto,  $\mathcal{P}_S$ , se define como:

$$\mathcal{P}_S = \{\mathbf{x} \in \Omega \mid \nexists \mathbf{x}' \in \Omega f(\mathbf{x}') < f(\mathbf{x})\} \quad (24)$$

**Frente de Pareto:** Para un POMO dado,  $f(\mathbf{x})$ , y un conjunto óptimo de Pareto,  $\mathcal{P}_S$ , el frente de Pareto  $\mathcal{P}_F$  se define como:

$$\mathcal{P}_F = \{f(\mathbf{x}) \in \mathbb{R}^n \mid \mathbf{x} \in \mathcal{P}_S\} \quad (25)$$

Es importante resaltar que, para un caso mono-objetivo, cada problema tiene una única solución y en los POMO normalmente el frente de Pareto contiene un número de soluciones que puede ser infinito, por lo tanto, en muchas ocasiones no se conoce el  $\mathcal{P}_F$  y se buscan las mejores aproximaciones tanto en cercanía como en diversidad del mejor conjunto solución encontrado (ver **Figura 6**).



**Figura 6.** Soluciones óptimas de Pareto (frente de Pareto) en el espacio objetivo ( $\Lambda$ ), para los cuatro casos posibles con dos funciones objetivo ( $f_1$  y  $f_2$ ). a). Minimización-Minimización. b). Minimización-Maximización. c). Maximización-Minimización. d). Maximización-Maximización.

La relación de dominancia ( $<$ ) presenta una debilidad en el sentido que no permite relacionar dos conjuntos iguales o que un conjunto contenga un mayor número de soluciones no dominadas que otro, es necesario otro tipo de enfoque que permita plantear la relación de los conjuntos no solo en el espacio objetivo sino preguntarnos por el comportamiento de los conjuntos en el espacio de decisión, de ahí algunos autores proponen un par de definiciones adicionales que permite comparar los conjuntos soluciones para los POMO, garantizando entregar siempre la mejor aproximación al frente de Pareto (Zitzler et al., 2003; Deb et al., 2016).

**Dominancia fuerte ( $<<$ ):** una solución  $v$  es fuertemente dominada por una solución  $u$  ( $u << v$ ), si para las  $n$  funciones objetivo,  $u$  tiene un mejor rendimiento que  $v$  i.e.  $\forall k \in \{1, \dots, n\}: u_k < v_k$ .

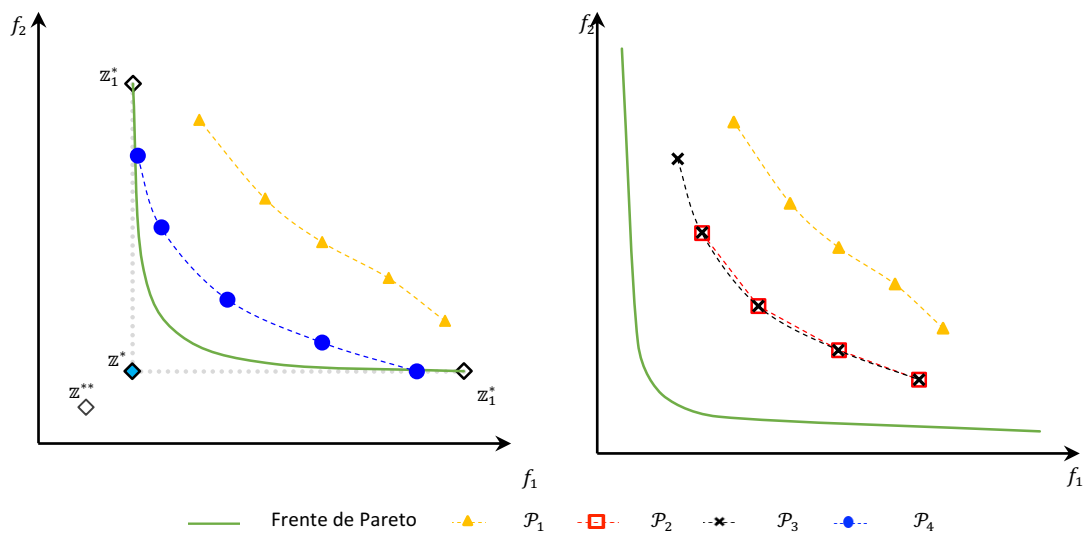
**Dominancia débil ( $\preceq$ ):** un conjunto solución  $\mathcal{P}_1$  es débilmente dominado por un conjunto  $\mathcal{P}_2$  ( $\mathcal{P}_2 \preceq \mathcal{P}_1$ ), si para las  $n$  funciones objetivo, ninguna de las soluciones en  $\mathcal{P}_2$  domina fuertemente a algún miembro de  $\mathcal{P}_1$ .

Para hacer claridad de la dominancia débil (Zitzler et al., 2003) proponen utilizar la relación “**mejor que**” ( $\triangleleft$ ), de manera que si  $\mathcal{P}_2$  es al menos tan bueno como  $\mathcal{P}_1$  pero  $\mathcal{P}_1$  no es tan bueno como  $\mathcal{P}_2$ , i.e.  $\mathcal{P}_2 \preceq \mathcal{P}_1 \wedge \mathcal{P}_1 \not\preceq \mathcal{P}_2$ , en otras palabras Deb et al. (2016) sugiere que una dominancia débil es cuando la cardinalidad del conjunto débilmente dominado ( $|\mathcal{P}_1|$ ) es inferior o igual a la cardinalidad del conjunto ( $|\mathcal{P}_2|$ ), dado que debe contener al menos las mismas soluciones. Por lo que podemos definir la dominancia débil ( $\preceq$ ) partir de la relación  $\triangleleft$  (mejor que) como:  $\mathcal{P}_2 \preceq \mathcal{P}_1 \implies \mathcal{P}_2 \triangleleft \mathcal{P}_1 \vee \mathcal{P}_2 = \mathcal{P}_1$ , es decir, si  $\mathcal{P}_2$  domina débilmente a  $\mathcal{P}_1$ , entonces  $\mathcal{P}_2$  es mejor que  $\mathcal{P}_1$  o son iguales. En la **Tabla 1** se presenta un resumen de las relaciones de dominancia presentadas anteriormente.

**Tabla 1.** Relaciones de dominancia entre dos soluciones (puntos en el espacio objetivo) y entre dos conjuntos de aproximación al frente de Pareto.

Relación	Espacio objetivo ( $\mathbb{R}^n$ )		Conjuntos de aproximación en $\Omega$	
Incomparables	$u \not\leq v$	Ni $u$ domina débilmente a $v$ ni viceversa.	$\mathcal{P}_2 \not\leq \mathcal{P}_1$	Ni $\mathcal{P}_2$ domina débilmente a $\mathcal{P}_1$ ni viceversa.
Dominancia débil	$u \leq v$	$u$ no es peor que $v$ para todos los $n$ objetivos.	$\mathcal{P}_2 \leq \mathcal{P}_1$	Cada $v \in \mathcal{P}_1$ es dominado débilmente por al menos un $u \in \mathcal{P}_2$ .
Mejor que	_____	_____	$\mathcal{P}_2 \triangleleft \mathcal{P}_1$	Cada $v \in \mathcal{P}_1$ es dominado débilmente por al menos un $u \in \mathcal{P}_2$ y $\mathcal{P}_2 \neq \mathcal{P}_1$ .
Dominancia	$u < v$	$u$ no es peor que $v$ para todos los $n$ objetivos, pero es mejor en mínimo un objetivo.	$\mathcal{P}_2 < \mathcal{P}_1$	Cada $v \in \mathcal{P}_1$ es dominado por al menos un $u \in \mathcal{P}_2$ .
Dominancia fuerte	$u \ll v$	$u$ mejor que $v$ para todos los $n$ objetivos.	$\mathcal{P}_2 \ll \mathcal{P}_1$	Cada $v \in \mathcal{P}_1$ es fuertemente dominado por al menos un $u \in \mathcal{P}_2$ .

Como ejemplo, la **Figura 7** presenta el frente de Pareto ( $\mathcal{P}_F$ ) de dos POMO diferentes, para dos casos de minimización con dos funciones objetivo ( $f_1$  y  $f_2$ ): Se muestra gráficamente el vector objetivo ideal ( $z^* = (z_1^*, z_2^*)$ ), a partir de los valores extremos de  $\mathcal{P}_F$ , un vector objetivo ficticio ( $z^{**}$ ) y un conjunto aproximación al frente de Pareto ( $\mathcal{P}_4$ ), el cual puede ser considerado como la mejor solución posible si se compara con  $\mathcal{P}_1$  (izquierda). Por otra parte para los tres vectores objetivo  $\mathcal{P}_1, \mathcal{P}_2$ , y  $\mathcal{P}_3$  presentados en la parte derecha de la figura, las relaciones de dominancia son:  $\mathcal{P}_3 \ll \mathcal{P}_1, \mathcal{P}_2 \ll \mathcal{P}_1, \mathcal{P}_3 < \mathcal{P}_1, \mathcal{P}_2 < \mathcal{P}_1, \mathcal{P}_3 \triangleleft \mathcal{P}_1, \mathcal{P}_3 \triangleleft \mathcal{P}_2, \mathcal{P}_2 \triangleleft \mathcal{P}_1, \mathcal{P}_3 \leq \mathcal{P}_1, \mathcal{P}_3 \leq \mathcal{P}_2, \mathcal{P}_2 \leq \mathcal{P}_1, \mathcal{P}_3 \leq \mathcal{P}_3, \mathcal{P}_2 \leq \mathcal{P}_2 \vee \mathcal{P}_1 \leq \mathcal{P}_1$ .



**Figura 7.** Vector objetivo ideal ( $z^*$ ) y ficticio ( $z^{**}$ ), frente de Pareto ( $\mathcal{P}_F$ ) y dos posibles conjuntos de soluciones  $\mathcal{P}_1$ , y  $\mathcal{P}_4$  (izquierda). Asimismo, tres vectores objetivo diferentes ( $\mathcal{P}_1, \mathcal{P}_2, \mathcal{P}_3$ ) y su respectivo  $\mathcal{P}_F$  (derecha), para dos POMO de minimización con dos funciones objetivo  $f_1$  y  $f_2$ .

### 3.3 Métodos de solución

Solucionar problemas de optimización ha sido objeto de estudio durante muchos años, donde los métodos de búsqueda para encontrar la mejor solución posible o sencillamente técnicas de optimización pueden ser tan simples como una selección aleatoria o tan costosos (en términos computacionales) como hacer una búsqueda exhaustiva en todo el espacio de decisión, evaluando una por una cada solución factible, realizando un ranking que permita seleccionar la solución con mejor rendimiento. Sin embargo, es evidente que para un problema con un espacio de búsqueda muy grande este tipo de técnicas es ineficiente, o imposible en algunos casos. Con el avance en matemáticas y sistemas computacionales es posible dividir las técnicas de optimización en dos grupos: los métodos exactos, p. ej. búsqueda exhaustiva o programación lineal, y las heurísticas como **recocido simulado**, **búsqueda tabú** o **algoritmos genéticos**.

Muchos de los problemas en ciencias o ingeniería tienen objetivos en conflicto y se necesita encontrar soluciones “aceptables” en un tiempo “aceptable” (Coello et al., 2007), por lo que la implementación de métodos exactos es inviable dado que pueden ser problemas con una dimensionalidad alta, discontinuos y/o NP-completos. Para abordar estos casos se hace necesario guiar la búsqueda o limitar el espacio basado en conocimiento previo del problema (heurísticas), diferentes métodos estocásticos han sido desarrollados gracias a la facilidad de implementación, bajo consumo de recursos computacionales y velocidad de respuesta aceptable, estos métodos utilizan una función de “*fitness*” o aptitud, con la intención de asignar un valor a cada posible solución y una codificación o mapeo entre el espacio de solución y el algoritmo. Este tipo de técnicas eventualmente puede lograr alcanzar el caso óptimo para un problema dado, sin embargo, en la mayoría de los casos no se puede garantizar que en todas las ejecuciones es posible alcanzar la solución óptima, pero en general las soluciones que resultan con este tipo de técnicas son consideradas “aceptables” en comparación al esfuerzo que debe realizar un método determinista.

Técnicas simples como búsqueda o caminata aleatoria, donde en cada iteración se selecciona la mejor opción (un valor alto en “*fitness*”) de cierta cantidad de soluciones aleatorias, o técnicas más robustas como búsqueda tabú o recocido simulado, donde la búsqueda se realiza de forma guiada por diferentes factores en pro de alcanzar una solución óptima, sin embargo, como se mencionó anteriormente para un POMO no se encuentra una única solución sino un conjunto de soluciones no comparables entre ellas, por lo que una técnica de tipo poblacional como los algoritmos evolutivos (AE) puede presentar un mejor rendimiento, ya que el conjunto solución evoluciona al mismo tiempo permitiendo converger al frente de Pareto con el paso de cada iteración del algoritmo.

Los AE están inspirados en la evolución natural, se basan en el concepto de “selección natural” y la “supervivencia del más apto”, concepto clásico en la teoría evolutiva de Charles Darwin. Esta técnica será detallada y ampliada en el Capítulo 4 del presente documento.

### 3.3.1 Clasificación de los métodos para POMO

Para POMO se han propuesto diferentes estrategias o técnicas, las cuales se pueden clasificar mediante la interacción con el tomador de decisiones, el cual se encarga de seleccionar o decidir, como su nombre lo indica, cuál de las soluciones tiene una compensación para cada uno de los objetivos en conflicto la cual satisface los requerimientos específicos que dependen del conocimiento *a priori* del problema, es decir, cuánto se está dispuesto a ganar o perder para cada objetivo y evaluar la solución que mejor se acomode a estas restricciones. El tomador de decisiones juega un rol importante a la hora de solucionar un POMO, por esta razón es popular la clasificación basada en su aporte durante la búsqueda de una solución y consiste en los siguientes métodos:

- **Método sin preferencias:** este tipo de técnicas no considera información del tomador de decisiones para resolver un POMO, entregando como resultado un conjunto de soluciones ordenadas por algún criterio de compromiso entre objetivos de tal manera que el tomador de decisiones puede aprobar o rechazar dicha solución. Este caso sucede cuando no se cuenta con el tomador de decisiones o este no cuenta con información relevante que aporte a la solución del problema.
- **Método a priori:** para este tipo de métodos se consulta al tomador de decisiones antes de realizar la búsqueda de tal manera que se utiliza la información de preferencias que provee el tomador de decisiones con el fin de estructurar las funciones objetivo y que el conjunto solución satisfaga estos requerimientos, dos de los métodos *a priori* más conocidos son el **ordenamiento lexicográfico** (Fishburn, 1974) y **“goal programming”** (programación por metas) (Charnes y Cooper, 1961).
- **Método a posteriori:** en los métodos *a posteriori*, un conjunto de soluciones no dominadas es presentado ante el tomador de decisiones el cual selecciona una o varias soluciones basado en sus preferencias, técnicas como **“ $\epsilon$  –constraint”** (Haines et al. , 1971) o la **combinación lineal** (Zadeh, 1963) son utilizados como métodos *a posteriori*.

- **Método interactivo:** en los métodos interactivos, el tomador de decisiones actúa en cada una de las iteraciones del algoritmo, se definen tres etapas dentro del método, la primera es encontrar un conjunto de soluciones no-dominadas, luego a partir de la reacción del tomador de decisiones al evaluar el conjunto de soluciones, modificar las preferencias de las funciones objetivo, por último repetir los dos pasos anteriores hasta que no se presenten nuevas mejoras o el tomador de decisiones esté satisfecho.

### 3.3.2 Método del criterio global (método sin preferencia)

Yu (1973) introduce el término de vector utópico (anteriormente mencionado como vector ideal, ver Capítulo 3 sección 3.2.1) con la intención de encontrar una solución óptima de Pareto lo más cercana posible a  $\mathbf{z}^*$ , se han propuesto métricas diferentes para evaluar esta “cercanía”, sin embargo, una de las más utilizadas es la métrica  $L_p$ .

$$L_p(\mathbb{F}) = \left[ \sum_{k=1}^n |f_k^* - f_k(\mathbf{x})|^p \right]^{1/p} \quad (26)$$

Este método conocido como método de criterio global o “*compromise programming*”, busca minimizar el valor de  $L_p$  para  $1 \leq p \leq \infty$  tal que  $\mathbf{x} \in \Omega$ , de tal manera que se puede obtener una solución óptima de Pareto para cada valor de  $p$ . Este método ha sido estudiado fuertemente durante décadas modificando las métricas con la finalidad de encontrar las mejores soluciones para diferentes problemas.

La idea básica de no contar con la información del tomador de decisiones para realizar la búsqueda (métodos sin preferencia) se puede observar en este tipo de técnicas; sin embargo es posible modificarlas de tal manera que funcionen igual que un método *a priori* como lo propone Wierzbicki (1980), quien utiliza un vector ficticio ( $\mathbf{z}^{**}$ ) para direccionar la búsqueda.

Por lo tanto  $\mathbf{z}^{**} = \mathbb{F}^{**} = [f_1^{**}, f_2^{**}, \dots, f_n^{**}]^T$  es el punto de referencia en el espacio objetivo, el cual es elegido por el tomador de decisiones y es considerado el vector con valores deseados o aceptables.

### 3.3.3 Ordenamiento lexicográfico (método *a priori*)

Este método utiliza la información del tomador de decisiones para definir un ranking de las funciones objetivo. La idea es buscar una solución que sea la mejor posible (o el óptimo en algunos casos) para el primer objetivo del ranking, continuando con cada uno de los objetivos hasta el menos relevante. La solución al final de la ejecución, i.e.  $\mathbf{x}_n^*$  para un problema de  $n$  objetivos, es presentada como la solución final al POMO ( $\mathbf{x}^*$ ) (S. Rao, 1984).

**Combinación lineal (método *a posteriori*):** para este método se utiliza un vector de pesos  $\mathbf{w} = (w_1, \dots, w_n)^T$  tal que  $w_i \geq 0$  para todos los  $i = 1, \dots, n$  y  $\sum_{i=1}^n w_i = 1$ , con el que se realiza una combinación lineal de los  $n$  objetivos, buscando resolver el problema utilizando técnicas para problemas de un único objetivo, minimizando la suma ponderada de los componentes de la función vectorial objetivo  $\mathbb{F}$ .

El problema de optimización de un solo objetivo (mono-objetivo) a resolver en el método de combinación lineal es:

Minimizar

$$\sum_{i=1}^n w_i f_i(\mathbf{x}) \quad (27)$$

sujeto a:

$$\mathbf{x} \in \Omega \quad (28)$$

Una vez que se obtiene una solución, esta es revisada por el tomador de decisiones quien evalúa la viabilidad de seleccionarla o por el contrario proceder a ajustar los valores de  $\mathbf{w}$  y volver a realizar el proceso.

### 3.3.4 Restricción- $\epsilon$ (método *a posteriori*)

Un método que se basa en la linealidad de un POMO es el método de restricción  $\epsilon$  más conocido como " $\epsilon$  -constraint" en inglés, donde uno de los objetivos es seleccionado como la función a optimizar y los objetivos restantes pasan a ser restricciones del problema que definen cotas específicas.

Para un problema donde se busca minimizar todas las funciones objetivo se representa de la siguiente manera:

Minimizar

$$f_j(\mathbf{x}) \quad (29)$$

sujeto a:

$$f_k(\mathbf{x}) \leq \epsilon_k \quad \forall k \neq j \quad (30)$$

$$\mathbf{x} \in \Omega \quad (31)$$

$\epsilon_k$  representa una cota superior para cada función objetivo; por lo tanto, al variar los valores  $\epsilon_k$  y/o seleccionar otra función objetivo  $f_j$  a optimizar, es posible obtener diferentes soluciones no dominadas, hay que ser cuidadoso, debido a que normalmente no se conoce de antemano los valores para  $\epsilon_k$  que pueden proveer los mejores resultados. Este método, también conocido como método de compensación ("*trade-off method*"), generalmente es utilizado como un método *a posteriori*, aunque es posible utilizar la información del tomador de decisiones preliminarmente con el fin de establecer unos valores iniciales para  $\epsilon_k$ .

### 3.3.5 Optimización evolutiva multiobjetivo (método *a posteriori*)

Schaffer es considerado como el primer investigador en utilizar los algoritmos evolutivos para resolver múltiples objetivos (AEMO de ahora en adelante) en su tesis "*Some experiments in machine learning using vector evaluated genetic algorithms*", su algoritmo VEGA (siglas en inglés para: "*Vector Evaluated Genetic Algorithm*") implementaba los tres operadores genéticos clásicos (selección, cruzamiento y mutación) de tal manera que cíclicamente evolucionaba los individuos mediante un método de selección basado en las funciones objetivo, por lo que para cada objetivo realizaba una selección y aplicaba cruzamiento y mutación (Schaffer, 1984).

Los AEMO son un método *a posteriori* fundamentado en los principios de los algoritmos evolutivos (AE) para optimización mono-objetivo, los cuales están descritos en el Capítulo 4 sección 4.2. Como se mencionó los AE son una estrategia atractiva para abordar POMO dada su capacidad de trabajar simultáneamente con un conjunto de posibles soluciones, conocido como población, los cuales pueden pertenecer al conjunto óptimo de Pareto y es posible alcanzarlos en una sola ejecución, a diferencia de un método tradicional donde en ejecuciones independientes se encuentran las soluciones. Adicionalmente



los AE son menos susceptibles a la forma y continuidad del frente de Pareto, es decir, pueden operar con frentes cóncavos o discontinuos.

Para el presente trabajo, se propone abordar el problema de MVTPP definido en el Capítulo 2 sección 2.6 con un tipo de AE conocidos como algoritmos genéticos (AG), puntualmente AG de tipo celular (**AGC**) (Alba y Dorronsoro, 2008; Nebro et al., 2009a), puesto que varios autores han destacado el rendimiento de este tipo de técnicas para casos de optimización multiobjetivo como los diferentes “*benchmark*” o problemas de referencia propuestos por Deb et al. (2005) y Zitzler et al. (2000), al igual que para posibles implementaciones reales en ingeniería o calendarización (Guzek et al., 2014; Zavala et al., 2015). Adicionalmente los AGC pueden ser implementados en paralelo, lo que supone una reducción en el tiempo de ejecución del mismo, aumentando su versatilidad y rendimiento. Los detalles y características de los AE, y sus diferentes variantes, son presentados en el Capítulo 4.

## Capítulo 4. Algoritmos evolutivos para problemas de OMO

---

### 4.1 Introducción

El cómputo evolutivo (CE) es un área de investigación dentro de las ciencias computacionales que en términos generales propone métodos de búsqueda estocásticos inspirados en el proceso de evolución natural, el CE propone técnicas como los **algoritmos genéticos** (AG), **programación genética** (PG), **estrategias evolutivas** (EE) y **programación evolutiva** (PE), que en general se conocen como algoritmos evolutivos (AE). Eiben y Smith (2015) proponen la siguiente metáfora:

*“The fundamental metaphor of evolutionary computing relates this powerful natural evolution to a particular style of problem solving – that of trial-and-error.”*

Lo que supone que la inspiración en procesos naturales proviene de la capacidad de las especies de evolucionar y es evidente que hoy en día estos procesos se mantienen fuertes y moldean nuestro entorno, es interesante como especies “débiles” o con un número reducido de individuos, han perdurado a través de los años y continúan mejorando con el pasar de las generaciones. El concepto de selección natural donde prevalece la supervivencia del más apto, propuesto por Darwin en 1859 en su obra “El origen de las especies”, complementado con la necesidad de adaptación al cambio como método de supervivencia, otro concepto importante también proveniente del darwinismo, sugieren que, mediante la herencia, las nuevas generaciones tienen en su ADN toda la información que ha permitido que la especie evolucione y sobreviva. La información genética heredada puede sufrir algunos cambios o mutaciones mediante prueba y error, generando nuevas características y habilidades, posibilitando que un individuo en la última generación pueda o no convertirse en el más apto.

En términos generales un AE está inspirado en el proceso de selección natural, creando una población inicial (conjunto de posibles soluciones) para después manipular la información genética de los individuos, buscando evolucionar hasta encontrar el individuo con mayor nivel de aptitud para un objetivo dado, comúnmente conocido como función de calidad, aptitud o “*fitness*”. Por lo tanto, los algoritmos evolutivos, resultan una herramienta útil para resolver problemas en los cuales la solución no se puede obtener de manera analítica y/o el tamaño del espacio de soluciones es muy grande para ser abordado por un método determinístico, dada su particular estrategia basada en poblaciones, donde es posible orientar la búsqueda en diferentes direcciones logrando escapar de óptimos locales y permitiendo un mejor rendimiento a la hora de aproximarse a la solución óptima.

En la década de los 40s, Alan Turing hace referencia a la búsqueda genética o evolutiva, en su reporte titulado *“Intelligent Machinery”* (Turing 1948, sección 12, página 18), al decir:

*“There is the genetical or evolutionary search by which a combination of genes is looked for, the criterion being the survival value.”*

Turing no especificaba como realizar o qué serie de pasos seguir para esta búsqueda genética o evolutiva, pero sí permitió abstraer la idea de aplicar los principios de darwinismo para resolver problemas de manera autónoma antes de la aparición de las computadoras (Eiben y Smith, 2015; Koza et al., 2005).

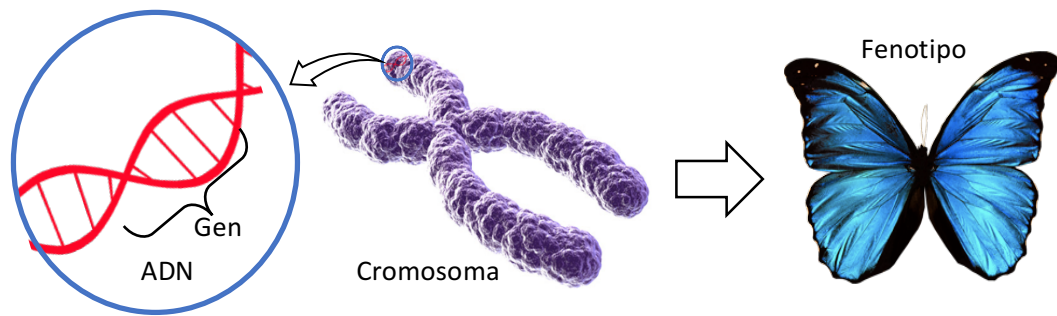
Dos décadas después, tres investigaciones diferentes estaban siendo desarrolladas, basados en la misma idea, en tres lugares diferentes. En Alemania, Rechenberg y Schwefel inventan las EE mientras que, en EEUU, L. Fogel, Owens y Walsh proponen la PE y John Hollan en colaboración con sus colegas y estudiantes en la Universidad de Michigan desarrollan los algoritmos genéticos, estas tres técnicas emergentes solo se vieron unidas hasta la década de los 90’s, cuando surge el CE y Koza introduce la programación genética.

En este capítulo aparte de la breve historia del cómputo evolutivo, definiremos algunos conceptos básicos para los algoritmos evolutivos (AE), profundizando en algoritmos genéticos y su aplicación para resolver problemas de optimización, particularmente, optimización con múltiples objetivos, describiremos algunos de los AE multiobjetivo (AEMO) clásicos y precisaremos el AEMO propuesto para el presente trabajo.

## 4.2 Conceptos básicos de Algoritmos evolutivos (AE)

Como punto de partida para entender el proceso de evolución natural, es importante definir algunos conceptos desde un punto de vista microscópico. La genética, según la definición de la Real Academia Española es: *“Parte de la biología que trata de la herencia y de lo relacionado con ella”*, precisa varios términos importantes muy utilizados en el CE, ya que sus significados son análogos. La genética aborda temas relacionados con la herencia de características de una generación a otra y como esta información esta codificada en el **ADN**.

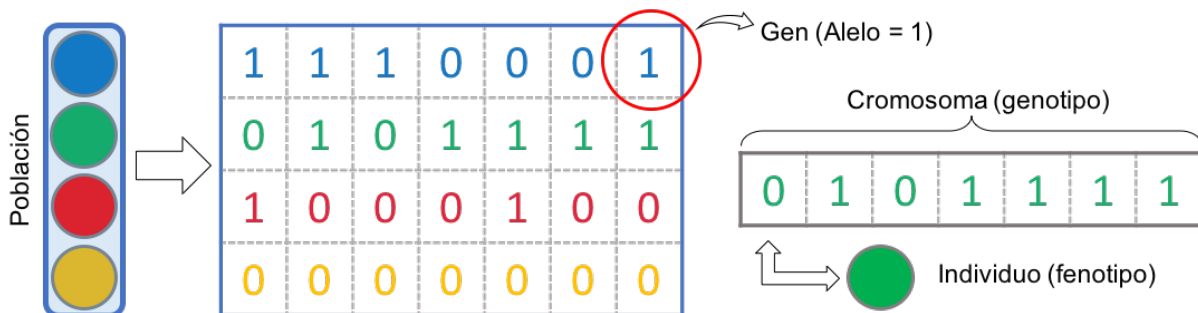
En términos generales los rasgos de un individuo u organismo a nivel externo o visibles, por ejemplo, en los humanos el color de cabello o la estatura, se conoce como **fenotipo**, el cual es una representación de un **genotipo** en un ambiente específico, este se compone de **genes** que solo pueden tomar ciertos valores definidos por los **alelos**. Los genes están codificados en  **cromosomas** o filamentos de ADN, cuyo número es constante para las células de cada especie (ver **Figura 8**).



**Figura 8.** Componentes principales de un genotipo que define los rasgos de un organismo, manifestados, según el ambiente al que está expuesto, como: colores, formas, y demás características visibles del individuo (fenotipo).

El **genoma** define la secuencia de nucleótidos que constituyen el ADN, es decir contiene toda la información genética de un organismo, dentro de una especie la mayor parte del material genético es igual así el fenotipo sea aparentemente muy diferente, esta manifestación puede tener dos opciones, que un gen afecte varios rasgos o que varios genes modifiquen una sola característica, en general pequeños cambios en el genotipo induce pequeños cambios en el fenotipo, como el color de los ojos.

La combinación del material genético entre el padre y la madre, es consecuencia de la fertilización al momento de fusionar los gametos (células sexuales), sin embargo, en CE la combinación de características de dos individuos se conoce como **cruzamiento**, continuando con la analogía de estos conceptos a términos computacionales, los individuos se forman como entidades dual a partir de un genotipo, que puede ser un arreglo de enteros, tuplas, cadenas de caracteres o simplemente números binarios, que representa una situación o problema específico (fenotipo). Los genotipos se forman como cromosomas, en los que cada gen puede tomar ciertos valores (alelos), por ejemplo, para un cromosoma binario cada gen solo puede tomar los valores de uno o cero (ver **Figura 9**).



**Figura 9.** Ejemplo de una representación binaria para un conjunto de individuos con diferentes características. Analogía entre los conceptos computacionales y su contraparte biológica, estructura de los datos y terminología.

Teniendo una estructura (genotipo) que puede ser modificada mediante **operadores genéticos**, es posible implementar una población que se define como un grupo de individuos de la misma especie o que comparten el mismo hábitat y se reproducen entre ellos, en donde estos operadores modificarán la información genética de una generación buscando mejorar las próximas generaciones, por ejemplo, individuos más fuertes, atractivos, con manos grandes, etc.

La clave para lograr mejorar la calidad de los individuos mediante herencia es la reproducción, dado que nuevos individuos son generados a partir de uno (reproducción asexual) o dos padres (reproducción sexual) que son **seleccionados** dentro de varios candidatos con alguna probabilidad (igual probabilidad para poblaciones panmíctica), en ambos casos el genoma del nuevo individuo no es idéntico al de los padres, ya que es posible la aparición de cambios o **mutaciones** del genotipo que se ven reflejadas como variaciones del fenotipo.

Sir Ronald Aylmer Fisher, considerado uno de los fundadores de la genética poblacional, inicia su obra con la frase: “La selección natural no es evolución”, haciendo referencia al papel del concepto de la supervivencia del más apto como parte del ciclo evolutivo, pero que ha sido confundido con evolución, sin demeritar la importancia de la teoría de Darwin (Fisher, 1930). Fisher introduce conceptos claves para comprender mejor el ciclo evolutivo, iniciando con la conservación de la variedad, la cual permite a una población posibilitar la supervivencia de individuos poco aptos pero que pueden aportar con algún componente genético de alto valor para futuras generaciones, por otro lado los cambios poblacionales se producen cuando el número de individuos se incrementa haciendo que probablemente se aumente el número de “buenos” individuos lo que supone más competencia, es entonces necesario adaptar este tipo de análisis a los métodos computacionales, definiendo el tiempo de vida de los individuos, posibilidades de supervivencia o métodos de recambio generacional, todo esto en pro de lograr un ciclo evolutivo fructífero para el problema planteado.

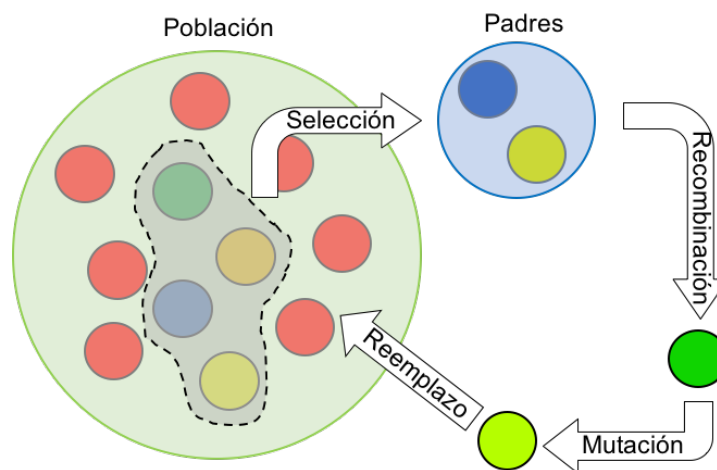
#### 4.2.1 Componentes de un AE

Para poder conocer que conforma un AE, es necesario resolver la pregunta ¿Qué es un algoritmo evolutivo? Eiben y Smith (2015) mencionan que, aunque hay variantes en los AE, en general la idea detrás de estas técnicas es la misma: Dado un conjunto de individuos en una población que comparte los recursos limitados en algún ambiente o contexto específico, la competencia por estos recursos provoca la selección natural (supervivencia del más apto).

Por lo tanto, podemos definir un AE como:

Dada una función objetivo (**función de calidad o aptitud**) que debe ser maximizada, podemos crear un conjunto de posibles soluciones (población de individuos) de manera aleatoria, como elementos del dominio de la función que al ser evaluados permite obtener de manera abstracta un valor de aptitud (el valor más alto es el mejor individuo). Con base en estos valores de calidad, las mejores soluciones son seleccionadas como semillas para la siguiente generación, para luego aplicarles algún operador genético de recombinación y/o mutación. Recombinación se aplica a dos o más candidatos seleccionados (también conocidos como padres o semillas), produciendo una o más posibles soluciones nuevas (hijos o descendencia). La mutación tiene la capacidad de modificar cada posible solución generando nuevos candidatos (hijos modificados).

Entonces después de aplicar recombinación y mutación se produce un conjunto de nuevas posibles soluciones (descendencia), quienes tienen un valor de aptitud que les permite competir por un lugar en la siguiente generación (supervivencia). Se repite este proceso hasta que un individuo alcance el valor de aptitud deseado (solución óptima) o presente suficiente calidad como para ser considerado como solución, adicional es posible definir algún límite computacional como: máximo número de iteraciones, cantidad de memoria o CPU utilizada, el valor máximo de aptitud se mantiene durante un largo periodo de tiempo (número de iteraciones o tiempo de ejecución) o cuando la diversidad de la población baja de un nivel establecido (ver **Figura 10**).



**Figura 10.** Ciclo evolutivo tradicional para un AG con población panmítica.

Los algoritmos evolutivos se componen de varios procesos que en su mayoría son estocásticos, por ejemplo, durante la creación aleatoria de las posibles soluciones iniciales o en el proceso de selección, teniendo en cuenta que los mejores individuos no se obtienen de manera determinística, facilitando que los individuos “débiles” conserven alguna oportunidad de ser padres o sobrevivir para la siguiente generación.

En la recombinación y mutación, la magnitud y el lugar donde se realizará la modificación se define de manera aleatoria. Este tipo de decisiones permiten aumentar la diversidad en la búsqueda, promoviendo el proceso de adaptación al entorno por parte de la población. El proceso evolutivo resulta entonces cuando una población puede adaptarse de manera incremental al entorno.

El pseudocódigo de un AE de manera genérica puede ser:

#### Algoritmo 1 Algoritmo evolutivo

- 
- 1: **Inicialización:** Se crean  $N$  posibles individuos ( $x$ ) para una población inicial ( $\mathbb{X}$ ).  
 $x_1, \dots, x_N \in \mathbb{X}$
  - 2: **Evaluación:** Se evalúan los candidatos con la función de aptitud.  $x_i \rightarrow f_{aptitud}(x_i), i = 1, \dots, N$
  - 3: **Mientras** (La condición de terminación  $\neq$  verdadero)
  - 4:   **Seleccionar** los  $j$  padres de  $\mathbb{X}$ .  $\mathbb{P} = \{p_1, \dots, p_j\}$
  - 5:   Crear los  $k$  hijos mediante **recombinación**.  $\mathbb{H} = \{h_1, \dots, h_k\}$
  - 6:   **Mutar**  $h_i$
  - 7:   **Evaluar**  $h_i \rightarrow f_{aptitud}(h_i), i = 1, \dots, k$
  - 8:   Seleccionar individuos de  $h_i \cup \mathbb{X}$  para la nueva generación  $\mathbb{X}'$
  - 9:    $\mathbb{X} \rightarrow \mathbb{X}'$
  - 10: **Fin**

#### 4.2.2 Representación del individuo

El primer paso para poder diseñar un algoritmo evolutivo (AE) es unir el mundo real con el computo evolutivo, lo que significa crear una estructura de datos que represente las características del problema y su contexto, la cual debe estar diseñada de tal manera que un sistema computacional pueda manipularla, en otros términos, toda la información inherente al problema y su contexto son los fenotipos y su codificación, crea una estructura de datos en un entorno de AE que contiene los genotipos (ver **Figura 9**).

Como se ha mencionado anteriormente, existen varios AE, con diferencias como la representación de los individuos, particularmente para los algoritmos genéticos (AG) es común utilizar caracteres de un alfabeto finito, en las estrategias evolutivas se utilizan vectores de números reales, máquinas de estado finito para la programación evolutiva clásica y representación con árboles para la programación genética (Eiben y Smith, 2015; Goldberg, 1989).

### 4.2.3 Población

Una población es un conjunto de individuos de la misma especie, por lo que en el contexto de AE es un multiconjunto de cromosomas. La población se convierte entonces en la unidad del proceso evolutivo, dado que los individuos no pueden evolucionar por si solos, necesitan la interacción con sus pares para lograr cambios o adaptarse. Inicializar una población es una tarea simple, teniendo en cuenta que solo es necesario definir cuantos individuos contiene o el tamaño.

En la mayoría de las aplicaciones de AE el tamaño de la población permanece constante durante todo el proceso evolutivo, aumentando la competencia, dada la necesidad de sobrevivir en el recambio generacional. En cuanto a tipos de poblaciones, en implementaciones clásicas es común utilizar poblaciones de tipo panmíctica, en las que el apareamiento es libre y al azar (con igual probabilidad). Por otra parte, en algoritmos más sofisticados, típicamente se utilizan poblaciones distribuidas en islas o vecindarios, lo que reduce el número de individuos con los que se puede interactuar y se comparte información genética entre subpoblaciones cada cierto tiempo.

La idea principal detrás de esta limitación es aumentar la diversificación de la población en general e incrementar la explotación en los grupos de individuos que están distribuidos. La diversidad de la población es un concepto muy utilizado en AE y representa la cantidad de individuos diferentes dentro de la población, comúnmente asociado a individuos con diferente valor de aptitud, sin embargo, no necesariamente es así, puesto que diferentes genotipos pueden tener el mismo rendimiento en términos de calidad. Herramientas estadísticas como la entropía pueden ayudar a medir la diversidad de una población específica o simplemente el número de fenotipos y/o genotipos diferentes.



#### 4.2.4 Función para evaluación de aptitud (función de calidad)

En términos generales la función objetivo para un problema de optimización, que define las condiciones de optimalidad, puede ser la misma función de calidad (en el dominio del AE) o en algunos casos requerir alguna modificación, que permite evaluar el rendimiento de los individuos, es decir que tan “bien” el candidato satisface las condiciones, calculando un valor de aptitud para los genotipos que describe un mayor o menor nivel en los fenotipos según sea el caso. La mayoría de problemas de optimización requieren ser minimizados (costos, esfuerzo, distancia, etc.), pero en términos de calidad, es comúnmente asociado con maximización, ya que un valor más alto supone un mejor rendimiento. Sin embargo, pasar de un problema de minimización a maximización y viceversa es una tarea simple, permitiendo poder abordar cualquier problema de optimización con AE.

#### 4.2.5 Mecanismo para la selección de padres

Selección por torneo o ranking son los mecanismos más populares implementados en AE, los cuales son implementados de manera probabilística de tal manera que los mejores individuos tengan una probabilidad mayor de ser elegidos como padre, a diferencia de los individuos con bajo rendimiento, no obstante, estos individuos siempre cuentan con la oportunidad (pequeña en la mayoría de casos) de ser seleccionados, de no ser así, la diversidad en la población se vería reducida y la búsqueda se convertiría en un método avaro y elitista que solo se evoluciona en torno a la primer mejora encontrada, lo que puede ocasionar estancamientos en óptimos locales disminuyendo la posibilidad de lograr la solución óptima.

#### 4.2.6 Recombinación

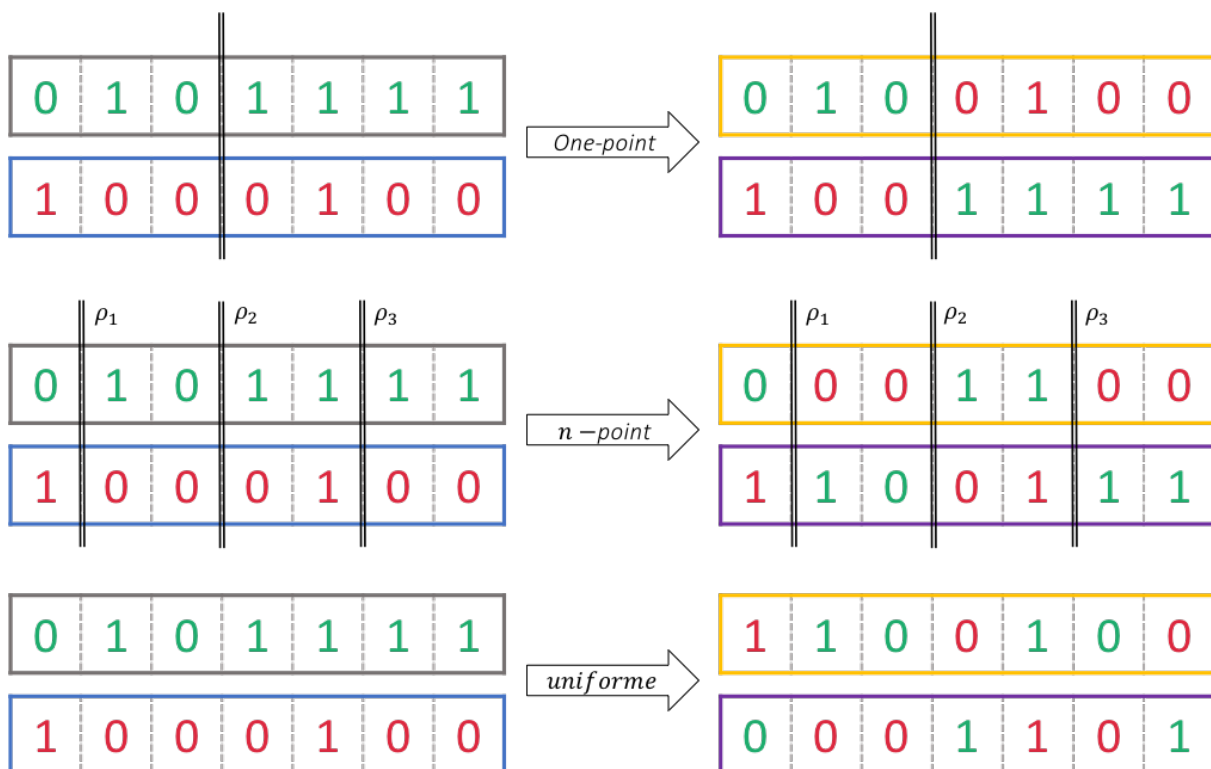
El operador genético conocido como recombinación o cruzamiento combina la información genética de dos o más individuos (padres) en uno o más descendientes (hijos). La mayoría de estos métodos están restringidos al uso de dos padres, aunque está demostrado que el uso de multipadres, así no se tenga una contraparte biológica, provee efectos positivos para el algoritmo (Eiben, 2003).

El proceso de recombinación busca distribuir los genes “buenos” de cada padre en los nuevos individuos, combinando las mejores características de cada uno, sin embargo, esta teoría no siempre se cumple, ya que en AE se realiza de manera probabilística, es importante resaltar que para cada representación existen

diferentes metodologías de recombinación, para el presente trabajo nos interesan los métodos utilizados para AG en representación binaria o entera.

Los operadores de recombinación pueden realizar segmentación y combinación del material genético de los padres o realizar operaciones aritméticas que modifiquen la metodología para crear el nuevo genotipo a partir de las semillas utilizadas.

Holland propone el método “*One-point crossover*” o cruzamiento en un punto, el cual primero calcula un número aleatorio ( $\rho$ ) en un rango entre  $[1, l - 1]$ , donde  $l$  representa el tamaño de la estructura de datos de la codificación (cromosoma), para luego utilizarlo como punto de cruce, dividiendo los dos padres, cabeza antes de  $\rho$  y cola después de  $\rho$ , los hijos entonces son la unión de la cabeza de un padre con la cola del otro (Holland, 1992). Este método se puede generalizar en “*n-point crossover*”, el cual divide el individuo en más de dos segmentos y reconstruyendo el material genético de cada padre alternando los segmentos para formar la descendencia; se selecciona un número  $n$  de puntos de corte como se ilustra en la **Figura 11**.



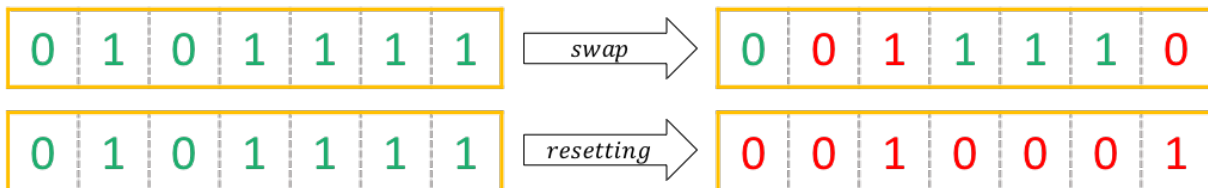
**Figura 11.** “*One-point crossover*” (arriba) con un  $\rho = 3$ , “*n-point crossover*” (medio) para un  $n = 3$ ,  $\rho = [1,3,5]$  y cruzamiento uniforme (abajo) para un arreglo de números aleatorios =  $[0.1, 0.7, 0.8, 0.2, 0.9, 0.8, 0.2]$  y  $\rho = 0.5$ .

Por último el método conocido como cruzamiento uniforme, propone tratar cada gen de manera independiente y realizar una selección para cada hijo que determina el valor del alelo(s) en el cromosoma, haciendo una elección aleatoria que determina que padre hereda este valor, en otras palabras, se define un valor para  $\rho$  entre 0 y 1 (comúnmente 0,5) luego de manera aleatoria se obtiene otro valor entre 0 y 1 el cual puede ser mayor a  $\rho$  indicando que debe tomar el material genético del padre 1 o del padre 2 para el caso que sea menor o igual a  $\rho$  (ver **Figura 11**).

Existen varios métodos de recombinación adicionales tanto para representaciones con permutación de enteros o representación con números reales, algunos populares como PMX y “Cut-and-crossfill” por lo que se recomienda revisar (Eiben y Smith, 2015; Goldberg, 1989).

#### 4.2.7 Mutación

El segundo método principal de cambios evolutivos es la mutación, el cual es un operador unario que modifica directamente el material genético de un hijo o descendencia. Como salida, un “mutante” siempre depende de una serie de elecciones aleatorias por lo tanto la mutación es un proceso completamente estocástico. Este operador es comúnmente utilizado en diferentes áreas del CE, por ejemplo, en PE es el único operador que se utiliza o en AG es el encargado de proporcionar nuevo material genético, útil para la explotación y exploración de la población. El método de mutación conocido como “*Bitwise*” (similar al método “swap” en representaciones por permutación de enteros), propone la posibilidad (con probabilidad  $p_m$ ) de intercambiar el valor en uno o  $|\kappa|$  –genes, determinado de manera aleatoria una cantidad de valores a  $k_i$  entre 1 y el tamaño del cromosoma como posiciones del gen a modificar, con un nuevo valor seleccionado de manera aleatoria de algún conjunto de valores permitidos. Este método funciona tanto para representaciones enteras como binarias, fácilmente modificable a un método conocido como “*random resetting*” o “*scramble*” (para una representación de permutación de enteros), en el que siempre se intercambian todos los genes del cromosoma (ver **Figura 12**).



**Figura 12.** Mutación “*swap*” (arriba) para un  $\kappa = [2,3,7]$ , “*random resetting*” (abajo) para un cromosoma de con representación binaria.

#### 4.2.8 Mecanismo de selección de los sobrevivientes (Reemplazo)

La estrategia de reemplazo funciona de manera similar al mecanismo de selección de los padres, utilizando ranking o torneo para decidir que individuos deben dejar la población y dar paso al nuevo material genético que contienen los hijos, se conoce como selección de sobrevivientes, dado que para un caso en el que el número de hijos es más de la mitad de la antigua población, es necesario eliminar un gran número de individuos que pueden ser seleccionados por edad, bajo desempeño o de manera aleatoria. Cuando se generan dos hijos por generación, es posible hablar de reemplazo, ya que por torneo binario se puede definir quien continua o no formando parte de la población a evolucionar.

#### 4.3 AE con enfoque multiobjetivo (AEMO)

Como se mencionó anteriormente los AE son una herramienta útil para problemas de optimización, utilizando la función objetivo como función de calidad (o con alguna modificación) para un caso con un único objetivo, realizando cierta cantidad de iteraciones que permitan que uno o más individuos alcancen algún valor aproximado aceptable o la solución óptima para el mejor de los casos. La estructura de funcionamiento de los AE basada en exploración y explotación, facilita escapar de óptimos locales y direccionar la búsqueda hacia el óptimo global.

El problema entonces surge a partir de la discusión de cómo comparar dos individuos que tienen un rendimiento diferente para cada una de las funciones de aptitud para un POMO, es decir que compensación produce cada uno de los individuos para los objetivos en conflicto, por ejemplo, para el problema clásico de la mochila ("*knapsack problem*") con enfoque multiobjetivo, en el que los objetivos en conflicto son aumentar las ganancias y disminuir el peso, cada posible solución codificada bajo alguna representación (estructura de datos del cromosoma) simboliza un conjunto de ítems con un peso y valor asociados, por lo tanto algún individuo puede contener el peso mínimo pero no la mejor suma de ganancias, mientras que otra posible solución maximiza las ganancias pero el peso excede el valor deseado; cada candidato realiza una compensación entre los objetivos, complicando la decisión que puede definir cuál es la mejor solución al problema planteado. Es posible buscar una solución a este problema con métodos como la combinación lineal de objetivos, ordenamiento lexicográfico o alguno de los detallados en el Capítulo 3 del presente documento, sin embargo, los AE permiten lidiar con POMO gracias a su estrategia de evolución de un multiconjunto de posibles soluciones (población) simultáneamente. Esta interacción entre candidatos a solución del POMO permite intercambiar las

mejores características o habilidades de cada individuo, facilitando crear una posible solución en cada recambio generacional que se aproxime al frente de Pareto (ver Capítulo 3). Schaffer surge entonces como pionero con el algoritmo “*Vector Evaluated Genetic Algorithm*” (VEGA) modificando los operadores genéticos en un AG, realizando ciclos de selección independientes para cada objetivo (uno por uno) hasta obtener un subgrupo de individuos a los que aplica cruzamiento y mutación (Schaffer, 1984).

Sin embargo, se hace necesario encontrar una métrica para evaluar los individuos bajo múltiples funciones objetivos, ya que los AE utilizan la función de calidad para la reproducción y evolución en el caso mono objetivo. Goldberg (1989) introduce el término de dominancia (ver Capítulo 3) y “*niching*” (operador utilizado para controlar la diversidad de la población) con la intención de subdividir la población en grupos en los que no es posible definir cuál es mejor para todos los objetivos en conflicto, surge entonces el uso de conjuntos no dominados como metodología para direccionar la búsqueda y evolución de un AE para POMO. Encontrar el frente de Pareto de un POMO es una tarea difícil en la mayoría de casos, por lo que se permite encontrar una “buena” aproximación en un tiempo de ejecución razonable. Las metodologías basadas en AE (AEMO) presentan un buen rendimiento y aunque no es posible considéralas como un método universal, “*No free lunch theorem*” (Wolpert y Macready, 1997), en general poseen una implementación sencilla que permite guiar la búsqueda con información obtenida en el dominio del problema, lo que acelera su desarrollo, ejecución, evaluación y entendimiento.

#### 4.3.1 Clasificación de los AEMO

En la literatura normalmente se dividen a los AEMO en dos generaciones marcadas por el uso de un mecanismo de selección conocido como “*preservación elitista*” o sencillamente elitismo. La inclusión del elitismo proporciona un rendimiento no degradante, debido a que se mantienen a los individuos con mejor rendimiento durante cada recambio generacional. El uso e implementación de metodologías para aplicar el elitismo es la diferencia principal entre los algoritmos que forman parte de la segunda generación. En los últimos años se han clasificado los AEMO en tres categorías: los basados en **dominancia**, los métodos por **descomposición** y los que utilizan algún **indicador**. Los primeros basados en el concepto propuesto por Goldberg como: *Strength Pareto Evolutionary Algorithm (SPEA, SPEA2)*, *Nondominated Sorting GA (NSGA, NSGA-II)* o *Pareto Envelope-based Selection Algorithm (PESA, PESA II)* (algunos algoritmos populares de segunda generación) utilizan la dominancia como se describió anteriormente, Después encontramos algoritmos como *MOEA/D (Multi-objective evolutionary algorithm based on decomposition)*, que utilizan el método de descomposición, dividiendo un POMO en varios problemas con un único objetivo y que son

resueltos simultáneamente, los cuales han ganado bastante atención por los investigadores debido a las dificultades que presentan los AEMO cuando se abordan problemas con cuatro o más objetivos, estas complicaciones están directamente relacionadas al uso de los conceptos de Pareto al intentar discriminar un conjunto de soluciones únicamente utilizando la dominancia (Von Lücken et al., 2014). Por último los que utilizan algún indicador por ejemplo, *HypE (Hypervolume Estimation Based MOGA)*, que utiliza la métrica hipervolumen como valor de aptitud para los individuos en un AE (Deb et al., 2016). Dada la dimensión del problema abordado en este trabajo (dos objetivos en conflicto) se decide utilizar una versión adaptada de ***MOCeII (MultiObjective Cellular genetic algorithm)*** (Nebro et al., 2009) como propuesta inicial y comparar los resultados obtenidos contra un AEMO muy popular en diversos trabajos de aplicación a problemas reales como lo es ***NSGA-II (Nondominated Sorting Genetic Algorithm II)*** (Deb et al., 2002).

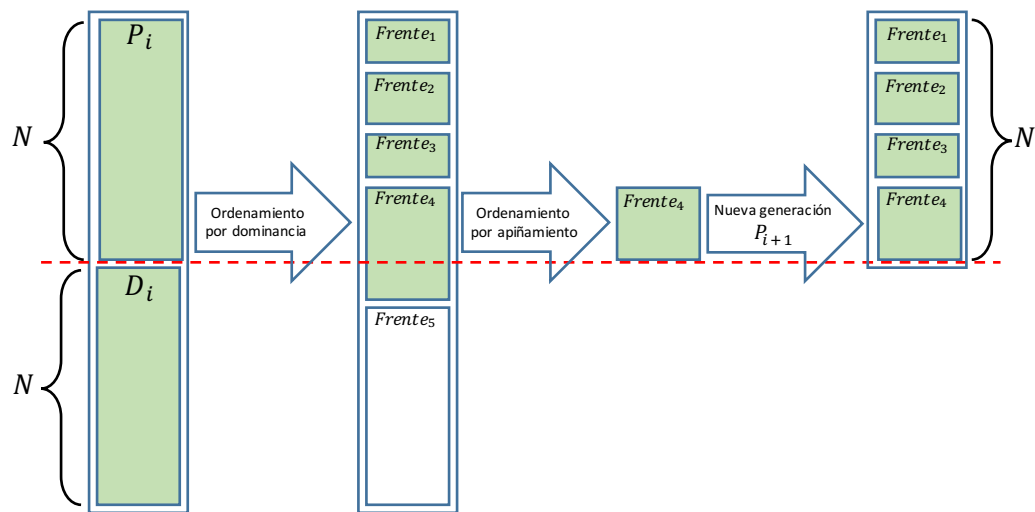
#### 4.3.2 Nondominated Sorting Genetic Algorithm II (NSGA-II)

La relación de *dominancia* descrita en el Capítulo 3 sección 3.2.2 permite relacionar dos soluciones o individuos, utilizada por varios algoritmos como un operador binario, en el que una solución  $A$  domina o no a una solución  $B$ ,  $dominancia(A, B) = 1$  o  $dominancia(A, B) = 0$ , respectivamente. Coello et al. (2007) definen tres posibles métodos que utilizan el concepto de dominancia para hacer ordenamiento:

- ***Ranking por dominancia:*** el método se basa en definir un nivel para cada individuo mediante el cálculo de la cantidad de individuos en la población actual por la cual está dominado. Algoritmos: “*Multi-Objective Genetic Algorithm*” (MOGA) y “*Niched Pareto Genetic Algorithm*” (NPGA)
- ***Dominancia por conteo:*** el criterio se basa en cuántas posibles soluciones son dominadas por algún individuo. Algoritmos: “*Strength Pareto Evolutionary Algorithm*” (SPEA y SPEA2)
- ***Dominancia por profundidad:*** este método distribuye los individuos en “frentes” de dominancia, para luego ordenarlos de tal manera que el frente uno contiene los mejores individuos. Quienes dominan al resto de la población. Algoritmos: “*Nondominated Sorting Genetic Algorithm*” (NSGA y NSGA-II).

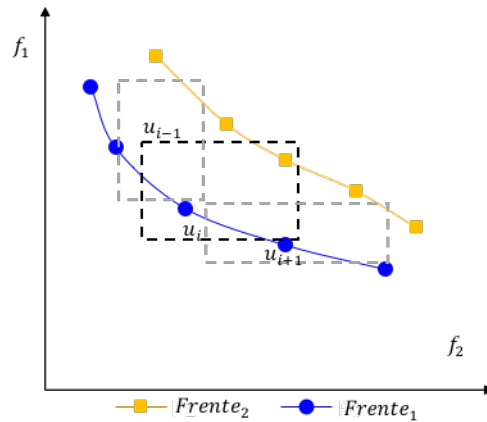
NSGA-II es uno de los AEMO más populares desarrollado por Deb et al. (2002) el cual utiliza el principio de elitismo (AEMO de segunda generación) y la relación de dominancia en profundidad, además de un mecanismo que preserva la diversidad conocido como “*crowding distance*” o distancia de apiñamiento. Para cada generación el algoritmo combina la población actual ( $P_i$ ) de tamaño  $N$  con la descendencia ( $D_i$ )

resultante para cada iteración  $i$  ( $N$  número de hijos), obtenidos con los operadores genéticos tradicionales (selección, recombinación y mutación). Luego se construye una nueva población ( $Q_i$ ) de tamaño  $2N$  la cual es clasificada en “frentes” no dominados, luego en cada frente se calcula la distancia de apiñamiento con la finalidad de poder clasificar a los individuos de un mismo frente en un ranking determinando que individuo aporta más o menos a la diversidad del frente; por lo tanto, tenemos cada individuo con un vector de  $k$  valores de aptitud (para un POMO de  $k$  objetivos) un frente asociado y un valor de distancia de apiñamiento. Por último, se construye una nueva población ( $P_{i+1}$ ) seleccionando los individuos de mejor a peor, es decir, los que pertenecen a los mejores frentes y los que generan mayor diversidad (ver **Figura 13**).



**Figura 13.** Cuadro esquemático de los pasos que ejecuta NSGA-II para producir una nueva generación  $P_{i+1}$ .

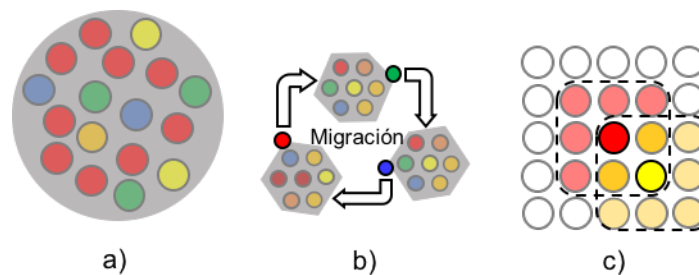
La distancia de apiñamiento es una medida del espacio objetivo alrededor de una solución  $u$  que no es ocupada por ninguna otra solución, por lo tanto, sencillamente se calcula el perímetro de un rectángulo construido en el espacio objetivo utilizando los vecinos inmediatos ( $u - 1$  y  $u + 1$ ) como diagonal (ver **Figura 14**). Aunque NSGA-II es un algoritmo relativamente viejo, sigue siendo muy utilizado para problemas con pocos objetivos en conflicto, ya que recientes estudios proponen que el uso del indicador de dominancia es inútil al trabajar una cantidad de objetivos alta. En este caso la presión de selección que utiliza NSGA-II (y los demás algoritmos similares) solo puede encontrar un grupo reducido de soluciones, estancándose sin posibilidad de enfocar la búsqueda hacia el conjunto óptimo de Pareto.



**Figura 14.** Cálculo de la distancia de apiñamiento (“*crowding distance*”), para los valores extremos del frente se asigna una distancia infinita.

#### 4.3.3 MultiObjective Cellular genetic algorithm (MOCeLL)

Varios algoritmos evolutivos aplican los operadores genéticos en una población donde todos los individuos tienen la misma probabilidad de ser seleccionados (población panmítica) o con algún tipo de preferencias sobre los individuos con mejor rendimiento (elitismo). Por otra parte, existe un gran número de implementaciones de algoritmos de manera distribuida, en islas conectadas, por ejemplo (ver **Figura 15**).

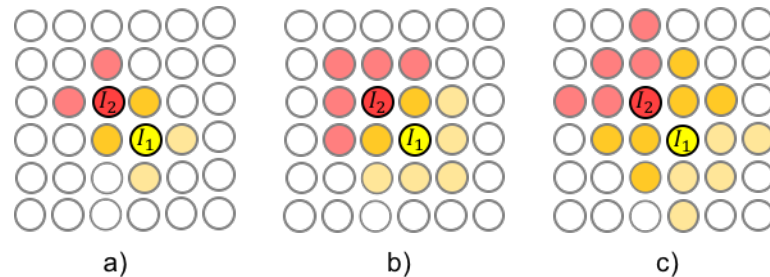


**Figura 15.** Diferentes tipos de poblaciones utilizadas en algoritmos evolutivos a) Panmítica, b) Islas conectadas, c) Vecindarios superpuestos.

MOCeLL (“*MultiObjective Cellular genetic algorithm*”) es un algoritmo propuesto por Nebro et al. (2009) el cual distribuye los individuos de una población en una malla, para luego asignar un vecindario a cada individuo (el tamaño del vecindario puede cambiar según lo requiera el problema, ver **Figura 16**). Se utiliza una malla toroidal con el fin de garantizar que todos los individuos tengan la misma cantidad de vecinos. Este algoritmo de tipo celular realiza esta clase de distribución de la población con el fin de restringir la cantidad de individuos que pueden interactuar entre sí, estas restricciones promueven la exploración del espacio de búsqueda de manera que se transmiten genes entre vecindarios sin necesidad de enfocarse únicamente en el mejor individuo de cada subgrupo, manteniendo una alta diversidad debido a la difusión

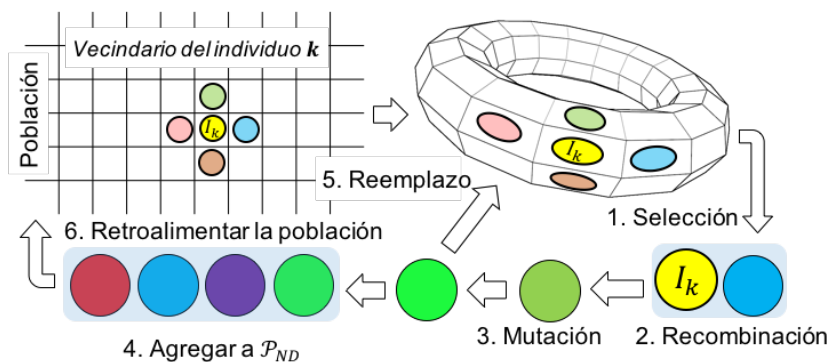


“lenta” del material genético. Por otra parte como los individuos solo pueden interactuar con un reducido grupo de vecinos, se busca fomentar la explotación en cada vecindario, utilizando las técnicas de ranking y distancia de apiñamiento propuestas por Deb et al. (2002) en NSGA-II en cada vecindario y así fomentar el elitismo a la hora de aplicar los operadores genéticos.



**Figura 16.** Vecindarios diferentes, a) cuatro, b) ocho, c) doce, para dos individuos  $I_1 = (x, y)$  e  $I_2 = (x - 1, y - 1)$ .

MOCCell construye un conjunto ( $\mathcal{P}_{ND}$ ) de  $N$  individuos no dominados (la cardinalidad del conjunto implica el impacto del elitismo en el algoritmo) ordenados por “*crowding distance*”, con los que se retroalimenta la población. El algoritmo no utiliza el concepto de recambio generacional de la misma manera que los demás AE, en este algoritmo es necesario revisar individuo por individuo para completar una generación de la población, es decir en cada evaluación de un individuo se realiza un torneo entre los vecinos para elegir los padres o semillas, los cuales se utilizarán en el proceso de recombinación para luego emplear el operador de mutación con los hijos resultantes. Una vez finalizado este proceso se hace un ranking entre los hijos, los padres seleccionados y el individuo que está siendo evaluado con el fin de definir quién ocupará este espacio en la población (el de mejor rendimiento) y continuar con el siguiente individuo. Por último, al finalizar la revisión de todos los individuos, de manera aleatoria se seleccionan  $N$  individuos que serán intercambiados por lo que se encuentran alojados en el conjunto  $\mathcal{P}_{ND}$  (ver **Figura 17**).



**Figura 17.** Ciclo de reproducción para un algoritmo genético celular con cuatro vecinos por individuo y un conjunto no dominado de cuatro individuos.

## 4.4 AEMO propuesto

Para abordar el problema planteado en el Capítulo 2 sección 2.6 se propone un AE, específicamente un algoritmo genético de tipo celular, basado en MOCeL en su versión asíncrona (ver **Algoritmo 2**).

---

### **Algoritmo 2.** Pseudocódigo de la metaheurística basada en MOCeL.

---

1.  $data = \text{setup}()$ ; /\*Configuración de parámetros del algoritmo y datos de entrada\*/
  2.  $pop = \text{popGen}()$  /\*Crea la población inicial\*/
  3.  $pop = \text{toroidal}(pop)$ ; /\*Distribuye los individuos en una malla toroidal\*/
  4.  $\mathcal{P}_{ND} = []$ ; /\*Crea un conjunto para almacenar los individuos no dominados\*/
  5. **Mientras** (condición Determinación==falso) **hacer**
  6.     **Para**  $k = 1$  **hasta**  $|pop|$  **hacer**
  7.          $I_k = pop(k)$ ; /\*individuo  $k$ \*/
  8.          $vPop = \text{vecindario}(pop, I_k)$ ;
  9.          $padres = \text{selección}(vPop)$ ;
  10.          $hijos = \text{recombinación}(data, padres)$ ;
  11.          $hijos = \text{mutación}(data, hijos)$ ;
  12.          $pop(k) = \text{reemplazo}(vPop, hijos)$ ;
  13.          $\mathcal{P}_{ND} = \text{agregar}(pop(k), \mathcal{P}_{ND})$ ;
  14.     **Fin**
  15.  $pop = \text{retroalimentación}(\mathcal{P}_{ND})$ ;
  16. **Fin**
- 

### 4.4.1 Representación (codificación)

La representación o codificación de las posibles soluciones está expresado como un arreglo de números enteros que simbolizan el tipo de vehículo asignado para cubrir un viaje con una hora de partida estipulada por la posición del número entero en el arreglo, en otros términos, la hora de partida del vehículo la define la posición del gen en el cromosoma y el tipo de vehículo está determinado por el alelo en este gen.

El tamaño del arreglo está definido a partir de un estudio previo de la demanda de pasajeros en una ruta específica, haciendo un análisis preliminar de la frecuencia mínima de operación para la capacidad promedio de los vehículos disponibles para cubrir la ruta ( $\overline{CAP}$ ). El método seleccionado calcula una frecuencia de operación ( $F_j$ ) a partir del perfil de carga por periodos de tiempo, fundamentado en los mecanismos discutidos en el Capítulo 2 sección 2.1 para el cálculo de la frecuencia mínima de operación, los valores obtenidos se consideran como una cota inferior de la frecuencia, ya que puede ser modificados los intervalos por el AE.

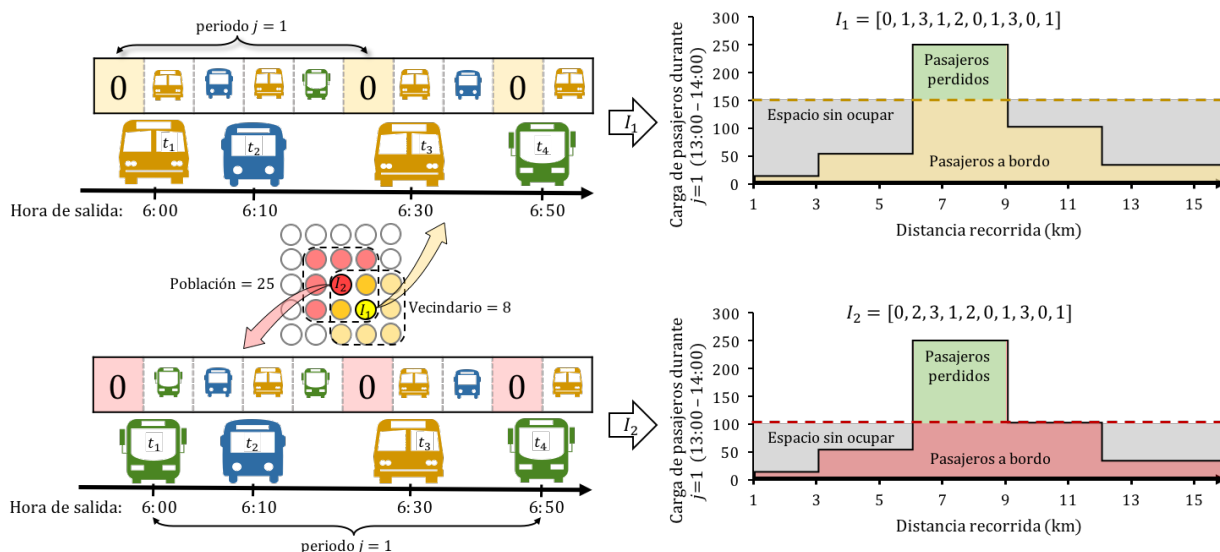
Una vez que se tiene el horario preliminar, para la creación de los individuos se divide el cromosoma con una serie de genes con valor cero (valor del alelo) que sirven para marcar los inicios de cada periodo de tiempo con el fin de poder evaluar el rendimiento del individuo en intervalos durante la duración total del funcionamiento y así tener una mejor idea del comportamiento para los dos objetivos en conflicto (costo de operación y pérdida de la calidad de servicio), la distribución de los ceros puede ser modificada, pero no pueden existir dos ceros consecutivos dada la restricción del problema que establece una frecuencia mínima de operación ( $f_{min}$ ), todos los periodos de tiempo tienen la misma duración y pueden estar relacionados con el tiempo total que utiliza un vehículo para cubrir un viaje (ver **Figura 18**).

El método utilizado para construir el cromosoma está definido por:

$$F_j = \max \left[ \frac{A_j}{LF_j \cdot CAP \cdot L}, \frac{P_j^{max}}{CAP}, f_{min} \right] \quad (32)$$

$$A_j = \sum_{s \in R} P_j^s \times \ell_s \quad (33)$$

$$L = \sum_{s \in R} \ell_s \quad (34)$$

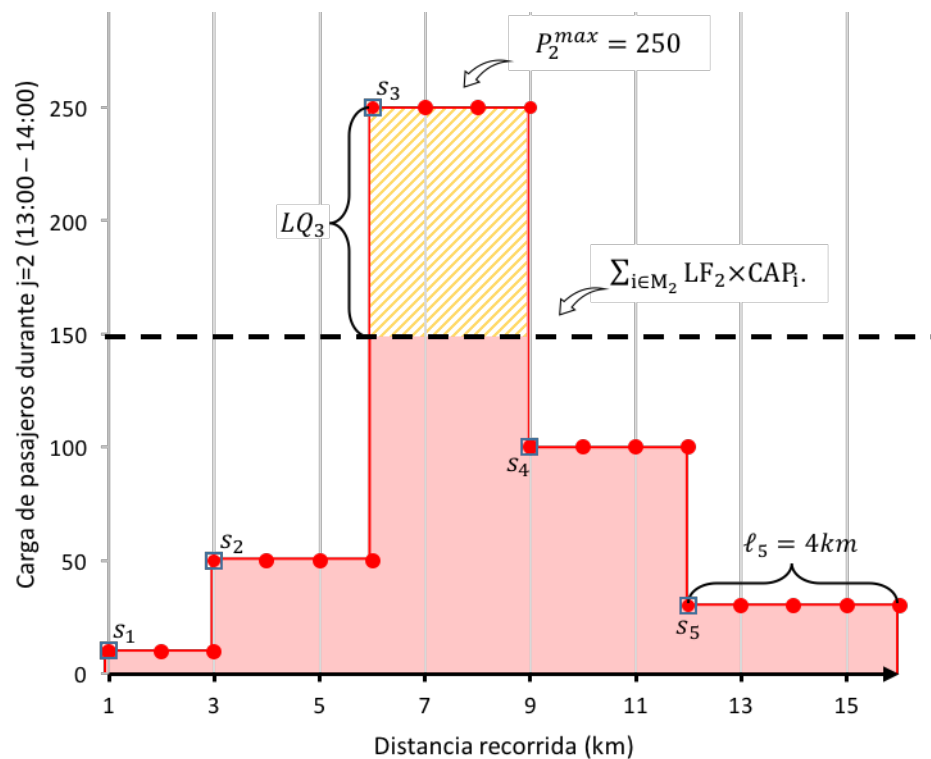


**Figura 18.** Representación para dos individuos vecinos y su respectivo comportamiento para un periodo de tiempo y una carga de pasajeros definida.

Donde  $P_j^s$  es el flujo o cantidad de pasajeros en la parada  $s$  de la ruta  $R$  ( $P_j^{max}$  para el caso máximo),  $A_j$  es el área bajo la curva formada por la cantidad de pasajeros a lo largo de toda la ruta.  $LF_j$  es el factor de carga definido para el periodo  $j$  o la máxima ocupación permitida en este periodo de tiempo.  $\ell_s$  es la distancia entre la parada  $s$  y la parada  $s + 1$ , que al sumarlas entregan la distancia total recorrida  $L$  para la ruta  $R$  (ver **Figura 19**).

#### 4.4.2 Funciones objetivo y evaluación de aptitud

El POMO es formulado proponiendo un vector con dos funciones objetivo en conflicto  $\mathbb{F} = [f_1, f_2]$ , la primera función representa el costo asociado a la operación de una flota de vehículos de diferentes tipos establecidos para cubrir un conjunto de viajes necesarios en una ruta específica, el cual debe ser minimizado como objetivo del problema. Por otro lado, también se desea minimizar la función  $f_2$  la cual evalúa la pérdida de la calidad del servicio cuando la demanda de pasajeros no se puede satisfacer en cada periodo de tiempo.



**Figura 19.** Perfil de carga para una ruta con seis paradas  $s_i, i = 1, \dots, 6$ , una carga máxima de 250 pasajeros en la tercera parada, una capacidad de vehículos total igual a 200 y un factor de carga  $LF_2 = 75\%$ .  $LQ_3 = 100$  indica que en la tercera parada se están perdiendo cien pasajeros ya que no pueden ser transportados por la flota asignada.

La ecuación para  $f_1$  (ver Capítulo 2 sección 2.6) consiste en la sumatoria de todos los costos para cada vehículo asignado a un viaje, el cual es calculado dependiendo del calendario (horario) y el costo de las variables dependen del tipo de vehículo.  $f_2$  indica la cantidad de pasajeros por kilómetro que no pueden ser transportados por la flotilla que debe circular en cada periodo de tiempo (ver **Figura 19**).

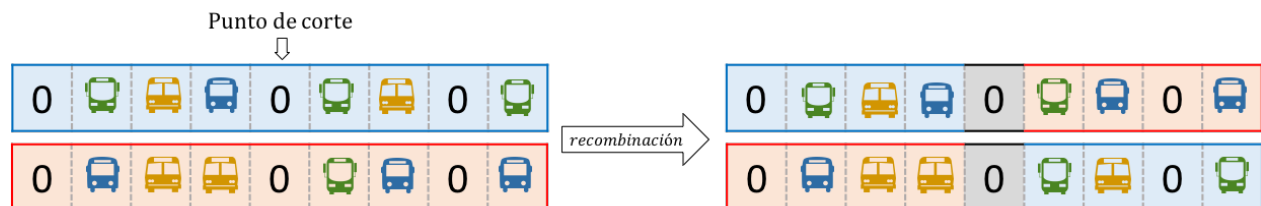
Los valores de aptitud que entregan las funciones son normalizados, debido a que es posible conocer el costo máximo de operación al seleccionar únicamente los vehículos más costosos disponibles para cubrir todos los viajes del calendario, por otra parte, cuando se seleccionan solo los vehículos con menor capacidad, la pérdida de pasajeros será la más alta posible para un conjunto de vehículos disponibles.

#### 4.4.3 Operadores genéticos

**Inicialización de la población:** el conjunto de individuos inicial (población) es generado de manera aleatoria asignando diferentes tipos de vehículos disponibles (proporcionados como parámetro de entrada del algoritmo), teniendo en cuenta el tamaño del horario calculado previamente y la distribución de los ceros, después de esto se procede a distribuir los individuos en una malla toroidal.

**Selección:** se realiza una selección por torneo para cada individuo y su correspondiente vecindario, obteniendo dos individuos utilizados como semillas para el nuevo material genético. Este torneo utiliza un ranking en frentes y distancia de apiñamiento de ser necesario para elegir a los dos mejores.

**Recombinación:** se utiliza una adaptación del método clásico “*n -point crossover*” el cual se restringe los puntos donde puede cortar el operador a solo la ubicación en la que se encuentran los genes con valor de cero. Esta restricción se propone con el fin de preservar el orden de los vehículos en periodos de tiempo que presentan un buen rendimiento y fomentar el elitismo (ver **Figura 20**).



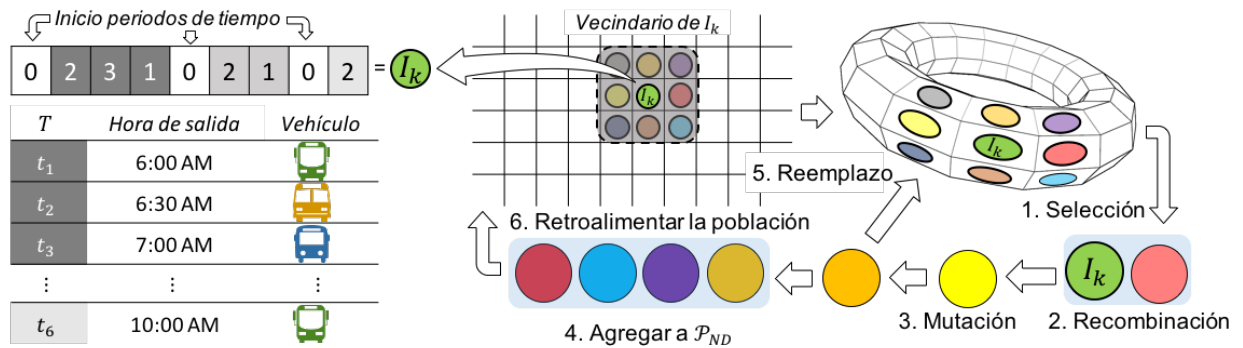
**Figura 20.** Método de recombinación implementado, modificación del tradicional “*one-point crossover*”.

**Mutación:** se define como método de mutación un mecanismo de intercambio, en el que se selecciona  $|\kappa|$  genes, determinado de manera aleatoria una cantidad de valores a  $k_i$  entre 1 y el tamaño del cromosoma como posiciones en las que intercambiará el vehículo asignado por otro (de diferente tipo) disponible (ver **Figura 21**).



**Figura 21.** Método de mutación implementado, con un  $k = 3$  (cantidad de genes que pueden ser modificados).

**Reemplazo:** para cada una de las celdas de la malla toroidal se realiza un torneo entre los padres, los hijos y el individuo, el que presente un mejor rendimiento ocupará este lugar en la población y se integrará con  $\mathcal{P}_{ND}$  en donde competirá con los individuos, que se encuentran almacenados previamente, por un lugar en el conjunto de individuos no dominados. Por último, cuando se evalué la totalidad de la población (individuo por individuo) se procede a realizar una selección aleatoria de  $|\mathcal{P}_{ND}|$  individuos que serán eliminados y reemplazados por los almacenados en  $\mathcal{P}_{ND}$  (ver **Figura 22**).



**Figura 22.** Ejemplo de la representación de una solución (cromosoma) para un caso con tres tipos de vehículos diferentes y un total de seis viajes en tres periodos de tiempo. Etapas de reproducción para un algoritmo celular asíncrono.

## Capítulo 5. Resultados experimentales

---

Para lograr encontrar una solución (o una “buena” aproximación) para un POMO, se han desarrollado un número considerable de métodos y algoritmos durante décadas, entre estos se encuentran los AE los cuáles producen un conjunto de soluciones no dominadas que se aproximan al frente de Pareto (conjunto de soluciones óptimo para un POMO) para problemas con dos o más objetivos. El aumento del interés por este tipo de métodos incrementó la necesidad de diseñar mecanismos que permitan comparar los resultados entregados por dos algoritmos y determinar cuál presentó un mejor rendimiento. Esto es complicado debido a que los AEMO no producen un número escalar como indicador de aptitud, el cual puede ser tratado con diversos métodos estadísticos. En contraparte, entregan un conjunto de soluciones no dominadas (puntos en el espacio objetivo) como resultado (Knowles y Corne, 2002), por lo tanto se han propuesto numerosas **métricas** de comparación de rendimiento como indicadores de calidad, permitiendo la comparación entre conjuntos obtenidos mediante la implementación de AEMO.

En el Capítulo 3 se introducen dos conceptos claves para el uso de este tipo de métricas: la proximidad al frente de Pareto y la diversidad del conjunto solución, varios autores proponen tres objetivos en los que deben enfocarse los métodos de solución para un POMO (Okabe et al., 2003; Zitzler et al., 2000):

- Minimizar la distancia entre el vector de soluciones no dominadas y el frente Pareto ( $\mathcal{P}_F$ ), en otras palabras, se busca estar lo más cerca posible de la solución óptima (**convergencia**).
- Obtener una buena (uniforme de preferencia) distribución de las soluciones del conjunto no dominado en el espacio objetivo (**diversidad**).
- Maximizar la cardinalidad del conjunto solución (**cantidad** de soluciones no dominadas)

Las métricas pueden ser unarias o binarias (m), en función de cuantos conjuntos solución pueden evaluar simultáneamente, evaluando alguno(s) de los tres criterios antes mencionados, (Riquelme et al., 2015) listan un total de 54 métricas diferentes, citadas entre 2005 y 2013.

La pregunta que surge es ¿Cuántas o cuáles métricas de rendimiento se deben utilizar? apoyado en el mismo estudio, se examinó las diez métricas más utilizadas con el fin de seleccionar un conjunto que evalué los tres criterios de rendimiento y permita comparar los resultados obtenidos de dos algoritmos diferentes.

En este capítulo se describe el diseño de los experimentos implementados, así como los indicadores de calidad (métricas de rendimiento) utilizados para evaluar el desempeño del algoritmo genético celular (AGC) propuesto. Por último, se presentan los resultados experimentales junto a una discusión en torno al análisis de los mismos.

## 5.1 Indicadores de calidad (métricas de rendimiento)

Antes de describir las métricas utilizadas para evaluar el rendimiento del algoritmo propuesto, es importante revisar los conceptos descritos en el Capítulo 3.

Zitzler et al. (2003) definen una métrica de rendimiento como:

Un indicador de calidad  $I$  de orden  $m$  es una función  $I: \mathbb{P}^m \rightarrow \mathbb{R}$ , la cual asigna para cada conjunto  $\mathbb{P} = [\mathcal{P}_1, \dots, \mathcal{P}_m]$  de  $m$  conjuntos de aproximación, un valor real  $I(\mathcal{P}_1, \dots, \mathcal{P}_m)$ . Es posible clasificar las métricas de rendimiento en tres categorías (Okabe et al., 2003; Riquelme et al., 2015):

- **Métricas de cardinalidad:** la cardinalidad de un conjunto solución  $\mathbb{P}$  se refiere a la cantidad de soluciones  $\mathcal{P}$  existentes.
- **Métricas de exactitud:** este tipo de métodos evalúan el nivel de convergencia de un  $\mathcal{P}$ , es decir, indica cuanta distancia separa  $\mathcal{P}$  de  $\mathcal{P}_F$  si este último es conocido, de lo contrario se utilizará un  $\mathcal{P}_R$  (regularmente calculado de forma experimental) como referencia (Hipervolumen, “Two set coverage”, indicador $-\epsilon$ ).
- **Métricas de diversidad:** este tipo de indicador evalúa la distribución y extensión de las soluciones en  $\mathcal{P}$ , aunque son dos términos que pueden sonar muy similares, la distribución representa la distancia relativa entre las soluciones no dominadas, mientras que la extensión se refiere al espacio que cubre una solución en  $\mathcal{P}$  (“Spacing”, “Spread”, Hipervolumen).

### 5.1.1 Espaciado y propagación

El indicador de calidad conocido como **espaciado** (Schott, 1995), más conocido como “spacing”, es una métrica evalúa la distribución de las soluciones no dominadas obtenidas mediante un AEMO en un

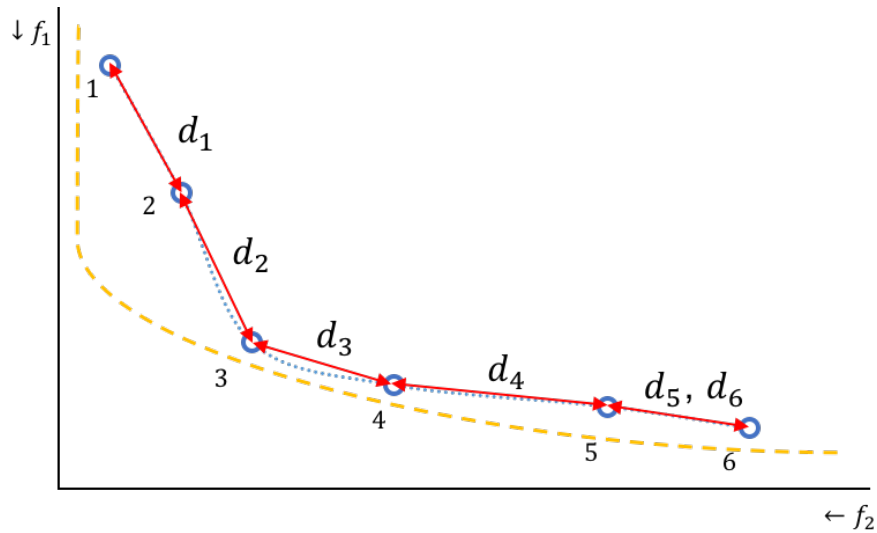


conjunto  $\mathcal{P}$ , calculando la varianza de la distancia relativa entre dos soluciones consecutivas. Es calculada en el espacio objetivo, como una métrica unaria, de la siguiente forma:

$$I_S = \sqrt{\frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} (d_i - \bar{d})^2} \quad (35)$$

$$d_i = \min_j \{|u_1^i - u_1^j| + |u_2^i - u_2^j|\} \quad (36)$$

$I_S$  calcula la desviación estándar de los valores de  $d_i$  ( $\bar{d}$  es el promedio de las distancias mostradas en la **Figura 23**), por lo tanto, un conjunto  $\mathcal{P}$  con menor “spacing” es considerado mejor, debido a que supone una distribución uniforme de las soluciones.



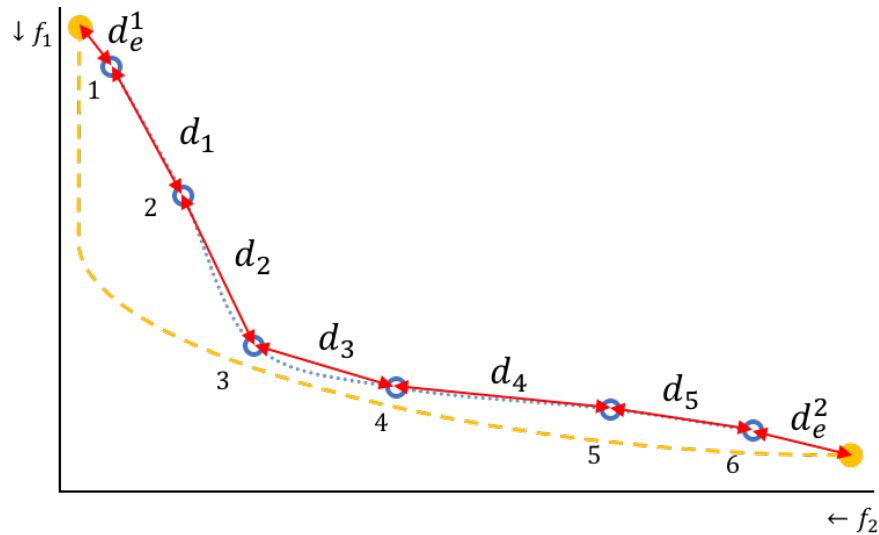
**Figura 23.** Distancias entre las soluciones de un  $\mathcal{P}$ .

La métrica de propagación propone la inclusión de las soluciones que se ubican en los extremos del frente Pareto para evaluar la extensión de las soluciones sobre  $\mathcal{P}_F$  (Deb, 2001; Deb et al., 2002).  $I_\Delta$  o “spread” emplea los mismos valores de  $d_i$  y  $\bar{d}$  utilizados por  $I_S$  además de la distancia entre los extremos del frente de Pareto y la solución en  $\mathcal{P}$  más cercana ( $d_k^e$ ) para las  $n$  funciones objetivo (ver **Figura 24**).

$$I_\Delta = \frac{\sum_{k=1}^n d_k^e + \sum_{i=1}^{|\mathcal{P}|} |d_i - \bar{d}|}{\sum_{k=1}^n d_k^e + |\mathcal{P}| \bar{d}} \quad (37)$$

El indicador toma el valor de cero  $I_\Delta = 0$  para una distribución ideal que incluye los extremos del frente de Pareto en el conjunto  $\mathcal{P}$ . Por lo tanto, cuando un algoritmo calcula un conjunto con un valor pequeño

de  $I_{\Delta}$  es posible encontrar una mejor diversidad de las soluciones no dominadas, dado que la extensión de las mismas a lo largo del frente es buena en comparación con otro  $\mathcal{P}$  que obtenga valores altos para  $I_{\Delta}$ .



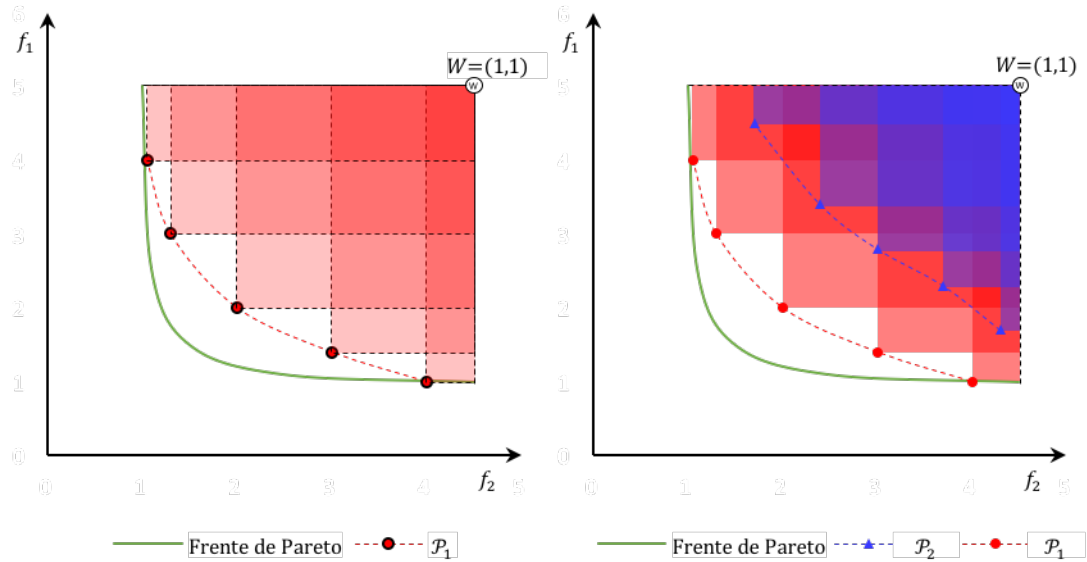
**Figura 24.** Distancias entre las soluciones de un  $\mathcal{P}$ , incluidas las distancias entre los extremos del frente de Pareto y las soluciones más cercanas.

### 5.1.2 Hipervolumen

Este indicador de calidad calcula la región  $n$  –dimensional encerrada entre el punto que representa una solución no dominada en el espacio objetivo y un punto de referencia  $W$  (Zitzler y Thiele, 1999). Si el conjunto aproximación está muy cerca (distancia) de  $\mathcal{P}_F$  aumenta el valor de este indicador. Por otra parte, si la diversidad (distribución y dispersión) de  $\mathcal{P}$  presenta un buen rendimiento, se incrementa el  $I_{HV}$ . Por lo tanto, un conjunto solución  $\mathcal{P}$  que obtenga el valor más alto de  $I_{HV}$  es considerado mejor, debido a que el valor más alto (valor óptimo) de esta métrica se obtiene al evaluar el frente de Pareto.

Para calcular el hipervolumen es necesario colocar las soluciones en el espacio objetivo y elegir un punto  $W$  que no sea rebasado por ninguna solución factible (p. Ej. construir un vector con los peores valores posibles de las funciones objetivo). Un hipercubo ( $HC_k$ ) es construido para cada solución utilizando como esquinas de referencia el punto de la solución y el punto de referencia (ver **Figura 25**). Por último, se realiza la unión de estos hipercubos y se procede a calcular el hipervolumen, matemáticamente:

$$I_{HV} = \text{volumen} \left( \bigcup_{k=1}^{|\mathcal{P}|} HC_k \right) \quad (38)$$



**Figura 25.** El hipervolumen encerrado por las soluciones que pertenecen al conjunto de soluciones no dominado  $\mathcal{P}_1$  y el punto  $W$  (izquierda). Comparación entre el hipervolumen construido por  $\mathcal{P}_1$  y  $\mathcal{P}_2$  (derecha).

Esta métrica es susceptible a la escala de las funciones objetivo, por lo que se recomienda utilizarla en espacios objetivo normalizados. Para el caso de dos funciones objetivo ( $n = 2$ ) se calcula el área de un rectángulo, construido al utilizar la línea recta entre el punto y  $W$  como diagonal principal (ver **Figura 25**).

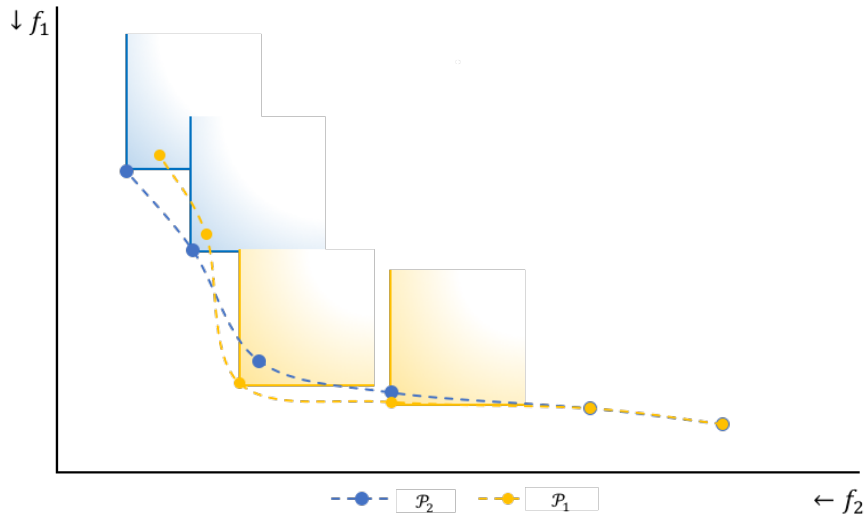
### 5.1.3 Cubrimiento entre dos conjuntos

Zitzler y Thiele (1998) proponen un indicador de calidad basado en el cubrimiento de conjuntos (“*Two set coverage*”), utilizado para evidenciar qué tanto los resultados de un algoritmo dominan a los resultados de otro, en otras palabras, puede estimar el porcentaje de soluciones, calculado por un algoritmo, que dominan a las soluciones de otro. Esta métrica provee una comparación relativa entre dos conjuntos de vectores de decisión  $X', X'' \in \Omega$  basado en su relación de dominancia de la siguiente manera:

$$I_{TSC}(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X': a' \succcurlyeq a''\}|}{|X''|} \quad (39)$$

El valor de  $I_{TSC}(X', X'') = 1$  representa que todas las soluciones en  $X''$  son dominadas o son iguales a  $X'$ . Por otra parte, el caso cuando ninguno de los puntos en  $X''$  son cubiertos por  $X'$ ,  $I_{TSC}(X', X'') = 0$ . Ambos  $I_{TSC}(X', X'')$  y  $I_{TSC}(X'', X')$  deben ser considerados para una evaluación de rendimiento, debido a que

$I_{TSC}(X', X'')$  no es necesariamente igual a  $1 - I_{TSC}(X'', X')$ . Este indicador binario es bastante aceptado para comparar dos conjuntos, aunque no refleja en qué medida uno es mejor que el otro (ver **Figura 26**).



**Figura 26.** Ejemplo de la interacción de dos conjuntos de soluciones  $\mathcal{P}_1$  y  $\mathcal{P}_2$  y sus posibles relaciones de dominancia.

#### 5.1.4 Indicador $-\epsilon$

Zitzler et al. (2003) introducen la métrica  $I_\epsilon$ , este indicador binario permite calcular un factor  $\epsilon$  que induce una relación de qué tanto un conjunto de soluciones es peor que otro. Entonces para un POMO con  $n$  objetivos en conflicto a minimizar  $\mathbb{F}(Z') \rightarrow \mathbb{R}^n$ , un vector objetivo  $Z' = z'_1, \dots, z'_n$  se dice que es dominado en factor  $\epsilon$  por  $Z'' = z''_1, \dots, z''_n$ , escrito de la siguiente forma:  $Z' \succ_\epsilon Z''$ , si y solo si para un  $\epsilon > 0$  se cumple:

$$\forall 1 \leq i \leq n: z'_i \leq \epsilon \cdot z''_i \quad (40)$$

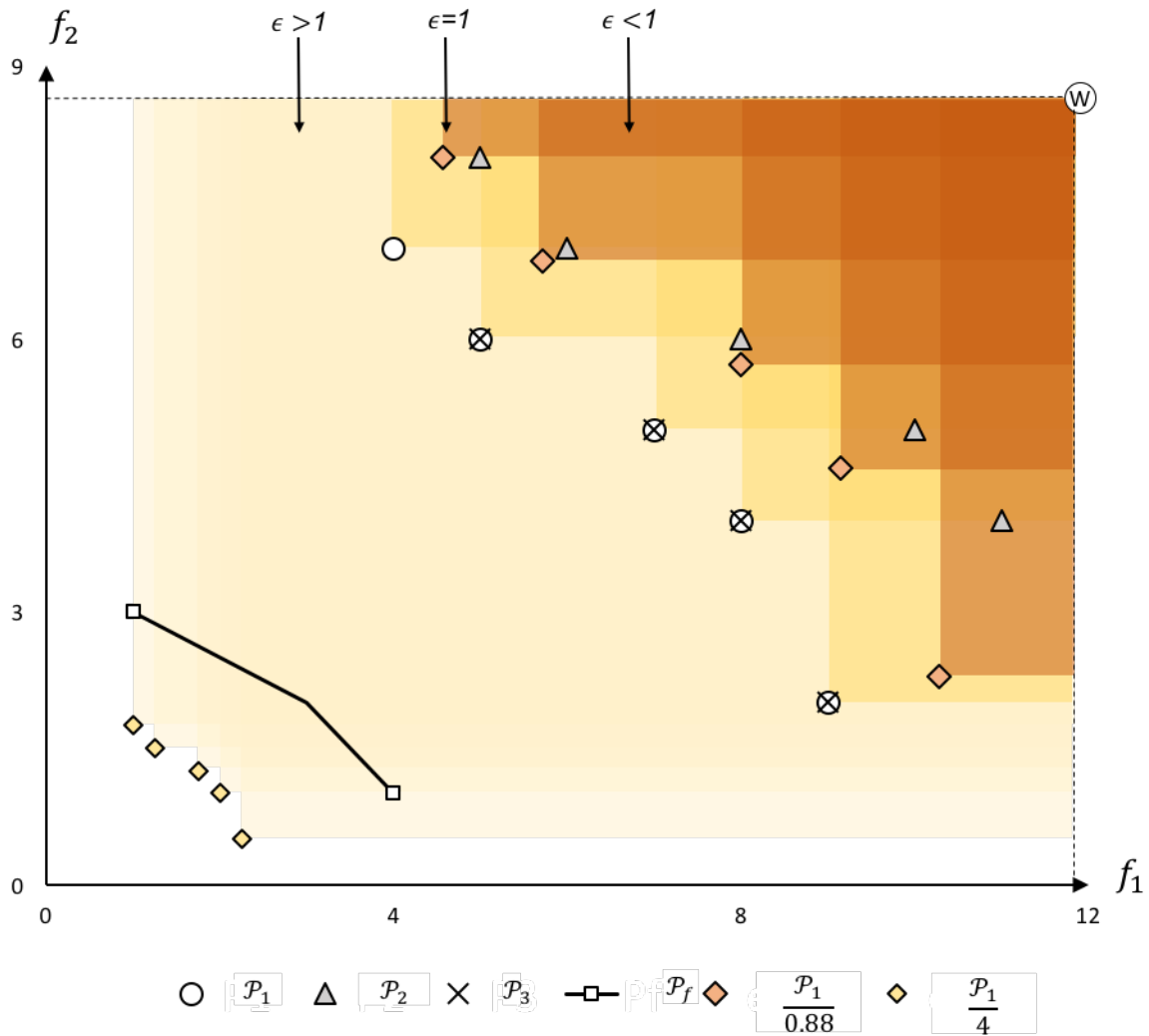
Es posible definir el indicador  $-\epsilon$  ( $I_\epsilon$ ) como:

$$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) = \inf_{\epsilon \in \mathbb{R}} \{ \forall Z'' \in \mathcal{P}_2 \exists Z' \in \mathcal{P}_1: Z' \succ_\epsilon Z'' \} \quad (41)$$

Entonces para dos conjuntos de aproximación  $\mathcal{P}_1, \mathcal{P}_2 \in \Omega$ ,  $I_\epsilon(\mathcal{P}_1, \mathcal{P}_2)$  es igual al mínimo valor de  $\epsilon$  tal que para cualquier solución en  $\mathcal{P}_2$  existe al menos una solución en  $\mathcal{P}_1$ , multiplicada por  $\epsilon$ , que no es peor al considerar todos los  $n$  objetivos. Este tipo de métrica es relativamente simple de calcular en tiempo  $O(n|\mathcal{P}_1||\mathcal{P}_2|)$  de la siguiente manera (Zitzler et al., 2003):

$$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) = \max_{Z'' \in \mathcal{P}_2} \min_{Z' \in \mathcal{P}_1} \max_{1 \leq i \leq n} \frac{z'_i}{z''_i} \quad (42)$$

El valor de  $I_\epsilon$  puede brindar información completa sobre la relación entre los resultados de dos algoritmos, debido a que este indicador puede calcular cuánto un conjunto de aproximación es mejor que otro (Riquelme et al., 2015). En la **Figura 20** se puede observar el comportamiento de un conjunto de soluciones no dominado al multiplicarlo por tres diferentes valores de  $\epsilon$ , y su interacción con otros conjuntos aproximados. En la **Tabla 2** se resume el comportamiento de los métodos binarios utilizados con respecto a las relaciones de dominancia abordadas en el Capítulo 2 (Zitzler et al., 2003).



**Figura 27.** Cada una de las áreas marcadas representa el espacio que es  $\epsilon$  – dominado al calcular la métrica así:  $I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) = 0.88$ ,  $I_\epsilon(\mathcal{P}_1, \mathcal{P}_3) = 1$ ,  $I_\epsilon(\mathcal{P}_1, \mathcal{P}_f) = 4$ . Multiplicando a la solución  $\mathcal{P}_1$  los valores obtenidos para  $I_\epsilon$  se obtienen las áreas que representan cada uno de los tres posibles casos de  $\epsilon$ .

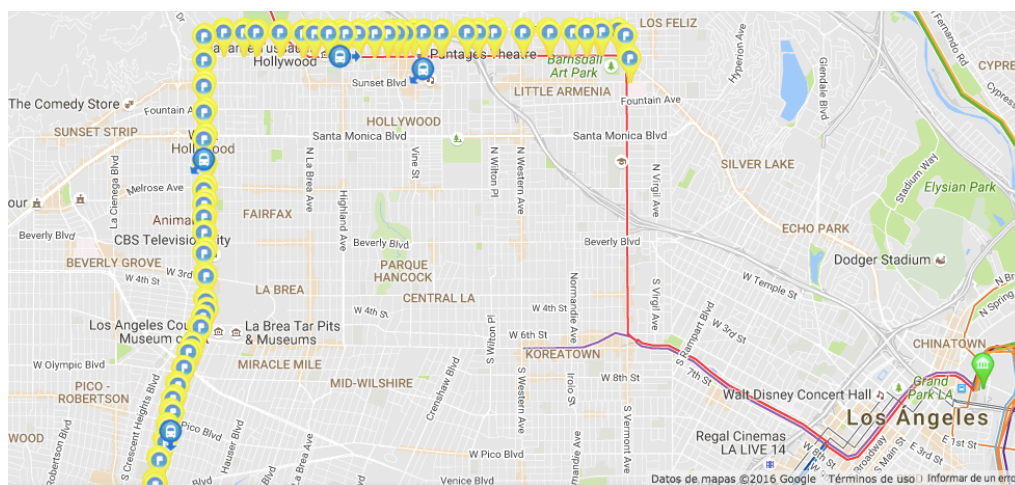
**Tabla 2.** Resumen general de las métricas binarias **Indicador- $\epsilon$**  y **Cubrimiento entre dos conjuntos** en términos de las relaciones de dominancia. Los espacios vacíos indican que no hay un método de comparación que permita definir que es completamente compatible con esta relación de dominancia.

		Métrica	
		$I_\epsilon$	$I_{TSC}$
Relación de dominancia	$\gg$	$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) < 1$	–
	$\succ$	–	$I_{TSC}(\mathcal{P}_1, \mathcal{P}_2) = 1$ $I_{TSC}(\mathcal{P}_2, \mathcal{P}_1) = 0$
	$\triangleright$	$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) \leq 1$ $I_\epsilon(\mathcal{P}_2, \mathcal{P}_1) > 1$	$I_{TSC}(\mathcal{P}_1, \mathcal{P}_2) = 1$ $I_{TSC}(\mathcal{P}_2, \mathcal{P}_1) < 1$
	$\succcurlyeq$	$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) \leq 1$	$I_{TSC}(\mathcal{P}_1, \mathcal{P}_2) = 1$
	$=$	$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) = 1$ $I_\epsilon(\mathcal{P}_2, \mathcal{P}_1) = 1$	$I_{TSC}(\mathcal{P}_1, \mathcal{P}_2) = 1$ $I_{TSC}(\mathcal{P}_2, \mathcal{P}_1) = 1$
	$\nexists$	$I_\epsilon(\mathcal{P}_1, \mathcal{P}_2) > 1$ $I_\epsilon(\mathcal{P}_2, \mathcal{P}_1) > 1$	$0 < I_{TSC}(\mathcal{P}_1, \mathcal{P}_2) < 1$ $0 < I_{TSC}(\mathcal{P}_2, \mathcal{P}_1) < 1$

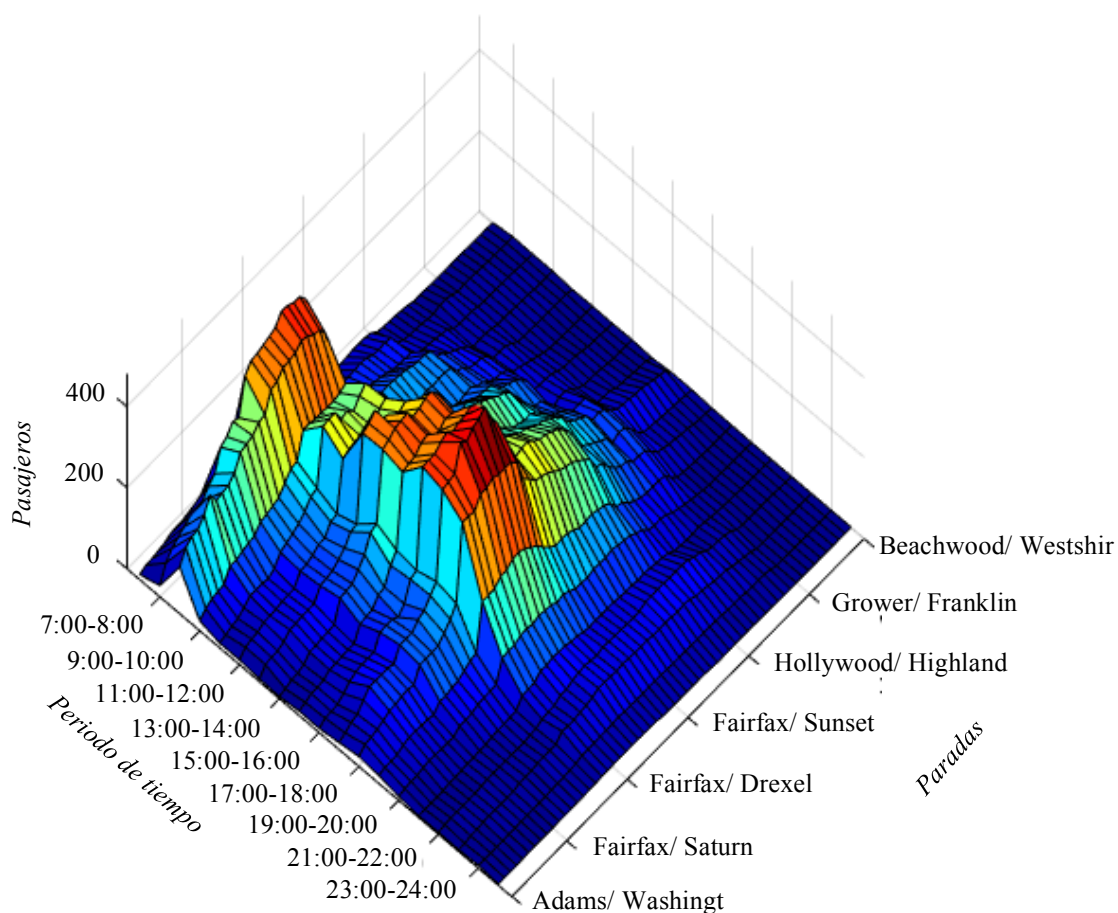
## 5.2 Diseño de experimentos

Para validar la implementación del algoritmo existen diferentes indicadores de calidad enfocados a evaluar la convergencia y la diversidad del conjunto de soluciones no dominadas resultante, sin embargo, es necesario encontrar la mejor combinación de los parámetros de ejecución que entregan los mejores resultados. Se utiliza el  $I_{HV}$  para la evaluación del rendimiento y selección de parámetros, debido a que en un solo valor entrega información relevante en cuanto a cercanía al conjunto óptimo de Pareto y la distribución de las soluciones a lo largo del frente aproximado encontrado. Luego se utilizan las métricas binarias para hacer una comparación entre los resultados de los dos algoritmos. Para la implementación se supone que la topología de la ruta está trazada (planificación estratégica), las paradas tienen una ubicación geográfica

Para realizar los experimentos se utilizó una carga de prueba encontrada en la literatura que representa el comportamiento de la ruta 217 en sentido sur-norte en la ciudad de Los Ángeles, California EEUU (ver **Figura 28**), los datos tienen la estructura de una matriz con el valor de la demanda promedio de pasajeros por parada en intervalos de tiempo de una hora (ver **Figura 29**). Este conjunto de datos se selecciona debido a la posibilidad de hacer una comparación con la asignación de vehículos que se observó en el momento de la recolección de los datos de la demanda de pasajeros y una asignación de vehículos iguales (Ceder, 2016).



**Figura 28.** Mapa digital de la ciudad de Los Ángeles, California EEUU con las paradas de la ruta 217 resaltadas en sentido sur-norte y norte-sur (Esidian, 2017).



**Figura 29.** Carga de prueba utilizada para la ejecución de los algoritmos, representa el promedio de la demanda de pasajeros por cada una de las 59 paradas en intervalos de tiempo de una hora (19 intervalos en total) para la ruta 217 en LA, carga máxima de pasajeros (481) en la parada Fairfax/ Rosewood durante las 17:00 a las 18:00 horas (hora pico).

Los experimentos fueron realizados con un Intel Core i5 @ 1.6Ghz, 4GB RAM 1.6GHz DDR3 con sistema operativo 64 bit macOS Sierra versión 10.12.1.

El desempeño de la planificación táctica del transporte público depende de las características propias del problema, los métodos de solución y los parámetros del algoritmo seleccionado. Por lo tanto, es necesario realizar diferentes combinaciones y análisis estadísticos que permitan interpretar los resultados, validando las hipótesis. Como se describió en el Capítulo 4 los algoritmos evolutivos constan de secciones que operan con diferentes probabilidades de elección (recombinación, mutación o selección) además de elementos de tamaño variable (población, torneo de selección o conjunto de individuos no dominados para retroalimentación) que afectan directamente en el comportamiento del algoritmo.

Los primeros parámetros para analizar son la probabilidad de recombinación  $P_r$  y mutación  $P_m$ , por lo cual dos posibles padres o semillas pueden o no crear nuevo material genético en función de  $P_r$ , de la misma manera esta descendencia puede presentar o no mutaciones estocásticas en función de  $P_m$ .

Para el algoritmo genético celular (AGC) propuesto se realizan 30 ejecuciones independientes con 10000 evaluaciones de calidad de los individuos para las combinaciones posibles entre probabilidades de recombinación y mutación, en total se obtienen resultados significativos para un intervalo de  $P_r$  y  $P_m$  diferentes, desde 0.15 hasta 1 con incrementos de 0.05, para un total de 324 combinaciones de parámetros diferentes (ver **Tabla 3**). Se ejecuta entonces un total de 9720 veces cada algoritmo y se evalúa el rendimiento de cada frente aproximado resultante mediante  $I_{HV}$ .

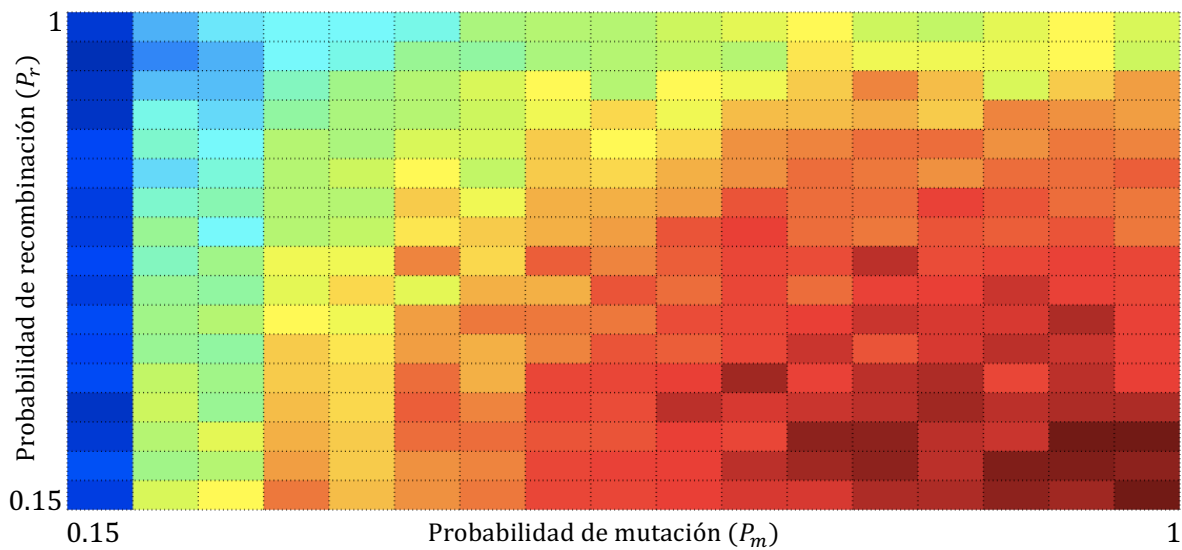
**Tabla 3.** Parámetros de configuración iniciales para el algoritmo genético celular.

<i>Condición de parada</i>	10000 evaluaciones de calidad del individuo
<i>Tamaño de la población</i>	100 individuos (10x10)
<i>Tamaño del vecindario</i>	8 vecinos circundantes
<i>Selección</i>	Torneo binario
<i>Recombinación</i>	Cruzamiento (posibles puntos de corte únicamente en al inicio de cada lapso)
$P_r$	[0.15,1]
<i>Mutación</i>	Intercambio aleatorio
$P_m$	[0.15,1]
<i>Reemplazo</i>	Reemplazo por ranking de dominancia y/o distancia de apiñamiento
<i>Estimador de densidad</i>	Distancia de apiñamiento
<i>Retroalimentación</i>	20 individuos



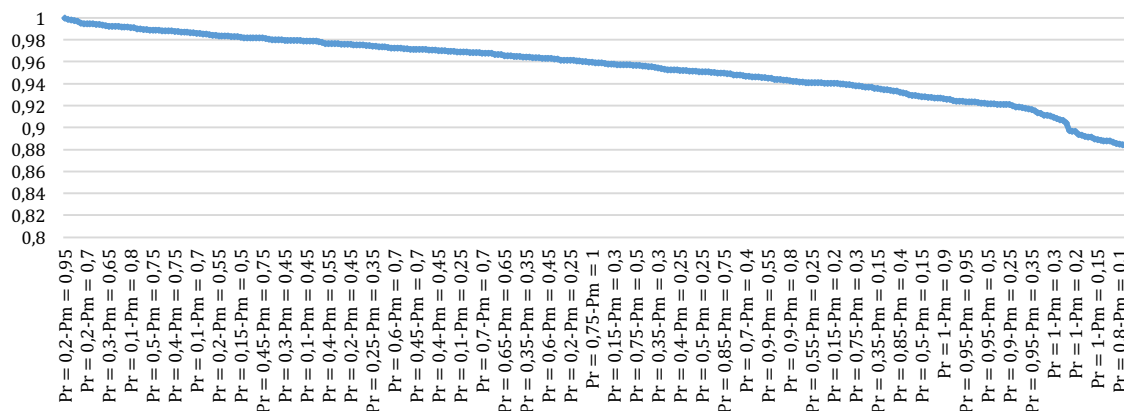
### 5.3 Resultados experimentales

Tomando como referencia los resultados presentados en el mapa de calor en la **Figura 30**, es posible observar el área donde los valores para la probabilidad de los operadores de recombinación y mutación ( $P_r$  y  $P_m$ ) presentan un mejor rendimiento en términos de hipervolumen (valor promedio de las 30 ejecuciones), se realiza una verificación de la normalidad del conjunto de datos con el test Shapiro–Wilk, obteniendo que los 318 de los 324 experimentos presentan una distribución normal ( $p > 0,05$ ), por lo tanto se aplica una prueba de Friedman (prueba no paramétrica), utilizando el software estadístico MiniTab versión 17.1.0.



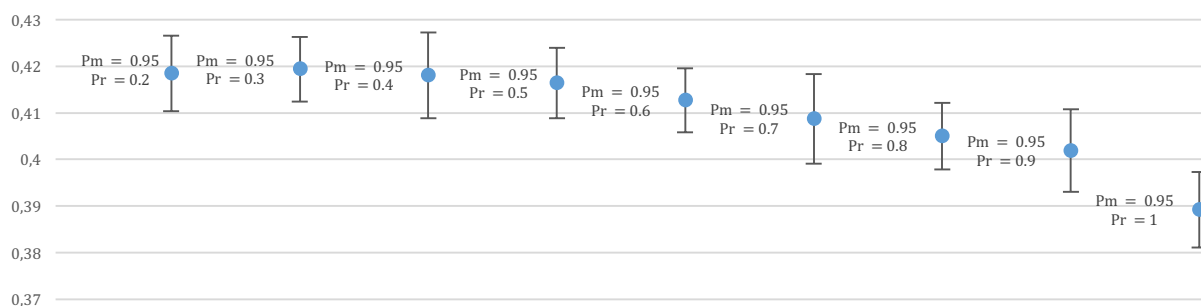
**Figura 30.** Mapa de calor para el rendimiento promedio ( $I_{HV}$ ) de los resultados del algoritmo genético celular basado en MOCeCell en su versión asíncrona.

Se obtiene como resultado un valor de  $p = 2.20e - 16$ , el cual es menor que 0.05 por lo que se puede rechazar la hipótesis nula constatando, indicando una diferencia estadísticamente significativa. Para poder seleccionar un conjunto de parámetros se decide validar el efecto de la recombinación utilizando la mutación con mejor rendimiento promedio ( $P_m = 0.95$ ) (ver **Figura 31**).



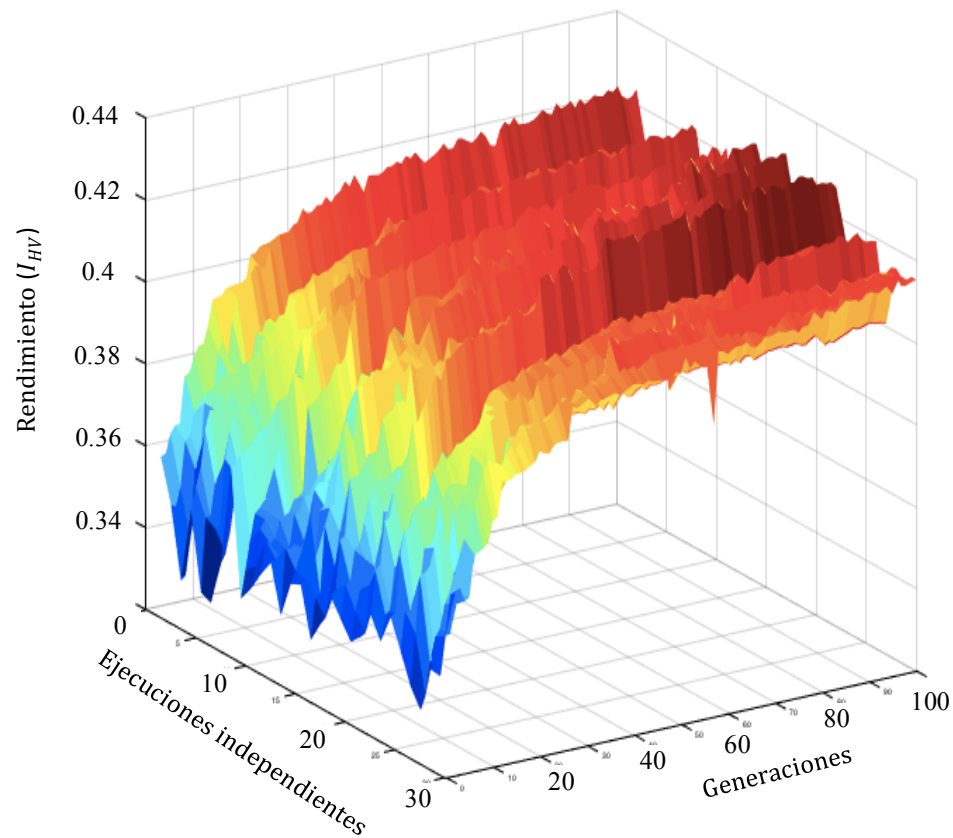
**Figura 31.** Degradación del promedio de  $I_{HV}$  para las 342 combinaciones de parámetros para el algoritmo genético celular ( $P_r$  y  $P_m$ ).

Después de verificar la combinación de parámetros adecuada para el AGC se seleccionan una  $P_m = 0.95$  y  $P_r = 0.3$ , dado que tiene un rendimiento promedio del 98.3% en base a la mejor solución encontrada en el primer experimento (ver **Figura 30**) y en el segundo (ver **Figura 32**) generó el valor de hipervolumen máximo encontrado además del rendimiento promedio más alto.



**Figura 32.** Impacto de la variación de la probabilidad de recombinación con un mismo valor de  $P_m$ .

En la **Figura 33** se muestra el rendimiento en términos de  $I_{HV}$  del AGC con los parámetros elegidos para 30 ejecuciones independientes, se busca evidenciar un comportamiento que incremente el valor de  $I_{HV}$  en cada iteración, ya que está demostrado que el valor máximo posible es el correspondiente al frente de Pareto (Zitzler et al., 2003).



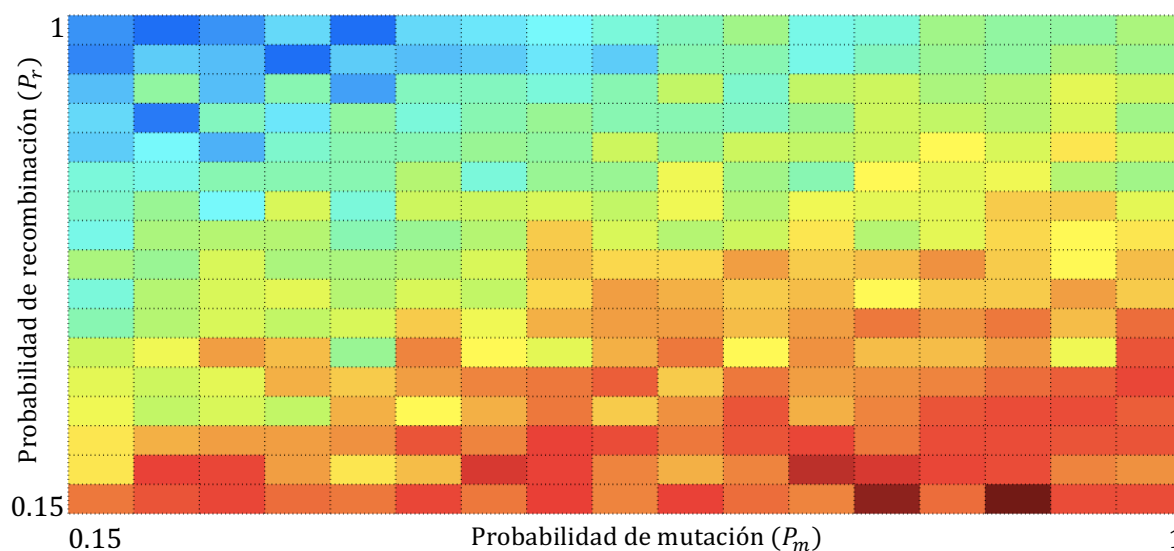
**Figura 33.** Rendimiento del hipervolumen en 30 ejecuciones independientes del algoritmo genético celular para 100 generaciones, es decir, evaluar la malla toroidal completa, individuo por individuo.

Para el algoritmo NSGA-II se realizó un procedimiento similar con una población de 100 individuos y 1000 iteraciones del algoritmo como criterio de parada, posteriormente se modificó la cantidad de individuos y el número de iteraciones utilizando la mejor combinación entre mutación y cruzamiento (ver **Tabla 4**), sin embargo para el algoritmos celular no presentó mejoras, a diferencia de NSGA-II donde con una población más grande y/o mayor número de iteraciones presentaba cierta mejora en términos de diversidad.

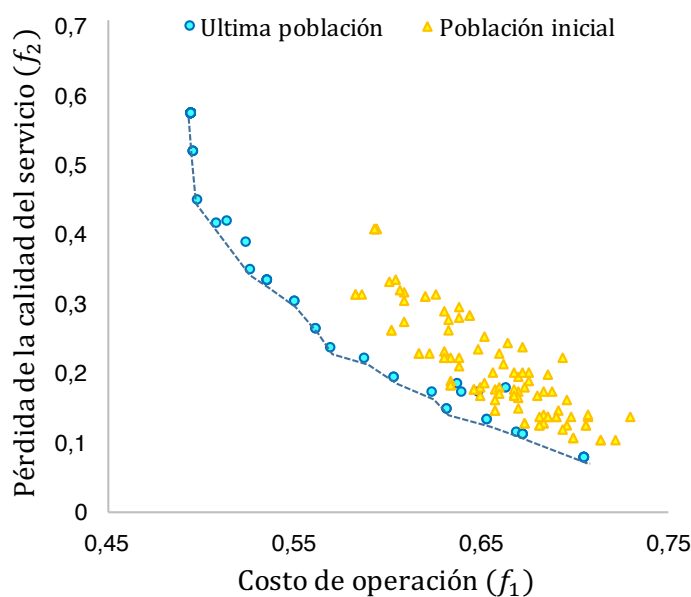
**Tabla 4.** Parámetros de configuración iniciales para el algoritmo genético NSGA-II.

<i>Condición de parada</i>	1000 evaluaciones de calidad
<i>Tamaño de la población</i>	100 individuos
<i>Selección</i>	Torneo binario
<i>Recombinación</i>	Cruzamiento (posibles puntos de corte únicamente en el inicio de cada lapso)
$P_r$	[0.15,1]
<i>Mutación</i>	Intercambio aleatorio
$P_m$	[0.15,1]
<i>Reemplazo</i>	Reemplazo por ranking de dominancia y/o distancia de apiñamiento
<i>Estimador de densidad</i>	Distancia de apiñamiento

Después de definir los parámetros para cada algoritmo se estudia su comportamiento en el espacio de las funciones objetivo, se analiza los frentes aproximados resultantes, en términos de  $I_{HV}$  (convergencia y dispersión) (ver **Figuras 30 y 34**). En la **Figura 35** se presenta la diferencia entre la población inicial y la última para una ejecución del AGC. Con la finalidad de evidenciar de manera gráfica cómo el algoritmo evoluciona la población conforme pasan las generaciones, realizando cambios en los individuos que representan una mejora en la calidad de cada una de las soluciones factibles para cada función objetivo.

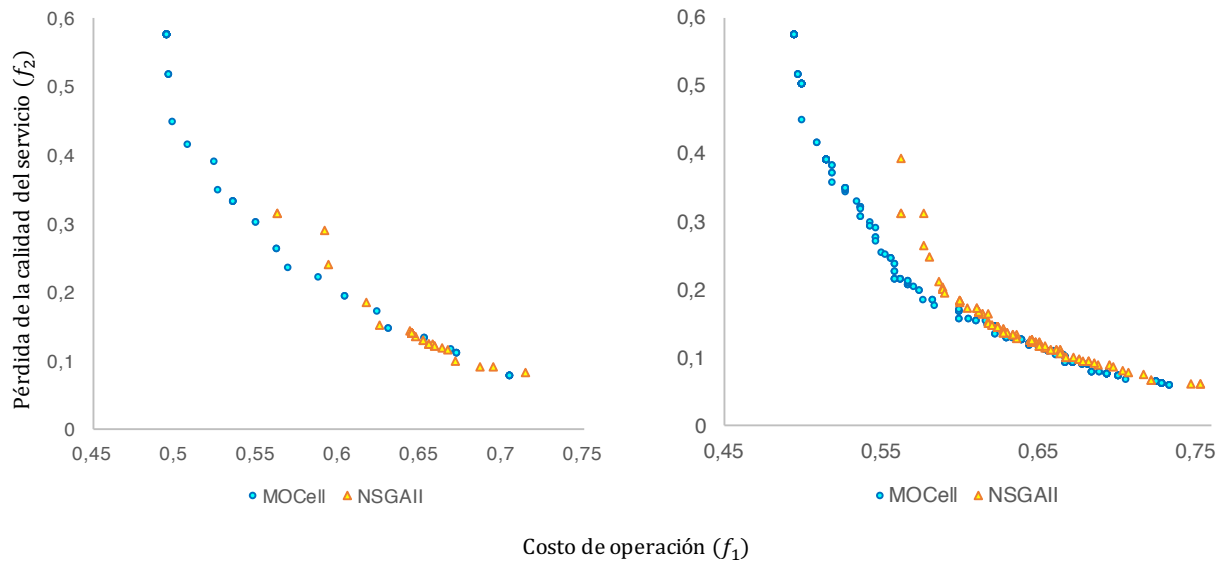


**Figura 34.** Mapa de calor para el rendimiento promedio ( $I_{HV}$ ) de los resultados del algoritmo genético NSGA-II.



**Figura 35.** Conjunto de soluciones obtenido con el algoritmo genético celular, correspondiente a un  $I_{HV} = 0.4398$  en comparación a la población inicial para una ejecución con  $Pr = 0,95$  y  $Pm = 0,3$ .

En la **Figura 36** se presenta el conjunto de soluciones no dominadas resultantes con mejor rendimiento para los dos algoritmos, además de una comparación al unir los conjuntos de soluciones de 30 corridas diferentes y encontrar la aproximación al frente de Pareto resultante.



**Figura 36.** Mejor aproximación al frente de Pareto obtenida para MOCcell y NSGAII, correspondiente a un  $I_{HV} = 0.4398$  y  $0.3866$  respectivamente (izquierda). Mejor aproximación al frente de Pareto obtenida, mediante la unión de las 30 soluciones encontradas en cada ejecución de los algoritmos (derecha).

Por último, se realiza una comparación entre los resultados de los dos algoritmos implementados utilizando tanto las métricas unarias como las binarias (ver **Tabla 5-7**).

**Tabla 5.** Métricas para POMO como indicadores de calidad calculadas para AGC y NSGA-II.

Indicador de calidad		AGC	NSGA-II
Puntos no dominados	Max	30	39
	Media	19.066	30.233
	$\sigma$	1.4840	3.4209
Spread ( $I_{\Delta}$ )	Min	0.6186	0.7415
	Media	0.8112	1.0547
	$\sigma$	0.1097	0.1569
Hiper volumen ( $I_{HV}$ )	Max	0.4398	0.3866
	Media	0.4193	0.3663
	$\sigma$	0.0069	0.0096

**Tabla 6.** Comparación de resultados aplicando la métrica  $I_{TSC}(X', X'')$  donde  $X'$  y  $X''$  representan el algoritmo en la fila y en la columna respectivamente.

		NSGA-II	AGC
NSGA-II	media	—	0.0721
	max	—	0.2857
AGC	media	0.6613	—
	max	1	—

**Tabla 7.** Comparación de resultados aplicando la métrica  $I_e(A, B)$  donde  $A$  y  $B$  representan el algoritmo en la fila y en la columna respectivamente.

		<i>NSGA-II</i>	<i>AGC</i>
<i>NSGA-II</i>	media	–	1.3035
	min	–	1.0653
	max	–	2.1065
<i>AGC</i>	media	1.0167	–
	min	0.9760	–
	max	1.4250	–

## 5.4 Análisis de resultados y discusión

Se puede observar en las **Figura 30** y **34** un patrón de comportamiento característico en cada algoritmo, sin embargo, es más notorio para el AGC (ver **Figura 30**), en ambos casos predomina una alta probabilidad para el operador genético de mutación y por el contrario la probabilidad de recombinación que generaba mejores resultados, tenía valores medios o bajos (ver **Figura 31**). Estos resultados resaltan las características del problema y la representación utilizada (cromosoma), dado que es más sensible a cambios drásticos producidos por el método de recombinación y presenta mejoras al intercambiar uno o varios vehículos, pero dentro del mismo intervalo de tiempo. El AGC implementado presenta un comportamiento similar en cada ejecución, corroborando que la elección de los parámetros para los operadores genéticos utilizados afecta positivamente las manipulaciones estocásticas del cromosoma, incrementando el valor del hipervolumen promedio del conjunto resultante en cada generación, lo que implica mejores aproximaciones al frente Pareto tanto en convergencia como en diversidad de las soluciones no dominadas, conforme avanzan las generaciones en el AGC.

En la **Figura 35** se puede apreciar como el AGC tiene la capacidad de mover gran parte de la población inicial en dirección al frente de Pareto, aunque la cantidad de individuos diferentes se reduce (lo que no necesariamente significa que dos individuos tengan el mismo cromosoma).

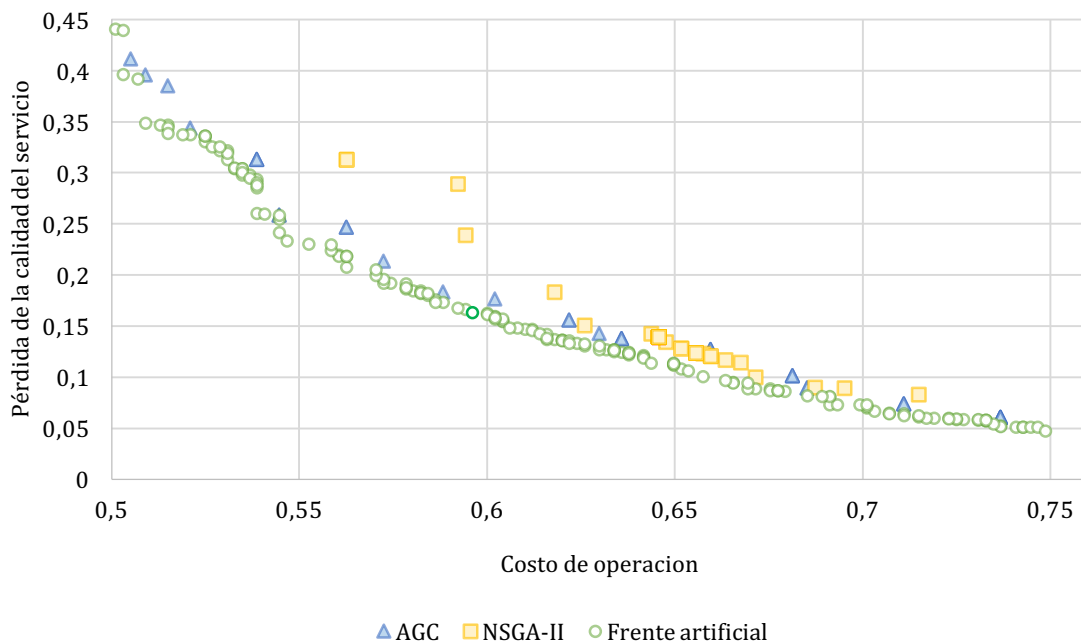
A diferencia del AGC el algoritmo NSGA-II puede encontrar un número mayor de puntos diferentes (30% más individuos en promedio). Sin embargo, la dispersión o extensión de las soluciones a lo largo del frente aproximado entregado por AGC presenta un rendimiento 31% mejor en el caso promedio. Para el caso del hipervolumen, el algoritmo genético celular puede encontrar un frente aproximado 14% más grande en promedio que los entregados por NSGA-II (ver **Tabla 5**), lo cual supone que AGC tiene un mejor rendimiento en términos de convergencia al frente de Pareto, ya que en términos de diversidad pueden ser similares.

Para validar la información anterior, en la **Tabla 6** se presentan las comparaciones realizadas en términos de cubrimiento de conjuntos, donde se indica que AGC cubre o domina en promedio el 66% de las soluciones de un conjunto calculado con NSGA-II (100% para el caso máximo), lo que supone una ventaja en términos de convergencia, aunque no indica cuánto o qué tan cerca esta AGC del frente de Pareto (Zitzler et al., 2003).

Para discutir lo anterior, en la **Tabla 8** se presentan los valores para el indicador-  $\epsilon$  demostrando que ambos algoritmos entregan resultados que tienen una distancia al frente de Pareto muy similar, basados en el concepto presentado en la

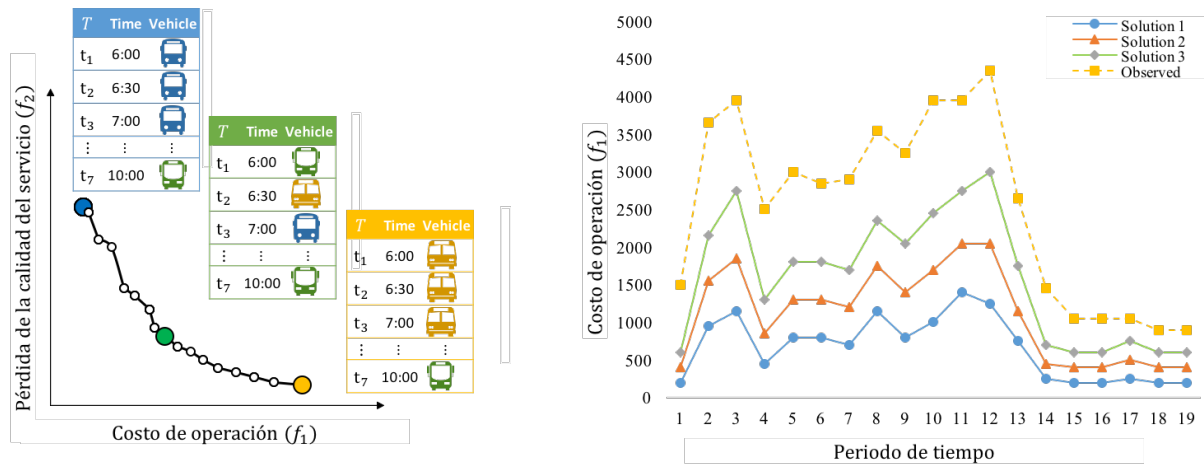
**Tabla 2** al evaluar 30 resultados diferentes para cada algoritmo, solo el 9% de los resultados de AGC son mejor o domina completamente algún resultado de NSGA-II y los frentes aproximados restantes no son comparables, esto se puede evidenciar en la **Tabla 8**, considerando que en promedio ambos algoritmos tienen un  $I_\epsilon > 1$ .

Para visualizar lo anterior de una manera gráfica en la **Figura 37** se presenta un frente de Pareto artificial, construido a partir de la unión de los 300 frentes aproximados encontrados en el experimento presentado en la **Figura 32** y el posterior cálculo del conjunto no dominado, comparado con dos frentes aproximados calculados por los algoritmos.



**Figura 37.** Comparación entre un frente de Pareto artificial y dos conjuntos de soluciones no dominadas.

En términos del rendimiento del algoritmo como solución al problema MVTTTP propuesto, se compara con los vehículos observados por un verificador. En la **Figura 38** se puede observar que utilizando los extremos y un punto intermedio de un frente aproximado calculado por AGC (valores más altos del indicador de densidad), ninguna de las tres asignaciones resulta ser más costosa, aunque la calidad del servicio se vea disminuida entre más económica sea la asignación. Es posible suponer entonces que mediante la implementación del AE se pueden lograr horarios o calendarios que brinden un servicio de calidad a diferentes niveles sin elevar los costos. El tomador de decisiones puede seleccionar alguna de las soluciones no dominadas y evaluar su viabilidad únicamente en términos de la calidad de servicio, teniendo en cuenta que el máximo costo puede no superar el umbral que tenían previamente presupuestado.



**Figura 38.** Ejemplo de tres calendarios diferentes (fenotipo) al interpretar tres cromosomas ubicados en el espacio objetivo (izquierda). Comparación de los tres horarios encontrados versus el costo de operación de una asignación vehicular observada en la ruta 217 en Los Ángeles, California EEUU.



## Capítulo 6. Conclusiones y trabajo futuro

---

En resumen, el problema MVTP se formuló con un enfoque de optimización multiobjetivo, estableciendo dos objetivos con intereses conflictivos como lo son reducir los costos de operación y minimizar la pérdida de la calidad de servicio. Se definieron restricciones basadas en los lineamientos de los ITS como la frecuencia mínima de operación y el factor de carga para cada lapso.

Se diseñó e implementó un algoritmo genético celular basado en la versión asíncrona de MOCeL que utiliza un método analítico para el diseño del cromosoma de los individuos. Se utilizó una carga de pasajeros con 59 paradas de una ruta en Los Ángeles, California, EEUU y los promedios del flujo de pasajeros (perfil de carga) durante 19 intervalos de una hora. Se ejecutaron más de 500 corridas independientes del algoritmo con parámetros diferentes en busca de la combinación adecuada de los mismos, de igual manera con la intención de comparar los resultados con un algoritmo genético multiobjetivo clásico se implementó NSGA-II, realizando un método de sintonización similar para la combinación de parámetros.

Es posible resaltar el buen desempeño del algoritmo genético de tipo celular para las condiciones y características del problema propuesto. Predominando siempre un rendimiento superior en convergencia y diversidad sobre NSGA-II, ambos operando con los parámetros que mejor rendimiento proporcionaban para cada ejecución. Lo que permite afirmar que los algoritmos genéticos de tipo celular pueden ser una herramienta útil para planificación táctica en sistemas de transporte público.

El análisis experimental demostró la capacidad del algoritmo propuesto para encontrar diferentes calendarios con asignación de vehículos diferentes para cubrir los viajes de una ruta específica. Encontrando las mejores compensaciones entre calidad de servicio para los pasajeros y costo de operación para los proveedores. Permitiendo el uso de diferentes restricciones y limitantes que pueden ser implementadas por los transportadores, ya que es un método flexible con tiempos aceptables en función del horizonte temporal de las decisiones que implican la planificación táctica del transporte público (mediano/corto plazo) con lo cual se cumple el objetivo propuesto para esta tesis.

Es posible diseñar e implementar otro tipo de estrategias u operadores genéticos que exploren y exploten el espacio de soluciones de mejor manera a partir de la experiencia obtenida del presente trabajo. Así como hacer modificaciones directamente a los calendarios que puedan incrementar la calidad de servicio sin dañar el costo de operación (p.ej. dividir un intervalo a cargo de un vehículo grande en dos asignados a minibuses eléctricos). De modo que se modifique el estilo *a posteriori* del algoritmo propuesto por un

estilo interactivo que pueda brindar soluciones más eficientes, calendarios más efectivos y métodos de optimización más rápidos.

Partiendo del trabajo realizado es posible modificar el problema planteado, modificando el objetivo relacionado con el costo de operación, de tal manera que cobre relevancia el impacto ambiental generado por el uso de energías no renovables como gasolina o gasóleo (diésel) para los sistemas de transporte, esto debido a que son uno de los principales generadores de gases de efecto invernadero (GEI), p.ej. 27% del total de GEI, ocupando el segundo lugar después de la generación de energía eléctrica (29%) para el caso de EEUU (Environmental Protection Agency, 2017). Por lo tanto, mediante decisiones logísticas y operativas se puede reducir los índices de contaminación partiendo desde una planeación táctica pensada para tal finalidad.

Uno de los posibles trabajos futuros basados en la presente tesis, está encaminado en la inclusión de la planificación operativa (calendarización de vehículos y del personal como conductores) dentro de un mismo sistema, realizando así las cinco etapas completas para la planificación de transporte público descritas en el documento, suponiendo que la etapa de planificación estratégica está resuelta (trazado y todas las características físicas de la red de rutas). Esta integración elevaría significativamente el costo computacional y el reto de encontrar soluciones que satisfagan los objetivos en conflicto con una compensación aceptable.

Como una línea de trabajo futuro adicional se propone la inclusión de incertidumbre como el uso de cargas dinámicas de pasajeros, dado que la integración de librerías **jMetal** con plataformas de datos en tiempo real como "**Spark**" (Cordero et al., 2016) facilitan este tipo de implementaciones. Es posible ampliar el vector objetivo y buscar otro tipo de compensaciones, dependiendo de las necesidades o información adicional del tomador de decisiones, el cual puede implementar otra heurística como búsqueda local para modificar la estructura del calendario y aumentar su rendimiento.

## Literatura citada

- Alam, M., Ferreira, J., & Fonseca, J. (2016). *Intelligent Transportation Systems*. (M. Alam, J. Ferreira, & J. Fonseca, Eds.) (Vol. 52). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-28183-4>
- Alba, E., & Dorronsoro, B. (2008). *Cellular Genetic Algorithms. Operations Research/ Computer Science Interfaces Series* (Vol. 42). Boston, MA: Springer US. <https://doi.org/10.1007/978-0-387-77610-1>
- Albino, V., Berardi, U., & Dangelico, R. M. (2015). Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, 22(1), 3–21. <https://doi.org/10.1080/10630732.2014.942092>
- Baaj, M. H., & Mahmassani, H. S. (1995). Hybrid route generation heuristic algorithm for the design of transit networks. *Transportation Research Part C: Emerging Technologies*, 3(1), 31–50. [https://doi.org/10.1016/0968-090X\(94\)00011-S](https://doi.org/10.1016/0968-090X(94)00011-S)
- Bianco, L., Mingozzi, A., & Ricciardelli, S. (1994). A set partitioning approach to the multiple depot vehicle scheduling problem. *Optimization Methods and Software*, 3(1–3), 163–194. <https://doi.org/10.1080/10556789408805563>
- Bielli, M., Caramia, M., & Carotenuto, P. (2002). Genetic algorithms in bus network optimization. *Transportation Research Part C: Emerging Technologies*, 10(1), 19–34. [https://doi.org/10.1016/S0968-090X\(00\)00048-6](https://doi.org/10.1016/S0968-090X(00)00048-6)
- Ceder, A. (1984). Bus frequency determination using passenger count data. *Transportation Research Part A: General*, 18(5–6), 439–453. [https://doi.org/10.1016/0191-2607\(84\)90019-0](https://doi.org/10.1016/0191-2607(84)90019-0)
- Ceder, A. (2007). *Public Transit Planning and Operation*. (Taylor & Francis, Ed.), *Transport* (1st ed.). Burlington: Elsevier. <https://doi.org/10.3846/16484142.2015.1114968>
- Ceder, A. (2016). *Public transit planning and operation : modeling, practice and behavior*. (Taylor & Francis, Ed.) (2nd ed.). Boca Raton, FL: CRC Press.
- Chakroborty, P. (2003). Genetic algorithms for optimal urban transit network design. *Computer-Aided Civil and Infrastructure Engineering*, 18, 184–200. <https://doi.org/10.1111/1467-8667.00309>
- Chakroborty, P. (2004). Brics Optimal Routing and Scheduling in Transportation : Using Genetic Algorithm to Solve Difficult Optimization Problems. *Directions-Indian Institute of Tech- Nology Kanpur*, [S.l.].
- Chamberlain, M., & Wren, A. (1992). Developments and Recent Experience with the BUSMAN and BUSMAN II Systems. In M. Desrochers & J.-M. Rousseau (Eds.), *Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop on Computer-Aided Scheduling of Public Transport held in Montréal, Canada, August 19--23, 1990* (pp. 1–15). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-85968-7\\_1](https://doi.org/10.1007/978-3-642-85968-7_1)
- Charnes, A., & Cooper, W. W. (1961). Management Models and Industrial Applications of Linear Programming. *Management Science*, 4(1), 63–64. <https://doi.org/10.1002/nav.3800090109>
- Chew, J. S. C., & Lee, L. S. (2012). A Genetical Algorithm for Urban Transit Routing Problem. *International*

- Journal of Modern Physics: Conference Series*, 9, 411–421.  
<https://doi.org/10.1142/S2010194512005491>
- Chvatal, V. (1979). A Greedy Heuristic for the Set-Covering Problem. *Mathematics of Operations Research*, 4(3), 233–235. <https://doi.org/10.1287/moor.4.3.233>
- Coello, C. a C., Lamont, G. B., & Veldhuizen, D. a Van. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Design. Boston, MA: Springer US. <https://doi.org/10.1007/978-0-387-36797-2>
- Cordero, J. A., Nebro, A. J., Barba-González, C., Durillo, J. J., García-Nieto, J., Navas-Delgado, I., & Aldana-Montes, J. F. (2016). Dynamic Multi-Objective Optimization with jMetal and Spark: A Case Study. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Vol. 10122 LNCS, pp. 106–117). [https://doi.org/10.1007/978-3-319-51469-7\\_9](https://doi.org/10.1007/978-3-319-51469-7_9)
- Costa, A. L., Cunha, M. C., Coelho, P. A. L. F., & Einstein, H. H. (2016). Application of the Simulated Annealing Algorithm for Transport Infrastructure Planning. In X.-S. Yang, G. Bekda\cs, & M. S. Nigdeli (Eds.), *Metaheuristics and Optimization in Civil Engineering* (pp. 235–253). Cham: Springer International Publishing. [https://doi.org/10.1007/978-3-319-26245-1\\_11](https://doi.org/10.1007/978-3-319-26245-1_11)
- Deb, K. (2001). Multi-objective optimization using evolutionary algorithms, 1, 497.  
<https://doi.org/10.1109/TEVC.2002.804322>
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), 182–197.  
<https://doi.org/10.1109/4235.996017>
- Deb, K., Sindhya, K., & Hakanen, J. (2016). Multi-Objective Optimization. In *Decision Sciences* (Vol. 24, pp. 145–184). Taylor & Francis Group, 6000 Broken Sound Parkway NW, Suite 300, Boca Raton, FL 33487-2742: CRC Press. <https://doi.org/10.1201/9781315183176-4>
- Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable Test Problems for Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization* (pp. 105–145). London: Springer-Verlag. [https://doi.org/10.1007/1-84628-137-7\\_6](https://doi.org/10.1007/1-84628-137-7_6)
- Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4), 28–39. <https://doi.org/10.1109/MCI.2006.329691>
- Edgeworth, F. Y. (1881). *Mathematical Psychics: An Essay on the Application of Mathematics to the Moral Sciences*. (C. Kegan Paul and Co, Ed.). Londres.
- Eiben, A. E. (2003). Multiparent Recombination in Evolutionary Computing (pp. 175–192). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-18965-4\\_6](https://doi.org/10.1007/978-3-642-18965-4_6)
- Eiben, A. E., & Smith, J. E. (2015). *Introduction to Evolutionary Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg. <https://doi.org/10.1007/978-3-662-44874-8>
- Enrique Peñalosa. (2013). Por qué los autobuses representan la democracia en acción | TED Talk | TED.com. Recuperado el 18 de julio de 2017, en [https://www.ted.com/talks/enrique\\_penalosa\\_why\\_buses\\_represent\\_democracy\\_in\\_action?language=es](https://www.ted.com/talks/enrique_penalosa_why_buses_represent_democracy_in_action?language=es)

- Environmental Protection Agency. (2017). *Inventory of U.S. Greenhouse Gas Emissions and Sinks: 1990–2015*. Recuperado de [https://www.epa.gov/sites/production/files/2017-02/documents/2017\\_complete\\_report.pdf](https://www.epa.gov/sites/production/files/2017-02/documents/2017_complete_report.pdf)
- Esidian. (2017). Rideschedules. Recuperado el 1 de Julio de 2017, en <https://rideschedules.com/schedule.html?23467>
- Fan, W., & Machemehl, R. B. (2008). Tabu search strategies for the public transportation network optimizations with variable transit demand. *Computer-Aided Civil and Infrastructure Engineering*, 23(7), 502–520. <https://doi.org/10.1111/j.1467-8667.2008.00556.x>
- Fishburn, P. C. (1974). Exceptional Paper—Lexicographic Orders, Utilities and Decision Rules: A Survey. *Management Science*, 20(11), 1442–1471. <https://doi.org/10.1287/mnsc.20.11.1442>
- Fisher, R. A. (1930). *The genetical theory of natural selection*. Oxford: Clarendon Press. <https://doi.org/10.5962/bhl.title.27468>
- Fores, S., Proll, L., & Wren, A. (1999). An Improved ILP System for Driver Scheduling. In N. H. M. Wilson (Ed.), *Computer-Aided Transit Scheduling: Proceedings, Cambridge, MA, USA, August 1997* (pp. 43–61). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-85970-0\\_3](https://doi.org/10.1007/978-3-642-85970-0_3)
- Furth, P. G., & Wilson, N. H. M. (1981). Setting frequencies on bus routes: Theory and practice. *Transportation Research Record: Journal of the Transportation Research Board*, 818, 1–7.
- Glover, F., Laguna, M., & Marti, R. (2007). Principles of tabu search. *Approximation Algorithms and Metaheuristics*, 23, 1–12. <https://doi.org/10.1016/j.ejor.2004.08.004>
- Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc.
- Gordon, R. (2016). *Intelligent Transportation Systems*. Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-14768-0>
- Guzek, M., Pecero, J. E., Dorransoro, B., & Bouvry, P. (2014). Multi-objective evolutionary algorithms for energy-aware scheduling on distributed computing systems. *Applied Soft Computing*, 24, 432–446. <https://doi.org/10.1016/j.asoc.2014.07.010>
- Hadas, Y., & Shnaiderman, M. (2012). Public-transit frequency setting using minimum-cost approach with stochastic demand and travel time. *Transportation Research Part B: Methodological*, 46(8), 1068–1084. <https://doi.org/10.1016/j.trb.2012.02.010>
- Haimes, Y. Y., Lasdon, L. S., & Wismer, D. A. (1971). On a Bicriterion Formulation of the Problems of Integrated System Identification and System Optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-1(3), 296–297. <https://doi.org/10.1109/TSMC.1971.4308298>
- Hassold, S., & Ceder, A. A. (2014). Public transport vehicle scheduling featuring multiple vehicle types. *Transportation Research Part B: Methodological*, 67, 129–143. <https://doi.org/10.1016/j.trb.2014.04.009>
- Holland, J. H. (John H. (1992). *Adaptation in natural and artificial systems : an introductory analysis with applications to biology, control, and artificial intelligence*. MIT Press.

- Houghton, J., Reiners, J., & Lim, C. (2011). Transporte inteligente Cómo mejorar la movilidad en las ciudades. Recuperado de <http://www-05.ibm.com/services/es/bcs/pdf/transporte-inteligente-como-mejorar-la-movilidad-en-las-ciudades.pdf>
- Ibarra-Rojas, O. J., Delgado, F., Giesen, R., & Muñoz, J. C. (2015). Planning, operation, and control of bus transport systems: A literature review. *Transportation Research Part B: Methodological*, 77, 38–75. <https://doi.org/10.1016/j.trb.2015.03.002>
- Israeli, Y., & Ceder, A. (1995). Transit Route Design Using Scheduling and Multiobjective Programming Techniques. In *Computer-Aided Transit Scheduling: Proceedings of the Sixth International Workshop on Computer-aided Scheduling of Public Transport* (pp. 56–75). [https://doi.org/10.1007/978-3-642-57762-8\\_5](https://doi.org/10.1007/978-3-642-57762-8_5)
- Jolliffe, J. K., & Hutchinson, T. P. (1975). A Behavioural Explanation of the Association Between Bus and Passenger Arrivals at a Bus Stop. *Transportation Science*, 9(3), 248–282. <https://doi.org/10.1287/trsc.9.3.248>
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>
- Knowles, J., & Corne, D. (2002). On metrics for comparing nondominated sets. In *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)* (Vol. 1, pp. 711–716). IEEE. <https://doi.org/10.1109/CEC.2002.1007013>
- Koutsopoulos, H. N., Odoni, A., & Wilson, N. H. M. (1985). Determination of headways as function of time varying characteristics on a transit network. *Computer Scheduling of Public Transport*, 2(1), 391–413.
- Koza, J., Keane, M., Streeter, M., Mydlowec, W., Yu, J., & Lanza, G. (2005). *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*. (2006 Springer Science & Business Media, Ed.) (Vol. 5). New York: Springer-Verlag. <https://doi.org/10.1007/b137549>
- Kwan, C. M., & Chang, C. S. (2008). Timetable Synchronization of Mass Rapid Transit System Using Multiobjective Evolutionary Approach. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(5), 636–648. <https://doi.org/10.1109/TSMCC.2008.923872>
- Li, J. (2002). *Fuzzy Evolutionary Approaches for Bus and Rail Driver Scheduling*. The University of Leeds.
- Li, Y., Xu, W., & He, S. (2013). Expected value model for optimizing the multiple bus headways. *Applied Mathematics and Computation*, 219(11), 5849–5861. <https://doi.org/10.1016/j.amc.2012.11.098>
- Luethi, M., Weidmann, U. A., & Nash, A. (2006). Passenger Arrival Rates at Public Transport Stations. In *Transportation Research Board 86th Annual Meeting*. Washington DC, United States: Transportation Research Board.
- Mehr, H. S., Rahnama, M. R., & Shokouhi, M. A. (2016). Optimization of Main Public Transport Paths Based on Accessibility — Case Study : Mashhad , Iran, 19(1), 114–128.
- Nebro, A. J., Durillo, J. J., Luna, F., Dorronsoro, B., & Alba, E. (2009). MOCell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, 24(7), 726–746. <https://doi.org/10.1002/int.20358>

- Newell, G. F. (1971). Dispatching Policies for a Transportation Route. *Transportation Science*, 5(1), 91–105. <https://doi.org/10.1287/trsc.5.1.91>
- Okabe, T., Yaochu Jin, & Sendhoff, B. (2003). A critical survey of performance indices for multi-objective optimisation. In *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*. (Vol. 2, pp. 878–885). IEEE. <https://doi.org/10.1109/CEC.2003.1299759>
- Osyczka, A. (1985). Multicriteria optimization for engineering design. In J. S. Gero (Ed.), *Design Optimization* (pp. 193–227). Academic Press.
- Pareto, V. (1896). *Cours d'économie politique*. (F. Rouge, Ed.), *Rouge, Lausanne* (Vol. I and II). Lausana, Suiza: Librairie de l'Université.
- Pattnaik, S. B., Mohan, S., & Tom, V. M. (1998). Urban Bus Transit Route Network Design Using Genetic Algorithm. *Journal of Transportation Engineering*, 124(4), 368–375. [https://doi.org/10.1061/\(ASCE\)0733-947X\(1998\)124:4\(368\)](https://doi.org/10.1061/(ASCE)0733-947X(1998)124:4(368))
- Potter, S. (2003). Transport energy and emissions: urban public transport. *Handbook of Transport and the Environment (Handbooks in Transport)*, 247–262.
- Quintero, J. A. (2005). *Modelo de optimización para vehículos de transporte público colectivo urbano*. Universidad Nacional de Colombia Sede Manizales.
- Ramírez, S., Hernandez, L., & Varela, H. (2015). Diseños de rutas de transporte público, 1171–1181.
- Rao, S. (1984). Multiobjective optimization in structural design with uncertain parameters and stochastic processes. *AIAA Journal*, 22(11), 1670–1678. <https://doi.org/10.2514/3.8834>
- Rao, S. S. (2009). *Engineering optimization : theory and practice* (4th ed.). John Wiley & Sons, Inc.
- Riquelme, N., Von Lucken, C., & Baran, B. (2015). Performance metrics in multi-objective optimization. In *2015 Latin American Computing Conference (CLEI)* (pp. 1–11). IEEE. <https://doi.org/10.1109/CLEI.2015.7360024>
- Schaffer, D. (1984). Some experiments in machine learning using vector evaluated genetic algorithms. Nashville, Tennessee.
- Schott, J. R. (1995). Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. *Massachusetts Institute of Technology, Boston, MA*.
- Serafini, P. (1994). Simulated Annealing for Multi Objective Optimization Problems. In G. H. Tzeng, H. F. Wang, U. P. Wen, & P. L. Yu (Eds.), *Multiple Criteria Decision Making: Proceedings of the Tenth International Conference: Expand and Enrich the Domains of Thinking and Application* (pp. 283–292). New York, NY: Springer New York. [https://doi.org/10.1007/978-1-4612-2666-6\\_29](https://doi.org/10.1007/978-1-4612-2666-6_29)
- Shen, Y., & Kwan, R. S. K. (2001). Tabu Search for Driver Scheduling. In S. Voß & J. R. Daduna (Eds.), *Computer-Aided Scheduling of Public Transport* (pp. 121–135). Berlin, Heidelberg: Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-56423-9\\_7](https://doi.org/10.1007/978-3-642-56423-9_7)
- United Nations. (2014). *World Urbanization Prospects: The 2014 Revision, Highlights (ST/ESA/SER.A/352)*. New York, United. <https://doi.org/10.4054/DemRes.2005.12.9>

- Verbas, I., & Mahmassani, H. (2013). Optimal Allocation of Service Frequencies over Transit Network Routes and Time Periods. *Transportation Research Record: Journal of the Transportation Research Board*, 2334, 50–59. <https://doi.org/10.3141/2334-06>
- Verbas, I., & Mahmassani, H. (2015). Integrated Frequency Allocation and User Assignment in Multimodal Transit Networks. *Transportation Research Record: Journal of the Transportation Research Board*, 2498, 37–45. <https://doi.org/10.3141/2498-05>
- Von Lüken, C., Barán, B., & Brizuela, C. (2014). A survey on multi-objective evolutionary algorithms for many-objective problems. *Computational Optimization and Applications*. <https://doi.org/10.1007/s10589-014-9644-1>
- Washburn, D., & Sindhu, U. (2010). Helping CIOs Understand “Smart City” Initiatives. *Cambridge, MA: Forrester Research, Inc.*
- Wierzbicki, A. P. (1980). The Use of Reference Objectives in Multiobjective Optimization. *Multiple Criteria Decision Making Theory and Application*, (August), 468–486. [https://doi.org/10.1007/978-3-642-48782-8\\_32](https://doi.org/10.1007/978-3-642-48782-8_32)
- Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1), 67–82. <https://doi.org/10.1109/4235.585893>
- Yu, P. L. (1973). A Class of Solutions for Group Decision Problems. *Management Science*, 19(8), 936–946. <https://doi.org/10.1287/mnsc.19.8.936>
- Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, 8(1), 59–60. <https://doi.org/10.1109/TAC.1963.1105511>
- Zavala, G., Nebro, A. J., Luna, F., & Coello, C. A. C. (2015). Structural design using multi-objective metaheuristics. Comparative study and application to a real-world problem. *Structural and Multidisciplinary Optimization*, (2016), 1–22. <https://doi.org/10.1007/s00158-015-1291-3>
- Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2), 173–195. <https://doi.org/10.1162/106365600568202>
- Zitzler, E., & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms — A comparative case study (pp. 292–301). Springer, Berlin, Heidelberg. <https://doi.org/10.1007/BFb0056872>
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4), 257–271. <https://doi.org/10.1109/4235.797969>
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. M., & da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2), 117–132. <https://doi.org/10.1109/TEVC.2003.810758>