

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
DIVISIÓN DE ESTUDIOS DE POSTGRADO  
POSTGRADO EN COMPUTACIÓN



**MECANISMOS DE HIBRIDACIÓN PARA LA SOLUCIÓN DE UN  
PROBLEMA DE OPTIMIZACIÓN MULTIOBJETIVO. CASO DE  
ESTUDIO: CONTROL DE FLUJO VEHICULAR EN RAMPAS DE  
ACCESO EN AUTOPISTAS**

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

Autor: Javier A. Maldonado C.  
Tutor: Jhon E. Amaya.

Trabajo de grado presentado ante la ilustre Universidad de Los Andes como requisito parcial  
para optar al grado de *Magister Scientiae en Computación*

Mérida, Octubre 2012

C.C. Reconocimiento

UNIVERSIDAD DE LOS ANDES  
FACULTAD DE INGENIERÍA  
DIVISIÓN DE ESTUDIOS DE POSTGRADO  
POSTGRADO EN COMPUTACIÓN



**MECANISMOS DE HIBRIDACIÓN PARA LA SOLUCIÓN DE UN  
PROBLEMA DE OPTIMIZACIÓN MULTIOBJETIVO. CASO DE  
ESTUDIO: CONTROL DE FLUJO VEHICULAR EN RAMPAS DE  
ACCESO EN AUTOPISTAS**

[www.digital.ula.ve](http://www.digital.ula.ve)

Autor: Javier A. Maldonado C. [jmaldo@unet.edu.ve](mailto:jmaldo@unet.edu.ve)  
Tutor: Jhon E. Amaya. [jedgar@unet.edu.ve](mailto:jedgar@unet.edu.ve)

Mérida, Octubre 2012

C.C. Reconocimiento

# Índice

Resumen.....	1
Introducción.....	2
Estructura del Documento.....	4
Capítulo 1: El Problema.....	5
1.1 Planteamiento y formulación del problema .....	5
1.2 Antecedentes.....	9
1.3 Justificación .....	11
1.4 Alcance.....	12
1.5 Objetivos .....	12
1.5.1 Objetivo General.....	12
1.5.2 Objetivos Específicos.....	12
1.6 Metodología.....	13
1.7 Resultados Esperados.....	15
Capítulo 2: Marco Teórico .....	16
2.1 Optimización Multiobjetivo.....	16
2.1.1 Definición de problemas de optimización multiobjetivo .....	17
2.1.2 Fundamentos matemáticos.....	20
2.1.3 Óptimo de Pareto.....	21

2.1.4	Proceso de solución de un problema de optimización multiobjetivo	24
2.1.5	Métodos tradicionales de optimización	27
2.2	Algoritmos Evolutivos Multiobjetivo	35
2.2.1	Técnicas de algoritmos evolutivos multiobjetivo	39
2.3	Heurísticas	53
2.3.1	Algoritmos exactos vs heurísticas	54
2.3.2	Algoritmos iterativos	57
2.4	Metaheurísticas	58
2.4.1	Algoritmos Genéticos (GA)	59
2.4.2	Variable Neighbourhood Search (VNS)	63
2.4.3	Hill Climbing (HC)	65
Capítulo 3: Metodología		67
Capítulo 4: Desarrollo		71
4.1	Formulación del problema	71
4.2	Modelado del problema	75
4.3	Optimización del problema	83
4.4	Implementación de la solución	88
4.4.1	Implementación de NSGA-II	88
4.4.2	Implementación del Hill Climbing (HC)	93
Capítulo 5: Discusión y análisis de resultados		103
5.5	Métodos comparativos	107

5.5.1	Indicador de Hypervolume .....	107
5.5.2	Indicador R2 .....	109
5.6	Comparación .....	110
	Conclusiones.....	115
	Recomendaciones.....	117
	Bibliografía.....	118

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## Índice de figuras

Figura 1: Asignación valor de adaptación basado en la clasificación por dominancia ( <i>fitness assignment nsga</i> ) (Von Lücken M., 2003).....	41
Figura 2: Asignación del valor de adaptación compartido ( <i>fitness sharing</i> ) (Von Lücken M., 2003).....	42
Figura 3: Pseudocódigo del algoritmo SPEA (Von Lücken M., 2003) ....	44
Figura 4: Procedimiento de clasificación por no dominancia ( <i>fast nondominated sort</i> ) del NSGA-II (Von Lücken M., 2003).....	49
Figura 5: Función de estimación de densidad ( <i>crowding distance assignment</i> ) (Von Lücken M., 2003) .....	51
Figura 6: Pseudocódigo del algoritmo NSGA-II.....	52
Figura 8: Pseudocódigo de un algoritmo genético simple .....	62
Figura 12: Esquema de una rampa de acceso.....	74
Figura 13: Estructura topológica de la autopista, separada en 8 segmentos. (Meng, 2009) .....	75
Figura 14: Gráfica de F1 contra F2 .....	92
Figura 15: Pseudocódigo del Hill Climbing utilizado.....	95
Figura 16: Gráfico de dispersión de las funciones objetivo 1 y 2 luego del HC.....	99
Figura 17: Frontera de Pareto generada por el NSGA-II y el Algoritmo Híbrido comparado en 100 generaciones .....	103
Figura 10: Ejemplo de cálculo para el indicador de hypervolume .....	108

Figura 11: Dos aproximaciones a un problema multiobjetivo .....	109
Figura 18: Índice de Hypervolume. Híbrido 100 generaciones contra NSGA-II 100 generaciones .....	111
Figura 19: Índice R2. Híbrido 100 generaciones contra NSGA-II 100 generaciones .....	112

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## Índice de tablas

Tabla 1: Resultados del NSGA-II. ....	90
Tabla 2: Promedio obtenido .....	91
Tabla 3: Resultados del algoritmo híbrido .....	97
Tabla 4: Promedios de las funciones objetivo del algoritmo híbrido.....	98
Tabla 5: Variables involucradas en una solución .....	100
Tabla 6: Resultado de las funciones objetivo .....	101
Tabla 7: Frentes de Pareto. Híbrido 100 generaciones contra NSGA-II en 200, 500, 1000 y 1500 generaciones.....	104
Tabla 8: Índices de Hypervolume y R2. Híbrido de 100 generaciones contra NSGA-II de 200, 500, 1000 y 1500 .....	113

## Resumen

En las rampas de acceso de las autopistas, existe la posibilidad de congestión motivada al tráfico afluente que llega de manera aleatoria y en algún momento puede superar la capacidad de las vías. El proyecto tiene por objetivo proponer un algoritmo de solución para optimizar el control de accesos (*ramp metering*) y salidas de las autopistas mediante un modelo de algoritmos metaheurísticos híbridos, en función de las características de la inequidad en los tiempos de espera por parte de los usuarios en proporción a los viajes de corta distancia y el rendimiento total del sistema vial en estudio. Dado el modelo matemático, se realizó un experimento numérico, mediante dos algoritmos evolutivos: *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) y un híbrido, donde se realizaron comparaciones en términos de calidad de los resultados obtenidos tomando como referencia la maximización el flujo vehicular. Los resultados obtenidos por el algoritmo híbrido resultaron notablemente superiores.

Palabras claves: metaheurística, multiobjetivo, optimización de tráfico en rampas de acceso, control de accesos.

## Introducción

Los problemas multiobjetivo se encuentran frecuentemente en situaciones de la vida diaria. Para el ser humano muchos de ellos son resolubles mediante un razonamiento medianamente complejo, en donde intuitivamente se puede lograr un balance adecuado entre las distintas condiciones que se plantean en el problema tratado. Sin embargo existen otras muchas situaciones que no pueden ser resueltas de manera intuitiva, para esto se han desarrollado diferentes estrategias para llegar a soluciones más acertadas con mecanismos más precisos, algunos de ellos apoyados en recursos computacionales, han alcanzado gran precisión y aceptación, así como también una variedad importante de algoritmos y métodos en esta área. Dos de estos métodos computacionales son los Algoritmos Genéticos y las Metaheurísticas, las cuales permiten configurar escenarios para la resolución de problemas reales, sin embargo pueden requerir alto costo computacional, es entonces donde se puede realizar la combinación de ambos métodos para generar algoritmos híbridos, los cuales aprovechan las bondades de los dos algoritmos combinados, produciendo frecuentemente mejores resultados en términos de rendimiento y calidad de los resultados.

Este trabajo consiste en la creación de un algoritmo híbrido, capaz de solucionar el problema planteado por (Meng, 2009), donde se describe un problema complejo para controlar el flujo vehicular que ingresa a una autopista,

maximizando el rendimiento del sistema vial completo mejorando entonces los tiempos de espera de los usuarios de la vía.

Las vías de alta velocidad han sido diseñadas con el fin de aumentar las tasas del flujo vehicular en los sistemas de transporte. Sin embargo para acceder a estas vías, es necesario hacerlo a través de rampas de acceso, las cuales permiten el cambio entre las calles de tránsito en las ciudades a las autopistas. Los accesos ocasionalmente se saturan por la afluencia vehicular, la cual puede superar la capacidad de dichas rampas. Es por ello que surge la necesidad de implementar dispositivos de control de tránsito en las rampas de acceso (*ramp metering*), con el fin de evitar la saturación de las vías de acceso, disminuyendo los tiempos de espera de los usuarios y mejorando el desempeño total del sistema de transporte (Zhang & Levinson, 2005).

Las implementaciones de los controles en las rampas de acceso a las autopistas, conllevan a un problema adicional, la inequidad en los tiempos de espera de los usuarios en las entradas (Zhang & Levinson, 2005). Algunos de ellos realizan viajes cortos y deben esperar un tiempo alto en proporción al recorrido que realizarán, por el contrario, algunos usuarios realizan viajes largos y deben esperar un tiempo relativamente corto en comparación con su trayecto. Siendo ambas esperas similares, son desproporcionadas comparándolas con sus tiempos de traslado efectivo.

## Estructura del Documento

**Capítulo 1,** El Problema. Se definen los antecedentes, problema, justificación, objetivos, metodología y el alcance del proyecto. Describiendo de donde y porque surge el problema, lo que intenta solucionar y la forma en que se desarrollará el proyecto.

**Capítulo 2,** Marco teórico. Contiene los fundamentos teóricos que enmarcan la formulación de problemas multiobjetivo, así como también las técnicas empleadas para su resolución, enfocado a los basados en algoritmos genéticos y metaheurísticas.

**Capítulo 3,** Metodología. En este capítulo se describe la metodología empleada en la investigación, orientada a las aplicaciones de resolución de problemas multiobjetivo.

**Capítulo 4,** Desarrollo. Se hace una descripción detallada de los algoritmos, elementos de software creado para la resolución del problema planteado y los componentes que intervienen en el funcionamiento de la implementación realizada.

**Capítulo 5,** Discusión y Análisis de Resultados. En este capítulo se discuten y analizan los resultados obtenidos de la investigación y posteriormente se exponen las conclusiones y recomendaciones para futuras investigaciones.

# Capítulo 1: El Problema

En este capítulo se definirán los antecedentes que soportan el marco de esta investigación, el origen del problema y su interés, se definirá el problema de manera concreta detallando la importancia, objetivos, metodología y alcance del trabajo planteado.

## **1.1 Planteamiento y formulación del problema**

El control del flujo de vehículos en las rampas de acceso de las autopistas (*ramp metering*), son aplicaciones que pretenden optimizar el flujo automotor mediante la evaluación del tráfico y la regulación en función de su comportamiento. Estas aplicaciones permiten mitigar la congestión generada por los automóviles en los accesos y líneas principales de autopistas, rampas de ascenso y descenso. Los sistemas de este tipo proporcionan mayor rendimiento en las vías, disminución del tiempo total de viaje y mejora en la seguridad del tránsito (Zhang & Levinson, 2005).

Otro de los objetivos de los controles en las rampas de acceso, es reducir el retraso causado a los conductores en las recurrentes “trancas” de tráfico, motivado a las restricciones de acceso ubicadas en las entradas de las autopistas. En diferentes experimentos se ha demostrado que los retrasos pueden mejorarse con técnicas que tomen en cuenta las variables de tiempos

de llegada y volumen de flujo vehicular, en vez de sólo un control temporal de luces como un semáforo común y corriente (Gomes G. , 2004).

Existe una situación fundamental en el desempeño del control de flujo de las rampas de acceso, denominada *Problema de inequidad de la medición de la tasa de tráfico en las rampas*. (Pinnell, Drew, McCasland, & Wattleworth, 1967). Consiste en favorecer los usuarios con viajes largos en comparación de aquellos que realizan recorridos cortos. Dentro de esta situación, existen dos tipos de inequidad: temporal y espacial. La temporal mide las diferencias de tiempo de viaje, retraso y velocidad en los conductores que utilizan la misma ruta pero llegan a la rampa en diferentes momentos. La espacial, se refiere a la medida de las diferencias de distancia entre los conductores que llegan a las rampas al mismo tiempo (Meng, 2009).

Desde el punto de vista administrativo, el objetivo de la implementación de estos controles de flujo, es mejorar la eficiencia de las rampas de acceso a las autopistas (Meng, 2009).

Una de las técnicas utilizadas en la solución de problemas de optimización multiobjetivo es la metaheurística, la cual consiste en un procedimiento de alto nivel para el desarrollo de una heurística orientada la búsqueda de una solución óptima de un problema, así como también proveer un marco de trabajo conceptual para este desarrollo. La heurística consiste en la búsqueda de la mejor solución disponible para un problema en particular,

partiendo de un grupo de soluciones y escogiendo la mejor según algunos criterios propios de su naturaleza, sea por ejemplo, maximización de utilización de recursos o minimización de alguna variable que afecte la solución. Las técnicas heurísticas se han utilizado en soluciones a problemas de optimización, pudiéndose aplicar a una gran variedad, convirtiéndola en una técnica predominante en los problemas de optimización (Pilegaard, 1998).

Existen diferentes técnicas de búsquedas heurísticas para la resolución de problemas, algunas de ellas son: *tabu search*, *simulated annealing*, *hill climbing*, entre otros. La *tabu search* usa una memoria para guiar su búsqueda en el espacio de las soluciones, de tal forma que memoriza las que ya ha revisado y las convierte en prohibidas (tabú), en la toma de decisiones del siguiente punto de búsqueda, lo cual permite la exploración del espacio de soluciones evitando estancarse en óptimos locales. El *simulated annealing* se basa en un proceso metalúrgico reconocido: cuando un metal se calienta a temperaturas muy elevadas, sus átomos pueden moverse a altas velocidades, y si es enfriado muy lentamente, los átomos se arreglan en estructuras o patrones de mínima entropía, suministrando un metal mucho más resistente que antes. En las primeras iteraciones de éste método describen grandes perturbaciones en las variables de decisión y en la función a optimizar, a medida que un parámetro va ajustándose lentamente (equivalente a la disminución de temperatura), las nuevas soluciones serán más selectivas. La técnica *hill climbing* recibe este nombre porque trata de elegir en cada paso un estado con

un valor heurístico más alto que el actual, para lo cual compara este último con todos sus vecinos para así decidir si existe una solución mejor que la conseguida hasta el momento, en cuyo caso se desplazará hacia ésta en la próxima iteración. Tiene algunos inconvenientes si se encuentra en un máximo local o en una meseta, motivado a que el método hace comparaciones locales y no una exploración exhaustiva de todo el espacio de soluciones. Es un algoritmo determinístico (Coello C. , 2010) (De la Cruz, 2003).

La solución de problemas multiobjetivo puede llevarse a cabo mediante el uso de algoritmos genéticos (GA) tales como NSGA II, SPEA2, IMOEA, entre otros (Coello, Lamont, & Van Veldhuizen, 2007). La velocidad de convergencia de los GA pueden afectar la calidad de los resultados, si converge más rápido (este problema es conocido como *convergencia prematura*), el GA es posible que esté perdiendo diversidad en las soluciones intermedias prematuramente, por lo cual pudiera perderse información relevante en la generación de las soluciones por medio de estas técnicas. Para solucionar esta situación, se puede mejorar la selección de la nueva generación de individuos con el uso de técnicas de heurística, las cuales permiten buscar y elegir los mejores individuos para que formen parte de la nueva generación, de tal manera que se pueda mejorar la tarea de selección de los AG tradicionales. Este tipo de algoritmos híbridos también se conoce con el nombre de *algoritmos meméticos*, los cuales permiten la combinación de diferentes técnicas para mejorar los resultados en los problemas de optimización multiobjetivo. Algunos de las

heurísticas utilizadas en combinación con los GA son *hill climbing*, optimización por colonia de hormigas, *synchronous particle local search* (SPLS) y *fuzzy global-best* (f-gbest), entre otras (Liu, 2007).

La presente investigación tiene como finalidad diseñar e implementar un algoritmo híbrido para la resolución de un problema multiobjetivo, tomando como caso de estudio el planteado por (Meng, 2009), con el fin de conseguir soluciones que optimicen el flujo vehicular dentro del sistema vial propuesto en función del comportamiento descrito por el mencionado trabajo, mediante el control de acceso en las rampas (ramp metering), que sirva como modelo inicial y extensible a cualquier autopista.

## 1.2 Antecedentes

Zhang y Levinson (2005), realizan un trabajo acerca de los controles de acceso en las rampas de las autopistas en diferentes localidades de Estados Unidos, donde resaltaron que estos dispositivos permitían minimizar el costo total de los traslados en estas vías. Sin embargo conducen a un problema adicional, balance entre eficiencia y equidad. Hasta entonces el comportamiento de las rampas de acceso es modelado como un problema bi-objetivo, obteniendo una solución no muy efectiva. Para mejorarla, Zhang y Levinson crean una nueva estrategia de control que permita optimizar el comportamiento dentro de las rampas de acceso mediante un algoritmo denominado BEEEX (*Balanced Efficiency and Equity*), el cual busca minimizar el tiempo de viaje total

ponderado, asignando peso al uso de la autopista y el retraso en la rampa de entrada.

En el trabajo realizado por Scariza (2003), en el MIT, identifica dos tipos de algoritmos para el control del comportamiento de las rampas de acceso: local y coordinado. Los algoritmos locales solo toman en cuenta las condiciones del tráfico de una rampa en particular. Los algoritmos coordinados lo hacen tomando en cuenta el comportamiento de las rampas presentes en un área. Se realiza la revisión de 4 algoritmos: 1) ALINEA que es un algoritmo de control de rampas local. 2) ALINEA/Q, pero es más eficiente manejando las colas en las rampas. 3) FLOW el cual pertenece al grupo de algoritmos coordinados, trata de mantener el tráfico por debajo de un umbral predefinido de cuello de botella. 4) El Algoritmo Enlazado, otro algoritmo coordinado que busca la optimización de una función objetivo lineal cuadrática. Cada uno de estos algoritmos fueron probados en la autopista M27 cerca de Southampton, UK y en una red de vías genérica.

Según Gomes y Horowitz (2006), el problema del control de acceso a las rampas en autopistas, se puede plantear utilizando un modelo de transmisión de celda llamado *Asymmetric Cell Transmission Model* (ACTM), esta formulación toma en cuenta las condiciones de descongestión y congestión de las vías, e incluye límites superiores para medición de flujo vehicular y longitud de la cola en las rampas. Muestra que una solución resultante de un problema de optimización no lineal puede conseguirse mediante la resolución de un

programa lineal, siempre y cuando algunas condiciones puedan alcanzarse. Una de ellas es que el tráfico afluente no puede ser detenido por una congestión en la vía principal. La solución alcanzada es capaz de modelar un comportamiento adecuado de las rampas en función del flujo vehicular observado.

### **1.3 Justificación**

El proyecto de investigación servirá como modelo para la solución de un problema complejo basado en datos generados a partir del modelo matemático, en este caso la optimización del tráfico automotor en vías alto flujo vehicular, específicamente en las autopistas. Tomando como base algunos estudios y desarrollos previos, se plantea el uso de una combinación de herramientas de computación emergente que permitan alcanzar soluciones eficientes.

El trabajo contribuirá al campo de la investigación de metaheurísticas aplicadas a problemas multiobjetivo (Pilegaard, 1998), como es el comportamiento eficiente de los controles instalados en las rampas de acceso a autopistas. Siendo el algoritmo desarrollado aquí, una base para estudios futuros en éste y otros campos donde se pueda aplicar dichas técnicas a soluciones complejas.

## **1.4 Alcance**

Diseñar una metaheurística híbrida para proporcionar soluciones al problema de equidad del control de rampas de acceso en autopistas, formulado como un problema multiobjetivo, a partir del planteado por Meng y Khoo (2009), con una arquitectura que permita integrar herramientas de generación de soluciones por medio de algoritmos genéticos y selección de las mejores de éstas mediante búsquedas heurísticas.

## **1.5 Objetivos**

### **1.5.1 Objetivo General**

Diseñar una metaheurística híbrida para la solución de un problema de optimización multiobjetivo, en particular para el control de flujo en rampas de acceso.

### **1.5.2 Objetivos Específicos**

- Revisar la documentación disponible acerca de mecanismos de hibridación para optimización multiobjetivo.
- Modelar el problema en función de las variables que se desean optimizar.

- Diseñar la metaheurística híbrida tomando en cuenta las variables del problema de optimización
- Implementar el algoritmo necesario para el procesamiento de las variables identificadas.
- Evaluar los resultados y comparar con el algoritmo NSGA-II.

## **1.6 Metodología**

Para la realización de la investigación, se seguirá una metodología aplicada al desarrollo de metaheurísticas, la cual abarca desde la concepción inicial del problema, hasta su implementación y mejora. A continuación se describe la metodología planteada según Talbi (2009):

- **Formulación del problema:** En primer lugar el problema debe ser identificado. Entonces se realiza un primer planteamiento. La formulación puede ser imprecisa pero debe hacer un esbozo de los factores internos, externos y objetivos que apoyaran la resolución del planteamiento.
- **Modelado del problema:** Se utilizan como punto de partida los basamentos obtenidos en la formulación del problema, con el fin de generar un modelo matemático abstracto capaz de representar la situación planteada. Se pueden utilizar modelos similares, de tal forma que se pueda reducir a un modelo de optimización bien estudiado.

Usualmente son simplificaciones de la realidad, están compuestos por aproximaciones y, en ocasiones, pasan por alto procesos complejos para representar en un modelo matemático.

- Optimización del problema: Una vez que el problema es modelado, el procedimiento de resolución genera una “buena” solución al planteamiento, ésta puede ser óptima o sub-óptima. Es de notar que se está buscando solución a un modelo abstracto, es por esto que la medición de las soluciones obtenidas, pueden determinar si es exacto o no en este contexto. Se pueden utilizar algoritmos similares ya implementados y probados, así como también se puede incluir conocimiento de una aplicación específica.
- Implementación de la solución: Luego de la optimización del problema, se obtienen una solución la cual es probada de forma práctica y su resultado se marca como “aceptable”, según un indicador definido previamente para realizar la comparación. Algún conocimiento práctico puede ser incluido en la implementación de la solución. Si es inaceptable, el modelo y/o el algoritmo de optimización debe ser mejorado y el proceso de decisión debe repetirse.

La metodología descrita comprende el desarrollo de algoritmos de optimización aplicado a cualquier situación, tomando como punto de partida un problema del mundo real que será objeto de investigación, planteado como una

abstracción en función de las variables a estudiar en un caso en particular, para luego definir un método de optimización, con el fin de obtener un grupo de soluciones posibles y luego evaluarlas con el modelo planteado, determinando si el algoritmo utilizado es aceptable o si necesita hacerse una revisión del modelo o de los algoritmos empleados.

### ***1.7 Resultados Esperados***

Se espera obtener un algoritmo híbrido equivalente al NSGA-II, generando así conjuntos de posiblemente mejores soluciones bajo las mismas premisas del problema planteado en el trabajo de Meng y Khoo (2009), el cual proporcionará un punto de partida en el desarrollo de las heurísticas de acuerdo al modelo matemático obtenido de la abstracción del problema.

## Capítulo 2: Marco Teórico

En este capítulo se describen los fundamentos teóricos que soportan esta investigación, se ilustran los conceptos necesarios para una comprensión del entorno teórico del desarrollo realizado.

### 2.1 Optimización Multiobjetivo

Los problemas multiobjetivo aparecen en muchas aplicaciones del mundo real y su solución ha desafiado a los investigadores durante mucho tiempo. Debido a la naturaleza compleja de este tipo de modelos, se han desarrollado muchas técnicas para su optimización dentro de la disciplina de Investigación de Operaciones (IO), sin embargo ha sido necesario la búsqueda de nuevos métodos, puesto que se plantean nuevas situaciones con soluciones tan complejas que no pueden ser resueltas trivialmente por la IO con las técnicas ya existentes.

Una de las primeras aplicaciones en esta área se remonta a 1776 en el tratado *The Wealth of Nations* (Stadler, 1986), en el cual identifica el equilibrio económico de un país como un problema de naturaleza multiobjetivo, desde entonces se han modelado infinidad de situaciones en este campo, siendo uno de las más comunes de los multiobjetivo o multicriterio. Otras aplicaciones bien conocidas son las hechas en materia de teoría de juegos, basado en el análisis de la psicología del adversario. En 1944 John Von Neumann y Oskar

Morgenster mencionaron un problema de optimización en el contexto de intercambio económico social como “un peculiar y confusa mezcla de un montón de problemas conflictivos” que no “son posibles manejar con las matemáticas clásicas”, sin embargo no fue un problema discutido luego, por lo cual no se hizo un aporte importante hasta 1950 (Coello, Lamont, & Van Veldhuizen, 2007).

Los Algoritmos Evolutivos (EA, por sus siglas en inglés), para la solución de optimización de problemas multiobjetivo han proporcionado una aproximación muy interesante para los investigadores, puesto que es posible generar un conjunto de elementos posiblemente óptimos (conocido como óptimo de Pareto), en cada ejecución. Adicionalmente algunos problemas de optimización multiobjetivo son bastante complejos en términos de su naturaleza: espacios de búsqueda muy grandes, incertidumbre, ruido, conjuntos de Pareto disjuntos, entre otros, lo cual hace que una solución a través de las técnicas de Investigación de Operaciones tradicionales no sea factible.

### **2.1.1 Definición de problemas de optimización multiobjetivo**

La mayoría de situaciones o problemas del mundo real requieren la búsqueda de soluciones que logren la optimización de varios objetivos simultáneamente. Esto es en cualquier disciplina: finanzas, ingeniería, toma de decisiones de negocios, entre otros, e igualmente se presenta frecuentemente en situaciones de la vida diaria. Un ejemplo de ello puede ser la compra de un

automóvil, se requiere un auto que cumpla con estética, lujo, potencia, confort, consumo de combustible, pero ajustado a un presupuesto determinado. Nótese que alguno de estos objetivos puede ir en detrimento de otros (por ejemplo el lujo vs. el presupuesto), entonces allí es donde interviene la optimización de múltiples objetivos para obtener un resultado con lo mejor posible de cada uno de ellos. En esta situación sencilla se destacan algunas de las características más importantes de este tipo de problemas (Von Lücken M., 2003):

- Los distintos objetivos pueden ser conflictivos, tener distintas unidades de medida y ser inconmensurables.
- Al existir objetivos contradictorios no existe una solución única que sea mejor que otras con respecto a todos los demás, sino que se tiene un conjunto de alternativas de solución, las cuales representan los mejores compromisos entre todos ellos.
- Solo interesan un conjunto de soluciones factibles, es decir aquellas que satisfacen las restricciones planteadas.
- Se requiere un método para seleccionar la solución que será implementada de todas aquellas que cumplen con las restricciones (conjunto de soluciones factibles), puesto que en problemas multiobjetivo no se puede hablar de un óptimo global.

En una definición un poco más formal, un problema de optimización multiobjetivo está definido por  $k$  diferentes funciones objetivo las cuales deben ser optimizadas simultáneamente, dado que la modificación de alguna de ellas puede ir en detrimento de otra, afectando así la solución calculada en un momento en particular, lo cual se puede conseguir maximizando las  $k$  funciones, minimizándolas o la combinación de maximización y minimización de algunas de ellas. Un problema de optimización multiobjetivo se resume entonces en la maximización o minimización de  $F(x) = (f_1(x), f_2(x), \dots, f_k(x))$  sujeto a  $g_i(x) \leq 0, i = \{1, \dots, m\}$  y  $h_j(x) = 0, j = \{1, \dots, p\}, x \in \Omega$ . Una solución a un Problema de Optimización Multiobjetivo (*Multiobjective Optimization Problem* - MOP), minimiza/maximiza los componentes del vector  $F(x)$  donde  $x$  es un vector de variables de decisión  $n$ -dimensional  $x = (x_1, \dots, x_n)$  de algún universo  $\Omega$ . Cabe destacar que  $g_i(x) \leq 0$  y  $h_j(x) = 0$  representan restricciones que deben ser cumplidas mientras se maximiza o minimizan las funciones objetivo,  $F(x)$  y  $\Omega$  contienen todas las posibles  $x$  que pueden ser utilizadas para satisfacer una evaluación de  $F(x)$ .

Por lo tanto un MOP consiste en  $k$  objetivos representados por  $k$  funciones objetivos,  $m + p$  restricciones en las funciones objetivo y  $n$  variables de decisión. Las  $k$  funciones objetivos pueden ser lineales o no lineales, discretas o continuas.

### 2.1.2 Fundamentos matemáticos

Los orígenes de la fundamentación matemática de la optimización multiobjetivo se remonta al período de 1895 a 1906 (Stadler, 1986), donde Georg Cantor y Felix Hausdorff, mostraron los *fundamentos espacios de dimensión infinita ordenados*. Sin embargo fue el concepto del problema de vector máximo introducido por Kuhn & Tucker (1951), dándole un nuevo impulso a la optimización multiobjetivo. La llamada "eficiencia adecuada" en el contexto de la optimización multiobjetivo se formuló también en este documento seminal que se puede considerar como el primer intento serio para derivar una teoría en este campo.

No fue sino hasta la década de 1960 que los fundamentos de la optimización multiobjetivo se consolidaron y fueron tomados en serio por los matemáticos puros, cuando Leonid Hurwicz (1964) hiciera una generalización de los resultados de Kuhn y Tucker hacia espacios vectoriales topológicos.

Las primeras aplicaciones se llevaron a cabo en la década de 1960, donde los problemas multiobjetivo en investigación pública se volvieron comunes y el término "compensación" se volvió favorito entre los administradores y planificadores (Cohon & Marks, 1975). Entonces esta área comenzó a ser muy común en las matemáticas económicas y muchas técnicas fueron desarrolladas por analistas de sistemas y especialistas en teoría de decisiones para problemas en el sector público y privado. En ese momento

aumentó el interés por las propuestas de Kuhn y Tucker sobre teoría de vector máximo hechas en la década de 1960. Luego de allí surgieron muchos otros trabajos y técnicas en otras áreas donde se comenzaron a modelar problemas reales como multiobjetivo.

### **2.1.3 Óptimo de Pareto**

Dentro del campo de optimización multiobjetivo la definición de óptimo varía un poco, más que una “mejor solución” global, se persigue conseguir un balance entre los objetivos. La definición de óptimo adoptada en este campo fue propuesta por Francis Edgeworth (1881) la cual fue luego generalizada por Vilfredo Pareto (Pareto V. , 1896). Esto fue denominado luego por otros autores como el Óptimo Edgeworth-Pareto, la cual se conoce como Óptimo de Pareto.

La definición consiste en que dado un  $x^*$  es un óptimo de Pareto si no existe un vector factible  $x$  que cause un decrecimiento en un criterio sin causar un incremento simultaneo en al menos uno de los otros criterios, esto es en el caso de minimización.

Las soluciones óptimas de Pareto para la optimización multiobjetivo, está compuesta por aquellas dentro de un espacio de decisión (o espacio de búsqueda), cuyo componentes del vector objetivo no pueden ser simultáneamente mejorados. También son denominadas soluciones no

inferiores, admisibles o eficientes. Los vectores seleccionados de este conjunto son soluciones aceptables óptimas de Pareto, variables de decisión.

La frontera de Pareto se define como el conjunto gráfico de soluciones o vectores no dominantes. En general, no existe una expresión analítica de los puntos o la superficie que contiene estos puntos y en la mayoría de los casos es imposible. La manera de obtener esta gráfica, es calculando muchos puntos en  $\Omega$  y calculando su  $f(\Omega)$ . Cuando se tiene una población suficiente de estos elementos, es posible determinar los puntos no dominantes y producir la frontera de Pareto. En los problemas de optimización de un objetivo existe una única solución óptima, en cambio en los problemas multiobjetivo existe un número incontable de soluciones, las cuales forman la frontera de Pareto. Cada uno de los puntos de la frontera de Pareto es un vector, cuyos componentes representa una compensación en el espacio de decisión o espacio de soluciones de Pareto.

A continuación se presentan algunos conceptos básicos de la teoría de Pareto (Von Lücken M., 2003).

Dominancia de Pareto: un vector  $\mathbf{u} = (u_1, \dots, u_k)$  se dice que domina a otro vector  $\mathbf{v} = (v_1, \dots, v_k)$  sí y sólo sí:  $\mathbf{u}$  es parcialmente mejor que  $\mathbf{v}$ , esto es  $\forall i \in \{1, \dots, k\}, u_i$  es mejor o igual  $v_i$  y  $\exists i \in \{1, \dots, k\} | u_i$  es estrictamente mejor que  $v_i$ . Es decir  $\mathbf{u}$  domina a  $\mathbf{v}$  si  $\mathbf{u}$  es mejor o igual a  $\mathbf{v}$  en todos los objetivos y

estrictamente en al menos uno de ellos. En este caso, se dice que existe una dominancia de  $\mathbf{u}$  respecto a  $\mathbf{v}$  lo que se denota como  $\mathbf{u} \succ \mathbf{v}$  ( $\mathbf{u}$  es mejor que  $\mathbf{v}$ )

Note que la definición anterior de dominancia Pareto es independiente si se trata de minimizaciones, maximizaciones o combinaciones de ambas.

Dominancia de Pareto débil: un vector  $\mathbf{u} = (u_1, \dots, u_k)$  se dice que domina de forma débil a otro vector  $\mathbf{v} = (v_1, \dots, v_k)$  sí y sólo sí:  $\mathbf{u}$  es parcialmente mejor o igual que  $\mathbf{v}$ , esto es  $\forall i \in \{1, \dots, k\}, u_i$  es mejor o igual que  $v_i$ . Es decir  $\mathbf{u} \succ \mathbf{v}$  o  $\mathbf{u} = \mathbf{v}$ .

Dominancia de Pareto Estricta: un vector  $\mathbf{u} = (u_1, \dots, u_k)$  se dice que domina de forma estricta a otro vector  $\mathbf{v} = (v_1, \dots, v_k)$  sí y sólo sí:  $\mathbf{u}$  es mejor que  $\mathbf{v}$  para todos los objetivos considerados, esto es  $\forall i \in \{1, \dots, k\}, u_i$  es mejor que  $v_i$

Relación de No dominancia: un vector  $\mathbf{u} = (u_1, \dots, u_k)$  se dice que no domina al vector  $\mathbf{v} = (v_1, \dots, v_k)$  sí y sólo si la relación  $\mathbf{u} \succ \mathbf{v}$  no se cumple.

Según las definiciones anteriores, se dice que una solución es Óptimo de Pareto si no existe algún vector  $\mathbf{v}$  que también pertenezca al conjunto de soluciones factibles que domine a  $\mathbf{u}$ , entonces se puede decir que  $\mathbf{u}$  es un vector óptimo de Pareto. El conjunto de **soluciones Pareto óptimas** son también llamadas *no inferiores*, *admisibles*, o *soluciones eficientes*, mientras que sus vectores objetivos correspondientes son *no dominados*.

#### **2.1.4 Proceso de solución de un problema de optimización multiobjetivo**

Un MOP es considerado resuelto cuando es encontrado el conjunto Pareto Óptimo. En la implementación de problemas reales, se requiere de una única solución, sin embargo desde un punto de vista matemático todas las soluciones del conjunto óptimo encontrado son igualmente buenas. Es en esta situación donde interviene un componente denominado tomador de decisiones (*Decision Maker* – DM), el cual se encarga de proporcionar la información de preferencia necesaria para seleccionar una solución del conjunto de Pareto. De esta manera el proceso de solución de un MOP puede dividirse en dos procesos:

- Optimización o búsqueda: Por el cual se explora el conjunto de soluciones factibles en busca de soluciones Pareto Óptimas.
- Toma de decisión: En este proceso se selecciona una solución de compromiso adecuada a partir del conjunto de Pareto Óptimo hallado.

Existen numerosos intentos para clasificar las diferentes técnicas de optimización multiobjetivo utilizadas actualmente. Sin embargo, es importante identificar dos estados de la solución de este tipo de problemas: La optimización de las diferentes funciones objetivo involucradas y el proceso de decisión

acerca del tipo de compensación apropiado para el problema tratado (también llamado proceso de decisión multicriterio).

Cohon y Marks (1975) han propuesto una de las más populares clasificaciones de técnicas en la comunidad de investigación de operaciones:

1. Técnicas de generación: una articulación a posteriori de las preferencias.
2. Técnicas basadas en la articulación antes de las preferencias: métodos no interactivos.
3. Técnicas que se basan en la articulación progresiva de las preferencias: interacción con el que toma las decisiones.

Sin embargo una clasificación más orientada a la manera en que cada técnica maneja las dos situaciones de buscar y tomar decisiones multicriterio (Van Veldhuizen & Lamont, 2000), (Horn, 1997), es la siguiente:

1. **Articulación de preferencias a priori:** Tomar decisiones antes de buscar (decidir  $\Rightarrow$  buscar). Este grupo de técnicas incluyen aquellas donde se asume que el tomador de decisiones puede ejecutar, ya sea un cierto número de logros deseados o un cierto preorden en los objetivos antes de realizar la búsqueda. Algunas técnicas pertenecientes a este grupo son: Criterio Global (Osyczka, 1984), Programación por Logros (Charnes & Cooper,

1961), (Ijiri, 1965), Método Lexicográfico (Rao, 1986), (Sarma, Sellami, & Houam, 1993), Optimización Min-Max (Coello C. A., 1999), ELECTRE (Benayoun, Roy, & Sussman, 1966), entre otras.

2. **Articulación de preferencias a posteriori:** Hacer la búsqueda antes de tomar decisiones (buscar  $\Rightarrow$  decidir). Estas técnicas no necesitan información de preferencias del tomador de decisiones. Algunas de ellas son las más antiguas en materia de optimización multiobjetivo de los enfoques propuestos. La razón es que la principal idea de estos se deriva directamente de las condiciones de Kuhn-Tucker para soluciones no inferiores. Algunas de estas técnicas son: Combinación Lineal de Pesos (Zadeh, 1963),  $\epsilon$ -Restricción (Osyczka, 1984).

3. **Articulación de preferencias progresiva:** Integración de búsqueda y toma de decisiones (decidir  $\Leftrightarrow$  buscar). Operan normalmente en tres estados: 1) Conseguir una solución no dominante, 2) Obtener la reacción del Tomador de Decisiones con respecto a la solución no dominante conseguida y modificar la preferencia de los objetivos de acuerdo a este, 3) Repetir los pasos anteriores hasta que el Tomador de Decisiones está satisfecho o no se pueden hacer mejoras a las soluciones. Dentro de este grupo se encuentran: Método de Compensación Probabilístico (Goicochea, Duckstein, & Fogel, Diciembre 1976),

STEP (Xevi & Khan), Método de Resolución Secuencial Multiobjetivo (Monarchi, Kisiel, & Duckstein, Agosto 1973), entre otros.

Con los métodos a priori, donde se combinan múltiples objetivos en un único criterio de optimización, se tiene la posibilidad de utilizar estrategias clásicas mono-objetivo sin modificación. Sin embargo es necesario un profundo conocimiento del dominio del problema para realizar una escalarización (transformar los vectores de solución en un único valor) correcta. Este conocimiento a priori usualmente no está disponible, al contrario, generalmente el planteamiento de este tipo de problemas obedece a la necesidad de generar más conocimiento acerca de su dominio, así como también de sus soluciones alternas. Realizar la toma de decisiones luego del proceso de búsqueda, resuelve el problema, sin embargo excluye la información que pueda proporcionar el tomador de decisiones, lo que podría reducir la complejidad del espacio de búsqueda. Los métodos de preferencias progresiva o interactivos superan las desventajas de los métodos a priori y posteriori, permitiendo al tomador de decisiones expresar sus preferencias a medida que avanza en el conocimiento del problema considerado (Von Lücken M., 2003).

### **2.1.5 Métodos tradicionales de optimización**

Antes de las técnicas de computación emergente o algoritmos evolutivos, los problemas de optimización multiobjetivo eran resueltos mediante otros

métodos apoyados en los fundamentos matemáticos que soportan los problemas formulados para su resolución. A continuación se describen algunos métodos tradicionales empleados para la solución de los MOP (Von Lücken M., 2003).

### 2.1.5.1 Método de suma ponderada

En este tipo de métodos las funciones objetivo del problema original se combinan en forma lineal utilizando coeficientes de peso para cada una de ellas, formando así una función escalar a ser optimizada según la importancia otorgada a cada una de las funciones objetivo. Esto transforma un problema de optimización multiobjetivo:

$$\text{Optimizar } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))$$

sujeto a

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0$$

se transforma en un problema de optimización de la forma:

Optimizar

$$y = f(\mathbf{x}) = w_1 \cdot f_1(\mathbf{x}) + \dots + w_k \cdot f_k(\mathbf{x}) = \sum_{i=1}^k w_i \cdot f_i(\mathbf{x})$$

sujeto a

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0$$

donde

$$w_i \geq 0 \text{ para todo } i = 1, \dots, k, \sum_{i=1}^k w_i = 1$$

Los métodos de suma ponderada pueden utilizarse a priori, interactiva o a posteriori. A priori el tomador de decisiones (DM) expresa su preferencia por algún objetivo dado mediante la asignación de los pesos a cada uno de estos, el problema es que esta asignación puede no ser la más adecuada para la naturaleza y tratamiento del problema. Además limita al DM sobre otras alternativas, puesto que el resultado será una única solución a partir de la función importancia construida. Este tipo de métodos no genera un conjunto de soluciones posibles, entonces el criterio de decisión podría ser bastante sesgado. Pueden existir mejores soluciones y no serán tomadas en cuenta.

Para generar un conjunto de soluciones utilizando el método ponderado, es necesario variar los pesos definidos en primera instancia, para entonces así obtener diferentes resultados. La búsqueda de una solución óptima comienza con la selección de un vector de pesos inicial, el cual producirá un resultado. Las variaciones de este vector de pesos se puede hacer según las preferencias del tomador de decisiones o en forma aleatoria, durante un número de iteraciones determinado o hasta que se consigan un cierto número de soluciones. Las variaciones de los pesos se denominan perturbaciones (Chankong & Haimes, 1983) para obtener diferentes soluciones de Pareto Óptimas. De esta manera no se refleja proporcionalmente la importancia

relativa de los objetivos, sino que la variación de los factores representa diferentes puntos del conjunto de Pareto. Luego de obtener dicho conjunto se presenta al DM para seleccionar alguna que satisfaga los requerimientos planteados. La principal desventaja con este método, es que no se pueden generar todas las soluciones de Pareto Óptimas cuando el espacio de soluciones compromiso no es convexo.

Este método también puede utilizarse de forma interactiva donde el DM puede ir ajustando los valores del vector de pesos a partir de uno definido inicialmente, con el fin de ir orientando las soluciones generadas hacia una tendencia deseada, sin embargo puede representar una significativa pérdida de tiempo para el DM.

### **2.1.5.2 Programación de metas**

En este método el tomador de decisiones debe asignar metas deseables para cada objetivo. Estos valores son incorporados al problema como restricciones adicionales, entonces se tratará de minimizar las desviaciones absolutas de cada objetivo con respecto a lo deseado. Una fórmula sencilla de este método se formula de la siguiente manera:

$$\text{Minimizar } \sum_{i=1}^k \left| \frac{f_i(x) - T_i}{T_i} \right| \text{ sujeta a } x \in X_f$$

Donde  $T_i$  denota la meta establecida por el DM para la  $i$ -ésima función objetivo  $f_i$ . El criterio se convierte entonces en minimizar la suma de los valores absolutos de las diferencias entre los valores deseados y los obtenidos.

Esta técnica puede ser muy eficiente en términos de procesamiento si conocemos las metas que se desean obtener y si estas se encuentran en la zona factible, aunque para lograr esto, el DM tiene la tarea de encontrar pesos adecuados para minimizar la distancia a los valores deseados, lo cual es difícil para la mayoría de los casos. Además si la región factible no es fácilmente localizable, puede volverse muy ineficiente. Si es posible hacer aproximaciones lineales parciales de las funciones objetivos puede ser muy útil, en caso de no linealidad generalmente se pueden obtener mejores resultados con otras técnicas.

### 2.1.5.3 Ordenamiento lexicográfico

Este método consiste en ordenar las funciones objetivos según su importancia antes de comenzar el proceso de búsqueda. Luego la solución óptima se obtiene minimizando dichas funciones objetivo, comenzando con la más importante procediendo con el orden asignado a cada uno de los objetivos. Según esta metodología, un problema en general se formula:

$$\text{Optimizar } f_1(x)$$

sujeta a:

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0$$

Obteniendo su solución  $\mathbf{x}_1^*$  y  $f_1^* = f_1(\mathbf{x}_1^*)$ . Posteriormente, el segundo problema se formula como

$$\text{Optimizar } f_2(\mathbf{x})$$

sujeta a:

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0$$

$$f_1(\mathbf{x}) = f_1^*$$

y la solución a este problema se obtiene como  $\mathbf{x}_2^*$  y  $f_2^* = f_2(\mathbf{x}_2^*)$ . Este procedimiento se repite hasta que todos los  $k$  objetivos han sido considerados, entonces el  $k$ -ésimo problema está dado por:

$$\text{Optimizar } f_k(\mathbf{x})$$

sujeta a

$$\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_m(\mathbf{x})) \leq 0$$

$$f_1(\mathbf{x}) = f_1^*$$

⋮

$$f_{k-1}(\mathbf{x}) = f_{k-1}^*$$

La solución obtenida al final  $x_k^*$ , se considera como la solución deseada del problema. Éste método es adecuado cuando se conoce claramente la importancia de cada objetivo, lo que en la práctica no es muy común.

#### 2.1.5.4 Restricciones $\epsilon$

Este método considera una función objetivo principal (o preferida), a optimizar, sujeta a los demás objetivos como restricciones con ciertos niveles de tolerancia  $\epsilon_i$ . Entonces se efectúa la optimización de un solo objetivo, el que ha sido considerado como preferido  $f_r$  sujeto a restricciones adicionales en las otras funciones objetivo. Los niveles de tolerancia  $\epsilon_i$  son alterados luego para generar otros puntos del conjunto de Pareto. Este método puede formularse de la siguiente manera:

1. Encontrar el óptimo de la  $r$ -ésima función objetivo, esto es, encontrar  $x^*$  tal que:

$$f_r(x^*) = f_r^*$$

sujeta a restricciones de la forma

$$f_i(x^*) \leq \epsilon_i$$

donde  $\epsilon_i$  son los valores supuestos de las funciones objetivo que no se quiere exceder, en un contexto de minimización.

2. Repetir (1) para diferentes valores de  $\epsilon_i$ . La información derivada de un conjunto de  $\epsilon_i$  puede ser útil para la toma de decisión final. La

búsqueda se detiene cuando el DM encuentra una solución satisfactoria.

Puede ser necesario repetir el proceso para diferentes índices  $r$ . Para obtener valores adecuados de  $\varepsilon_i$ , se efectúan normalmente optimizaciones individuales para cada una de las funciones objetivo, utilizando técnicas de programación matemática. El inconveniente que presenta esta técnica es el consumo de tiempo elevado y la codificación compleja de las funciones objetivo (incluso imposible en algunos casos), particularmente en problemas con múltiples objetivos. Adicionalmente, tiende a conseguir soluciones débilmente no dominadas, lo que puede no ser adecuado para ciertas aplicaciones.

#### **2.1.5.5 Discusión de los métodos clásicos**

Los distintos métodos tradicionales que han sido presentados cuentan con algunas dificultades mostradas a continuación:

1. El algoritmo de optimización se debe aplicar varias veces para encontrar un conjunto de soluciones óptimas de Pareto. Cada una de estas ejecuciones es independiente y, generalmente, no correlacionada, por lo tanto delinear el frente óptimo de Pareto resulta computacionalmente costoso.
2. La mayoría de los algoritmos requieren conocimiento previo del problema a resolver y son muy sensibles en cuanto a parámetros:

pesos de los objetivos, orden de evaluación, nivel de objetivos, valor de las restricciones, entre otros.

3. Algunos algoritmos son sensibles a la forma del frente de Pareto.
4. La variación entre las diferentes soluciones encontradas depende de la eficiencia del optimizador de un solo objetivo. Podría darse el caso de encontrar siempre la misma solución o soluciones muy parecidas en distintas ejecuciones.
5. En problemas que involucren azar o incertidumbre, los métodos clásicos no son confiables.

Por tanto, se han desarrollado técnicas de computación evolutiva orientadas a la solución de problemas multiobjetivo, las cuales son capaces de superar las limitaciones aquí mencionadas ofreciendo muy buenos resultados, en cuanto a calidad de las soluciones, costo computacional, complejidad y costo de implementación.

## ***2.2 Algoritmos Evolutivos Multiobjetivo***

Los métodos clásicos presentan ciertos inconvenientes y dificultades para la resolución de problemas multiobjetivo, sin embargo se habría logrado un gran éxito en la optimización de un solo objetivo utilizando algoritmos evolutivos, lo cual despertó interés en su desarrollo e investigación en el campo de la optimización multiobjetivo. Para los investigadores, es un punto de partida

importante para nuevas contribuciones de algoritmos originales. Desde el punto de vista del DM es deseable que un método produzca solo unas pocas soluciones para simplificar el proceso de decisión, equilibrando el compromiso de cada uno de los objetivos y un balance entre eficiencia y efectividad. Por ejemplo, el problema clásico de la bolsa, donde se requiere un balance entre el beneficio y el peso asociado a los elementos que pueden entrar en ella (Kellerer, Pferschy, & Pisinger, 2003).

Los algoritmos evolutivos y algoritmos evolutivos multiobjetivo (MOEA – *MultiObjective Evolutionary Algorithms*), tienen la capacidad de representar o codificar una solución individual de distintas formas sencillas (cromosomas), así como calcular directamente los valores asociados a los objetivos, tienen una ventaja considerablemente más robusta que las técnicas tradicionales hasta ahora utilizadas. Las técnicas tradicionales imponen restricciones o mapeados complejos del dominio del problema o del algoritmo matemático utilizado para su resolución. Por supuesto el teorema no Free Lunch (NFL) (Wolpert & Macready, 1997), implica que un MOEA no es una técnica de solución robusta universal para todos los problemas de optimización multiobjetivo, sin embargo generalmente puede ser fácilmente guiado por la información del dominio del problema, sin tener que modificar el modelo para ser utilizado con esta técnica. Entonces el proceso de desarrollo, comprensión y pruebas es más sencillo, puesto que se utiliza de forma nativa para una aplicación dada.

Obtener el frente de Pareto exacto de cualquier problema planteado, es usualmente una tarea complicada, los MOEA's por definición intentan conseguirlo, en un tiempo computacional aceptable dentro de un rango implícito o explícito de error medible.

Los diferentes métodos evolutivos multiobjetivo se pueden clasificar en técnicas de primera y segunda generación. Las de primera generación abarca desde los que no consideran conceptos de Pareto hasta los primeros que lo consideraban, pero no tenían mecanismos de preservación de buenas soluciones durante el proceso de evolución, es decir carecían de elitismo. La segunda generación está caracterizada por los algoritmos basados en Pareto e incorporan mecanismos de elitismo (Coello, Lamont, & Van Veldhuizen, 2007).

Los primeros enfoques evolutivos para lidiar con múltiples objetivos se basan en la idea de combinar un algoritmo genético simple con métodos de escalarización de la función objetivo, entonces se encargaban de optimizar la función agregada en vez de optimizar la función multiobjetivo original (Coello C. A., 1999), (Fonseca & Fleming, 1994), (Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, 1999).

Shcaffer presentó el primer algoritmo evolutivo que no utiliza funciones de agregación, al que llamó *Vector Evaluated Genetic Algorithm* (VEGA). Este algoritmo utiliza un procedimiento de división de una población en subpoblaciones, en las que se considera el valor de adaptación de los individuos de

acuerdo a un objetivo distinto para cada sub-población y realiza el cruzamiento entre distintas de ellas. El resultado final del proceso de selección es el promedio de cada uno de los objetivos. A pesar de haber obtenido buenos resultados con este método en el ámbito de aprendizaje automático, resultó ineficiente para explorar espacios de objetivos no convexos.

Algunos trabajos posteriores no tuvieron mucho éxito en el área de los MOEA, puesto que combinaban métodos tradicionales con algoritmos evolutivos (Fonseca & Fleming, 1994), (Van Veldhuizen & Lamont, 2000). En (Fourman, 1985), se presenta un algoritmo que utiliza un método de selección basado en ordenamiento lexicográfico, donde pares de individuos se comparan de acuerdo a un objetivo prioritario, en caso de empate se considera el segundo objetivo de mayor prioridad.

Un nuevo impulso en el campo de los MOEA se produjo en la década de 1990, con la aparición de los primeros algoritmos que consideraban de forma simultánea la optimización de objetivos múltiples utilizando el concepto de dominancia de Pareto. El *Multiobjective Genetic Algorithm* (MOGA), de Fonseca y Fleming (1993), el *Niched Pareto Genetic Algorithm* (NPGA) de Horn y Nafpliotis (1994), y el *Nondominated Sorting Genetic Algorithm* (NSGA) de Srinivas y Deb (1994), fueron los primeros MOEA basados en Pareto. Los anteriores, pertenecientes a la primera generación de algoritmos evolutivos multiobjetivo, fueron aplicados a una amplia gama de problemas multiobjetivo, donde mostraron mejores resultados que los métodos no basados en Pareto.

A pesar del éxito tenido por los MOEA's de primera generación basados en Pareto, existía un problema de implementación: una vez que conseguían una solución no dominada, se podía perder al aplicarse operadores genéticos en las sucesivas generaciones. Para evitarlo, se introdujo el concepto de *elitismo* utilizado en algoritmos de un solo objetivo, considerando la existencia de múltiples soluciones posibles. Así los MOEA's de segunda generación están caracterizados por utilizar los conceptos de Pareto junto con mecanismos de elitismo para conservar buenas soluciones. Dentro de este grupo se pueden conseguir el *Strength Pareto Evolutionary Algorithm* (SPEA), de Zitzler y Thiele (1999), el NSGA-II de Deb, Agrawal, Pratab y Meyarivan (2000), y el *Controlled Elitist NSAGA-II* (CNSGA-II) de Deb y Goel (2001), entre otros.

www.bdigital.ula.ve

### **2.2.1 Técnicas de algoritmos evolutivos multiobjetivo**

A continuación una descripción de algunos MOEA según von Lücken M., (2003).

#### **2.2.1.1 Nondominated Sorting Genetic Algorithm (NSGA)**

También conocido como NSGA por sus siglas en inglés (Srinivas & Deb, 1994), es un algoritmo evolutivo de primera generación, difiere del algoritmo genético simple en la selección, específicamente en la asignación de los valores de adaptabilidad. Los operadores de cruzamiento y mutación son iguales al *Multiobjective Genetic Algorithm* (MOGA) (Fonseca & Fleming, 1994), el cual es

una adaptación de un algoritmo genético simple, solo que modificado en la etapa de selección, de tal forma que pueda trabajar con múltiples objetivos. Tanto el NSGA como el MOGA se basan en un método de selección para la clasificación enfatizando los puntos no dominados y un método para preservar la diversidad de la población. Sin embargo el NSGA utiliza un método de clasificación que no considera la cantidad de elementos que dominan a un individuo para clasificarlos, sino su nivel de dominancia. Entonces basa su método de búsqueda de un óptimo en la no dominancia de un elemento (Goldberg, 1989).

El procedimiento para asignar el valor de adaptación basado en la clasificación por dominancia (*nondominated sorting procedure*), utilizado en el NSGA se presenta a continuación:

**Procedimiento:** `fitness_assignment_nsga`

**Entrada :** Población genética  $P$

**Salida :** Población genética  $P$  con el valor de adaptación asignado a cada individuo

```
 $\forall P[i] \in P; P[i]_{dom} = falso$  //marcar todos los elementos de la población como no dominados
 $f = 0$  // hacer la cuenta de frentes igual a 0
 $df = N$  //valor inicial del dummy fitness igual al tamaño de la población  $P$ 
while  $\|P\|_c > 0$  do //mientras queden elementos sin considerar
     $f = f + 1$  //incrementar la cuenta de frentes
    for  $i = 1$  to  $\|P\|_c$  do //para cada elemento  $P[i]$  de  $P$ 
        for  $j = 1$  to  $\|P\|_c$  do //para cada elemento  $P[j]$  de  $P$ 
            if  $P[j] \succ P[i]$  then //si el elemento  $P[j]$  domina a  $P[i]$ 
                 $P[i]_{dom} = true$  //marcar elemento  $P[i]$  como dominado
                break //salir del ciclo
            end if
        end for
    end for
     $\mathcal{F}^f = \{P[i] \mid P[i]_{dom} = falso\}$  //los elementos de  $P$  no marcados conforman el frente  $f$ 
     $\forall \mathcal{F}[i]^f \in \mathcal{F}^f; \mathcal{F}[i]^f_{fitness} = df$  //asignar a cada elemento del frente  $f$  un fitness grande
    fitness_sharing_procedure( $\mathcal{F}^f$ ) //aplicar sharing a los elementos del frente  $\mathcal{F}^f$ 
     $df = \min\{\mathcal{F}[i]^f_{fitness} \mid \mathcal{F}[i]^f \in \mathcal{F}^f\}$  //guardar el peor valor de fitness del frente  $f$  en  $df$ 
     $df = df - \epsilon$  //reducir  $df$  en  $\epsilon$  para usarlo como dummy fitness del siguiente frente
     $P = P \setminus \mathcal{F}^f$  //extraer de  $P$  el frente recientemente identificado reasignando los índices
     $\forall P[i] \in P; P[i]_{dom} = falso$  // marcar todos los elementos de la población como no dominados
end while
 $P = \cup_{i=1}^f \mathcal{F}^i$  //recomponer la población  $P$  con los elementos de todos los frentes
```

**Figura 1:** Asignación valor de adaptación basado en la clasificación por dominancia (*fitness assignment nsga*) (Von Lücken M., 2003)

En el procedimiento mostrado en la Figura 1 se identifican en primera instancia los elementos no dominados de la población en estudio, los cuales constituyen el primer frente no dominado de la población. Se marcan todos ellos como no dominados en la primera iteración. Luego se toma un individuo y se compara su dominancia con cada uno de los demás, hasta encontrar uno que lo domine. Una vez ocurrido esto, se marca este elemento como dominado, se escoge el siguiente elemento y se repite el procedimiento hasta agotar la

totalidad de la población, entonces se han marcado todos los individuos dominados de esta población. Aquellos que no han sido marcados, son los elementos no dominados presentes, conformando el primer frente de individuos no dominados. A todos los elementos del frente identificado se les asigna un valor de adaptación, inicialmente es igual al tamaño de la población, el cual es un valor ficticio que provee de igual condición de reproducción a todos ellos. Para mantener la diversidad, el valor de adaptación asignado a cada individuo del primer frente se comparte entre los elementos que lo componen, dividiéndolo por la cuenta del nicho en el frente. Esto se implementa mediante el uso del algoritmo mostrado en la Figura 2:

**fitness\_sharing\_procedure**

**Entrada:** conjunto de individuos  $\mathcal{F}^f$  del  $f$ -ésimo frente nodominado, radio de nicho  $\sigma_{share}$ .

**Salida :** conjunto de individuos  $\mathcal{F}^f$  con *fitness* compartido asignado a cada elemento.

```

for  $i = 1$  to  $|\mathcal{F}^f|_c$  do                                //desde  $i = 1$  hasta el número de elementos en el frente  $f$ 
  for  $j = 1$  to  $|\mathcal{F}^f|_c$  do                                //desde  $j = 1$  hasta el número de elementos en el frente  $f$ 
     $d_{ij} = \|\mathcal{F}^f[i]_{objs} - \mathcal{F}^f[j]_{objs}\|$                 // $d_{ij}$  igual a dist. Euclid. entre objs. de  $\mathcal{F}^f[i]$  y  $\mathcal{F}^f[j]$ 
    if  $d_{ij} \leq \sigma_{share}$  then                          // calcular valor de sharing
       $sh = 1 - (\frac{d_{ij}}{\sigma_{share}})^2$ 
    else
       $sh = 0$ 
    end if
     $\mathcal{F}^f[i]_{ncount} = \mathcal{F}^f[i]_{ncount} + sh$                 // calcular la cuenta de nicho para  $\mathcal{F}^f[i]$ 
  end for
   $\mathcal{F}^f[i]_{fitness} = \frac{\mathcal{F}^f[i]_{fitness}}{\mathcal{F}^f[i]_{ncount}}$           //calcular el fitness compartido
end for

```

Figura 2: Asignación del valor de adaptación compartido (*fitness sharing*) (Von Lücken M., 2003)

Una vez finalizado todo esto, se obtiene el peor valor de adaptación entre los individuos del primer frente. Un valor ligeramente inferior a este será

asignado para los elementos del siguiente frente considerado. Luego los individuos del primer frente son eliminados temporalmente de la población. El procedimiento de clasificación se repite para los elementos de la población sin los del primer frente. Para ello, la población ahora se vuelve a marcar como no dominado para identificar un nuevo frente. Esto continúa hasta que todos los elementos tengan su correspondiente valor de adaptabilidad.

Finalizado el procedimiento de asignación de adaptabilidad, la población es sometida al operador de selección utilizando el método de la ruleta ponderada y a los demás operadores genéticos. Puesto que los elementos del primer frente tienen valores de adaptación mejores, estos siempre obtienen más copias que el resto de la población. Así este método dirige la búsqueda a las regiones de no dominancia, lo que al final conduce al frente óptimo de Pareto.

#### **2.2.1.2 Strength Pareto Evolutionary Algorithm (SPEA)**

En (Zitzler & Thiele, 1999) se presenta un nuevo enfoque evolutivo para la optimización multiobjetivo. El *Strength Pareto Evolutionary Algorithm* (SPEA) pertenece al grupo de técnicas de segunda generación y difiere de las de primera generación porque utiliza dos poblaciones, incorporando el concepto de elitismo a través del almacenamiento de las soluciones no dominadas en una población externa. El cálculo del valor de adaptación se realiza utilizando un procedimiento basado en la asignación de un valor denominado *strength* a todos los elementos de la población externa. Esto induce la formación de nichos

a partir del concepto de dominancia de Pareto llamado *strength niching*. Puesto que el conjunto de soluciones en la población externa puede ser grande, también se utiliza un procedimiento llamado *clustering*, para reducir el número de soluciones de dicho conjunto. Un pseudocódigo del proceso principal de SPEA es el siguiente:

- Paso 1** Generar una población inicial  $P$  y crear el conjunto nodominado externo  $P' = \emptyset$ .
- Paso 2** Copiar los miembros nodominados de  $P$  a  $P'$ .
- Paso 3** Eliminar las soluciones en  $P'$  cubiertas por cualquier otro miembro de  $P'$ .
- Paso 4** Si el número de soluciones en el almacenamiento externo excede un máximo  $N'$ , reducir  $P'$  por medio de *clustering*.
- Paso 5** Calcular el *fitness* de cada individuo en  $P$ , así como en  $P'$ .
- Paso 6** Seleccionar individuos de  $P(t) + P'(t)$  (unión multiconjunto), hasta que el pool de apareamiento  $P(t+1)$  se llene.
- Paso 7** Aplicar los operadores de mutación y cruzamiento específicos del problema como es usual.
- Paso 6**  $t = t + 1$
- Paso 8** Si se alcanza el máximo número de generaciones parar, sino ir al Paso 2.

Figura 3: Pseudocódigo del algoritmo SPEA (Von Lüken M., 2003)

Inicialmente se crea una población genética y un conjunto vacío de soluciones no dominadas, luego estas soluciones de la población inicial son copiadas al conjunto externo, eliminando las soluciones cubiertas por cualquier otro miembro de dicho conjunto.

#### 2.2.1.2.1 Soluciones cubiertas por una población

Dada una población  $P$  de tamaño  $N$ , para cualquier par de individuos en la población  $P[i]$  y  $P[j]$ , donde  $i, j \in \{1, \dots, N\}$ , se dice que  $P[i]$  cubre a  $P[j]$  si y solo si  $i \neq j$  y  $P[i] \succcurlyeq P[j]$ . Es decir, si dados dos individuos en una población,  $P[i]$  es mejor que  $P[j]$  si  $P[i] \succcurlyeq P[j]$ .

En algunos problemas, el conjunto de Pareto puede ser extremadamente grande (incluso infinito). Si el conjunto de soluciones no dominadas encontradas es extremadamente grande, no resulta muy útil para el DM recibirlas todas. Incluso el tamaño del conjunto externo no dominado influencia el comportamiento del SPEA, debido a que participa en la selección demasiadas soluciones podría reducir el rendimiento del proceso en general, así mismo el *strength niching* se basa en una disposición uniforme del conjunto externo, de no serlo el proceso de asignación del *fitness* posiblemente tenderá hacia ciertas regiones del espacio de búsqueda, desbalanceando la distribución general de la población. Por lo tanto si el número de elementos en el conjunto no dominado supera el máximo dado, se procede a reducir el conjunto utilizando un procedimiento de *clustering*.

Luego es calculado el valor de *fitness* para cada individuo del conjunto externo de no dominados. En primer lugar se le asigna un valor real entre [0,1], el cual indica la utilidad de un individuo o su *strength*. Este valor está directamente relacionado con la cantidad de elementos en la población para los cuales es mejor o igual que él. El cálculo de este valor para un individuo de la población externa es:

$$P'[i]_{strength} = \frac{|\{j | j \in \{1, \dots, N\} \wedge P'_c[i] \geq P[j]\}|_c}{N + 1}$$

El valor de adaptación del individuo  $P'[i]$  será igual a la inversa de su valor de *strength*, esto es:

$$P'[i]_{fitness} = \frac{1}{P'[i]_{strength}}$$

Luego de calcular el valor de adaptación de los individuos en la población externa, se calcula el valor de adaptación de los individuos de la población genética  $P$ . El *Strength* de un individuo  $P[j] \in P$  se calcula a partir de los *strength* de todas las soluciones no dominadas de  $P'[i] \in P'$  que lo dominen, esto es:

$$P[j]_{fitness} = \frac{1}{P[j]_{strength}}$$

Como resultado los individuos del conjunto externo, el cual es menor que el conjunto  $P$  reciben mayores valores de *fitness* que los demás miembros de la población. Los individuos que tienen muchos vecinos en el nicho son penalizados debido al alto valor de *strength* de los puntos no dominados, este procedimiento se conoce como *strength niching*. La idea de este mecanismo es preferir siempre los individuos más cercanos al frente de Pareto óptimo y al mismo tiempo distribuirlos en toda la superficie factible. La principal diferencia de este método con respecto al *fitness sharing* es que el nicho no está definido en términos de la distancia, sino de acuerdo a la dominancia de Pareto y no requiere el establecimiento de ningún parámetro predefinido. Esto es más adecuado puesto que en muchos problemas del mundo real la distancia no tiene un significado práctico, ya que cada objetivo puede estar expresado en magnitudes no comparables y totalmente diferentes.

Luego de asignado los valores de adaptabilidad, se procede a la selección de los individuos para el cruce. Esta selección se realiza entre los elementos de la población y el conjunto externo. El proceso de selección puede llevarse a cabo mediante un procedimiento de selección binaria (Zitzler & Thiele, 1999), luego se aplican los operadores genéticos usuales hasta llegar al máximo de generaciones definido.

### **2.2.1.3 Nondominated Sorting Genetic Algorithm II (NSGA-II)**

El *Nondominated Sorting Genetic Algorithm II (NSGA-II)* es un MOEA de segunda generación, basado en el NSGA original, utiliza como principio de clasificación la no dominancia para asignar el valor de adaptabilidad a los elementos de la población genética. Surgió como una versión mejorada del algoritmo NSGA (Srinivas & Deb, 1994), de quién heredó su estructura principal, pero incluyendo características distintivas para resolver tres aspectos fuertemente criticados en la comunidad de investigadores sobre el NSGA: el ordenamiento no dominado, la ausencia de elitismo y la dependencia del parámetro para aplicar la técnica de *sharing*. El ordenamiento no-dominado elitista mediante una técnica de comparación que utiliza una subpoblación auxiliar, que le permite disminuir la complejidad de los chequeos de dominancia de  $O(MP^3)$  a  $O(MP^2)$ , siendo M el número de funciones objetivo y P el tamaño de la población utilizada. La utilización de una técnica de *crowding* (densidad de soluciones dentro del espacio objetivo, esta técnica mide la distancia entre

soluciones para identificar las más cercanas al objetivo que se persigue (Rudenko, Schoenauer, Bosio, & Fontana, 2001)), que no requiere especificar parámetros adicionales para la preservación de diversidad en la población, eliminando la dependencia de parámetros como el de *sharing* utilizado por el NSGA original

En el siguiente algoritmo se presenta el procedimiento rápido de clasificación por no dominancia sobre el cual funciona el NSGA-II:

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

```

Procedimiento :fast_nondominated_sort
Entrada :Una población  $P$ 
Salida : Una lista de los frentes nodominados  $\mathcal{F}$ 
for each  $P[i] \in P$  do
  for each  $P[j] \in P$  do
    if  $P[i] \succ P[j]$  then //si  $P[i]$  domina al elemento  $P[j]$ 
       $S_i = S_i \cup P[j]$  //incluir  $P[j]$  en  $S_i$ 
    else if  $P[j] \succ P[i]$  then //si  $P[i]$  es dominado por  $P[j]$ 
       $nd_i = nd_i + 1$  //incrementar  $nd_i$ 
    end if
  end for
  if  $nd_i = 0$  then //ninguna solución domina a  $P[i]$ 
     $\mathcal{F}^1 = \mathcal{F}^1 \cup \{P[i]\}$  //  $P[i]$  pertenece al primer frente
  end if
end for
 $f = 1$  //el contador de frentes se hace igual a 1
while  $\mathcal{F}^f \neq \emptyset$  do
   $\mathcal{H} = \emptyset$ 
  for each  $P[i] \in \mathcal{F}^f$  do //para cada miembro  $P[i]$  en  $\mathcal{F}^f$ 
    for each  $P[j] \in S_i$  do //modificar cada miembro del conjunto  $S_i$ 
       $nd_j = nd_j - 1$  //decrementar  $nd_j$  en uno
      if  $nd_j = 0$  then //si  $nd_j$  es cero
         $\mathcal{H} = \mathcal{H} \cup \{P[j]\}$  //  $P[j]$  es dominado sólo por elementos de  $\mathcal{F}^f$ 
      end if
    end for
  end for
   $f = f + 1$  //se incrementa el contador de frentes
   $\mathcal{F}^f = \mathcal{H}$  //el frente actual está formado con los elementos de  $\mathcal{H}$ 
end while

```

Figura 4: Procedimiento de clasificación por no dominancia (*fast nondominated sort*) del NSGA-II (Von Lücken M., 2003)

Mediante el algoritmo anterior se determinan para cada una de las soluciones  $P[i]$  de una población a clasificar:

- El conjunto  $S_i$  de las soluciones dominadas por  $P[i]$
- El número  $nd_i$  de soluciones que dominan a  $P[i]$

Se comparan entre sí con respecto a la dominancia todos los miembros de la población a clasificar: Si un elemento  $P[i]$  domina a un elemento  $P[j]$ , este último se agrega a un conjunto  $S_i$ . En caso contrario si  $P[j]$  domina a  $P[i]$  entonces se incrementa el valor del contador  $nd_i$  de soluciones que dominan a  $P[i]$ . Cabe destacar que un elemento  $P[i]$  puede pertenecer simultáneamente a  $nd_i$  conjuntos de soluciones dominadas.

Una vez identificados los conjuntos de soluciones dominadas y aquellas que la dominan para cada elemento de la población, se forma el primer frente de soluciones no dominadas ( $\mathcal{F}^1$ ), con todos los elementos cuyos individuos que lo dominan es 0. El algoritmo prosigue recorriendo para cada elemento de  $P[j] \in \mathcal{F}^1$  su respectivo conjunto de soluciones dominadas de  $S_i$ , reduciendo para cada elemento  $P[j] \in S_i$  el valor de  $nd_i$  que le corresponde.

Cuando el valor de  $nd_j$  se hace cero, se agrega  $P[j]$  a una lista  $\mathcal{H}$  inicialmente vacía. Cuando se han recorrido todos los elementos del primer frente, en  $\mathcal{H}$  quedan los elementos que sólo son dominados por los elementos del primer frente, es decir, aquellos del segundo frente  $\mathcal{F}^2$ , luego se consideran estos últimos, repitiendo el procedimiento para cada uno de éste nuevo frente. El procedimiento finaliza cuando no quedan individuos cuyos elementos que lo dominan se haga cero, es cuando todos los frentes han sido identificados y  $\mathcal{H} = 0$ .

Uno de los procedimientos incluidos en el NSGA-II, es la estimación de la densidad de soluciones alrededor de una de ellas en particular con respecto a los demás elementos del frente de Pareto definido. Se calcula un valor distancia que sirve como un estimador del tamaño del cuboide más grande que encierra la solución sin incluir ningún otro punto de la población (a esto se le llama *crowding distance*), esto se consigue determinando el mayor y menor elemento inmediato de cada individuo y objetivo. Esto se consigue ordenando de menor a mayor los elementos del frente para cada uno de los objetivos considerados.

A continuación se muestra un pseudocódigo de la función de estimación de densidad:

```

Procedimiento: crowding_distance_assignment
Entrada: un conjunto de soluciones  $\mathcal{F}^f$ 
Salida : el conjunto  $\mathcal{F}^f$  con las distancia de crowding de sus elementos calculada
 $l = \|\mathcal{F}^f\|_c$ 
Para cada  $i$ ,  $\mathcal{F}^f[i]_{distance} = 0$  //inicializar a cero la distancia de todos los elementos de  $\mathcal{F}^f$ 
for  $j = 1$  to  $k$  do //para cada objetivo considerado
  Ordenar  $\mathcal{F}^f$  de acuerdo al objetivo  $j$ 
   $\mathcal{F}^f[1]_{distance} = \mathcal{F}^f[l]_{distance} = \infty$ 
  for  $i = 2$  to  $(l - 1)$  do
     $\mathcal{F}^f[i]_{distance} = \mathcal{F}^f[i]_{distance} + (\mathcal{F}^f[i + 1]_{objetivo[j]} - \mathcal{F}^f[i - 1]_{objetivo[j]})$ 
    // El sub-índice  $objetivo[j]$  representa al valor del objetivo  $j$ 
  end for
end for

```

Figura 5: Función de estimación de densidad (*crowding distance assignment*) (Von Lücken M., 2003)

A continuación se presenta el cuerpo principal del algoritmo NSGA-II:

```

t = 0
Generar una población  $P(t)$  de tamaño  $N$  en forma aleatoria
Utilizar fast_non_dominated_sort para obtener una lista  $F$  con los frentes de  $P(t)$ 
Asignar a cada elemento de  $P(t)$  un valor de inadaptabilidad igual a su nivel de nodominancia
Utilizar torneo binario para seleccionar elementos de  $P(t)$  de acuerdo con su inadaptabilidad
Efectuar cruzamiento y mutación para producir una población hijo  $Q(t)$  de tamaño  $N$ 
while El criterio de parada no se cumpla do
    f = 1 //hacer la cuenta de frentes igual a 1
     $R(t) = P(t) \cup Q(t)$  //combinar la población padre e hijo
     $\mathcal{F} = \text{fast\_non\_dominated\_sort}(R(t))$ 
    while  $\|P(t+1)\|_c < N$  do //Hasta llenar la población padre
        crowding_distance_assignment( $\mathcal{F}^f$ ) //Calcular la distancia de crowding en  $\mathcal{F}^f$ 
         $P(t+1) = P(t+1) \cup \mathcal{F}^f$  //incluir el  $f$ -ésimo frente en la población padre
        f = f + 1
    end while
    Ordenar de forma descendente de acuerdo al operador  $\geq_n$ 
    Tomar los primeros  $N$  elementos de  $P(t+1)$ 
    Seleccionar individuos de  $P(t+1)$  utilizando torneo binario de acuerdo al operador de crowding
    Aplicar cruzamiento y mutación sobre los individuos seleccionados para obtener  $Q(t+1)$ 
    t = t + 1
end while

```

Figura 6: Pseudocódigo del algoritmo NSGA-II

Inicialmente se genera una población padre de tamaño  $N$ , la cual es clasificada según su no dominancia. Luego se asigna a cada solución un valor de inadaptabilidad igual a su no dominancia. Mediante el uso de los operadores de selección, cruzamiento y mutación, se genera una nueva población de tamaño  $N$ . Luego se realizan sucesivas generaciones hasta cumplir con la condición de parada.

El elitismo se asegura haciendo cruzamientos sucesivos y comparaciones de la población padre y la hija. Esto genera una población  $R(t)$  resultado de tamaño  $2N$  clasificada según su no dominancia utilizando el

algoritmo rápido de ordenamiento mostrado en la Figura 4, identificando así los diferentes frentes no dominantes existentes. Obtenidos los frentes mencionados, la nueva población padre se forma agregando las soluciones del primer frente, continuando con los demás frentes hasta que el tamaño resultante sea menor o igual que  $N$ . La nueva población se ordena con el operador de *crowding* definido anteriormente, y se forma la nueva población con los  $N$  primeros elementos. Los individuos de cada frente se utilizan para calcular la distancia entre las soluciones vecinas. Las soluciones del último frente aceptado es ordenado de acuerdo a un criterio de comparación de *crowding* y se toman los elementos de este frente hasta completar el total de  $N$  soluciones en  $P$ . Una vez hecho esto, los elementos obtenidos se utilizan para generar una nueva población mediante los operadores de selección, cruzamiento y mutación. Esto se repite hasta que el criterio de parada o el máximo de generaciones sea alcanzado.

### **2.3 Heurísticas**

El conocimiento previo de la dificultad de un problema puede ser bastante útil para evitar perder el tiempo realizando la implementación de un algoritmo solución que, probablemente, no exista. Sin embargo esto no significa la desaparición del problema, se debe conseguir una solución sin importar su dificultad. Aquí es donde intervienen la heurística y las aproximaciones o *algoritmos aproximados*. (Canales, 2004)

Los *algoritmos heurísticos* o sencillamente *heurísticas* se refieren a un procedimiento que puede producir una buena o incluso una óptima solución para el problema planteado, sin embargo no siempre ocurre esto. También puede resultar en una solución no tan buena o sencillamente no producir solución factible alguna, ya sea mediante una heurística probabilística o determinística.

El término *algoritmo aproximado* se refiere al procedimiento que siempre proporciona una solución, aun cuando esta no sea óptima. Para que este algoritmo sea útil, es necesario calcular una diferencia o un cociente entre una solución óptima tipo y la producida por el algoritmo, con el fin de medir la exactitud de las soluciones halladas por éste último.

### 2.3.1 Algoritmos exactos vs heurísticas

Para comprender las diferencias entre ambos conceptos, se plantea un problema de optimización genérico:

Sea el problema de optimización combinatoria  $\pi$  y sea  $S_\pi(I)$  el conjunto de soluciones factibles asociado a un ejemplo  $I \in D_\pi$ . Se introduce al valor de la función objetivo  $m_k(I, \sigma)$  asociado a la solución  $\sigma \in S_\pi(I)$ . La solución óptima  $\sigma^*$  debe verificar, (en un problema de mínimo):

$$m_\pi(I, \sigma^*) \leq m_\pi(I, \sigma) \forall \sigma \in S_\pi(I)$$

Al valor  $m_\pi(I, \sigma^*)$  se le denomina  $OPT_\pi(I)$ .

Un algoritmo  $A$  se dice que es una heurística o un algoritmo aproximado para el problema de optimización combinatoria  $\pi$  cuando  $\forall I \in D_\pi$  Encuentra en tiempo razonable una solución factible  $\sigma \in S_\pi(I)$ . Se verifica entonces  $A(I) = m_\pi(I, \sigma) \geq OPT_\pi(I)$ .

La esencia está en la manera de aproximación de  $A(I)$  al valor  $OPT_\pi(I)$ . Si el problema de decisión asociado a  $\pi$  es de la clase  $P$ , el algoritmo polinomial correspondiente sería una heurística verificando  $A(I) = OPT_\pi(I)$  para cualquier ejemplo de  $I \in D_\pi$ . Sin embargo, cuando el problema sea de clase  $NP$ , el algoritmo no determinístico polinomial no puede ser utilizado como algoritmo, en este caso hay dos opciones básicas:

- **Algoritmos exactos:** Diseñar un algoritmo de búsqueda exacto que recorra todas las soluciones (enumeración explícita), o parte de ellas (enumeración implícita). Este enfoque sólo vale para problemas de tamaño reducido.
- **Algoritmos aproximados y/o heurísticas:** Buscan una buena solución aunque no sea la óptima en un tiempo reducido y no pueden asegurar la optimalidad, al no hacer un recorrido exhaustivo por todas las soluciones.

Entonces es deseable poder medir el grado de exactitud de este tipo de algoritmos, midiendo el grado de aproximación de las soluciones producidas. Se identifican así las medidas *a priori* y *a posteriori*. (Canales, 2004)

### 2.3.1.1 Medidas de aproximación a priori

Una formulación un poco más formal de las medidas a priori se define en función de los siguientes conceptos:

**Cociente**  $R_A(I)$ : Se define este cociente a partir del problema de minimización  $\pi$  y de un algoritmo  $A$ :

$$R_A = \frac{A(I)}{OPT_{\pi}(I)}$$

En el caso de maximización se define de forma inversa, entonces en cualquier caso se verifica  $R_A(I) \geq 1$ . Al depender del ejemplo  $I \in D_{\pi}$ , sólo es válido cuando se conozca el óptimo, por lo que limita fuertemente su validez. Una generalización se logra eliminando la dependencia del ejemplo proporcionado y la limitación anterior.

**Cociente de comportamiento absoluto**  $R_A(I)$ : Está definido según:

$$R_A = \text{Inf}\{r \geq 1 / R_A(I) \leq r \forall I \in D_{\pi}\}$$

Es muy difícil identificar este índice para una heurística en particular que resuelve un problema de optimización, puesto que se debe acotar la función objetivo de cualquier ejemplo.

### 2.3.1.2 Medidas a posteriori

Generalmente son más precisas que las anteriores y están basados en resultados obtenidos para distintas instancias del problema planteado. Su dificultad radica en el conocimiento de la solución exacta del problema.

Las heurísticas pueden evaluarse empíricamente tomando como punto de partida ejemplos aportados por la comunidad científica, esto es denominado *análisis medio* en contraposición al análisis del peor de los casos que caracteriza los índices a priori (Canales, 2004) .

### 2.3.2 Algoritmos iterativos

Su funcionamiento se basa en tomar una solución factible y comenzar a iterar sobre ella para tratar de obtener una mejor, esta operación se repite hasta cumplir algún criterio de parada definido con anterioridad.

Un ejemplo genérico de este tipo de algoritmos es el siguiente: Sea  $x^1 \in X$  la solución factible inicial. A partir de  $x^k \in X$ , se construye otra  $x^{k+1} = h(x^k) \in X$  verificando  $f(x^{k+1}) \leq f(x^k)$  donde  $f()$  es la función objetivo del problema de minimización. La elección de la función  $h()$  y el valor inicial  $x^1$  caracterizan el método iterativo.

Un caso particular de métodos iterativos son los *algoritmos vecinales*. Para cada solución factible  $x \in X$  se define el conjunto de soluciones vecinas

$V(x)$  y a partir de una distancia  $\lambda$  se limita este conjunto  $V_\lambda(x) = \{y \in X / d(x, y) \leq \lambda\}$

## **2.4 Metaheurísticas**

El término metaheurística fue acuñado por Glover (1986), al introducir la búsqueda tabú. Consiste en una estrategia que controla un grupo de heurísticas y/o algoritmos aproximados. En su diseño existen un conjunto de parámetros cuyos valores determinan el conjunto de heurísticas asociadas. Desde un punto de vista operativo, una metaheurística es un procedimiento parametrizado, su codificación permite aplicar con un mínimo esfuerzo computacional gran cantidad de heurísticas. (Canales, 2004).

El auge de los procedimientos metaheurísticos radica en la necesidad de resolución rápida de problemas reales. Cabe destacar que los algoritmos heurísticos por si solos no garantizan una solución óptima, aunque éste es su propósito. Sin embargo el uso de metaheurísticas permite llegar a soluciones mucho más exactas en un menor tiempo.

Dentro de las distintas técnicas heurísticas se pueden mencionar: Métodos constructivos, de descomposición, de reducción, de manipulación del modelo y de búsqueda local. Tradicionalmente se diseñaba un algoritmo específico que encajara en alguna de estas clases, sin embargo esto se fue transformando en el diseño de métodos generales que sirvan para resolver

clases o categorías de problemas. Dado que estos métodos generales sirven para construir o guiar el diseño de métodos que resuelvan problemas específicos se les ha dado el nombre de Metaheurísticas (Jaziri, 2008). Una definición más formal introducida por Osman y Kelly en 1995 se muestra a continuación:

*“Los procedimientos metaheurísticos son una clase de métodos aproximados que están diseñados para resolver problemas difíciles de optimización combinatoria, en los que las heurísticas clásicas no son ni efectivos ni eficientes. Las metaheurísticas proporcionan un marco general para crear nuevos algoritmos híbridos combinando diferentes conceptos derivados de: inteligencia artificial, evolución biológica y mecanismos estadísticos”.*

A continuación se describirán algunos de los algoritmos metaheurísticos más utilizados en la optimización multiobjetivo.

#### **2.4.1 Algoritmos Genéticos (GA)**

Son métodos adaptativos utilizados en problemas de búsqueda y, como en este caso, de optimización basados en los principios naturales de la reproducción sexual y supervivencia del más apto.

Dado un problema específico, la entrada del Algoritmo Genético o *Genetic Algorithm* (GA por sus siglas en inglés), es un conjunto de soluciones potenciales a ese problema, codificadas de alguna manera, y una métrica llamada función de aptitud o *fitness* que permite evaluar cuantitativamente a

cada una candidata. Estas candidatas pueden ser soluciones válidas al problema, con el objetivo de que el AG las mejore gradualmente, sin embargo se suelen generar aleatoriamente.

Luego el GA evalúa cada candidata de acuerdo con la función de aptitud o *fitness*. En un conjunto de soluciones generadas aleatoriamente, la mayoría no serán válidas para el problema, y serán eliminadas. Sin embargo, aleatoriamente, unas pocas pueden ser “prometedoras”, pueden mostrar un ajuste parcial a una buena solución al problema planteado.

Estas candidatas “prometedoras” se conservan y se les permite reproducirse. Se realizan múltiples copias de ellas, con algunas imperfecciones entre estas; según cierta probabilidad definida se introducen cambios aleatorios durante el proceso de copia, llamado mutación. Luego, esta descendencia prosigue con la siguiente generación, formando una nueva población de individuos o soluciones candidatas, y son sometidas a una ronda de evaluación de aptitud. Las que han empeorado o no han mejorado con los cambios en su código son eliminadas. Sin embargo, las variaciones aleatorias introducidas en la población pueden haber mejorado algunos individuos, convirtiéndolos en mejores soluciones del problema, más completas o más eficientes. De nuevo, se seleccionan y copian estos mejores individuos hacia la siguiente generación con cruces aleatorios, y el proceso se repite. Las expectativas son que la aptitud media de la población se incrementará en cada ronda y repitiendo este proceso cientos o miles de rondas, pueden descubrirse soluciones muy buenas

del problema. Es aquí donde se evidencia el principio natural de evolución de las especies, en el cual se basa esta técnica (Marcyz, 2004).

#### **2.4.1.1 Operaciones de un algoritmo genético**

Los Algoritmos Genéticos funcionan sobre poblaciones de individuos, los cuales representan elementos del espacio de soluciones del problema que se está tratando (Gestal & Rivero, 2010).

Para poder conseguir generaciones de individuos o soluciones a partir de un par de estas, el algoritmo genético tiene que valerse de algunas operaciones de reproducción, descritos brevemente a continuación:

- Selección: Se encarga de elegir cuales individuos deben tener posibilidades de reproducción, es decir, ser los padres de la próxima generación, los cuales son los más aptos o los que tienen mejor función de aptitud, con el fin de otorgar mayores probabilidades de mejora a la siguiente generación.

- Cruce: También conocida como reproducción de tipo sexual, por cuanto es necesario al menos un par de individuos para crear una generación o conjunto de soluciones posibles, tomando como punto de partida la información genética del par de individuos iniciales.

- Copia: También conocida como reproducción asexual, consiste en pasar a la siguiente generación un grupo de individuos sin ningún tipo de variación.

•Mutación: Consiste en variar la información genética de un individuo de forma aleatoria, con el fin de agregar entropía a la población haciendo posible la aparición de alguna característica probablemente deseada o evitar estancarse en una solución local.

En la

Figura 7 se muestra el pseudocódigo de un algoritmo genético sencillo:

```
Inicio Algoritmo Genético Simple
    Generar una población inicial.
    Computar la función de evaluación de cada individuo.
    Mientras No Terminado HAGA
    Inicio Producir nueva generación
    Para Tamaño población/2 HAGA
    Inicio Ciclo Reproductivo
        Seleccionar dos individuos de la anterior
            generación, para el cruce (probabilidad de
            selección proporcional a la función de
            evaluación del individuo).
        Cruzar con cierta probabilidad los dos individuos
            obteniendo dos descendientes.
        Mutar los dos descendientes con cierta
            probabilidad.
        Computar la función de evaluación de los dos
            descendientes mutados.
        Insertar los descendientes mutados en la nueva
            generación.
    Fin
    Si la población ha convergido Entonces
        Terminado:= TRUE
    Fin
Fin
```

Figura 7: Pseudocódigo de un algoritmo genético simple

Un ejemplo de aplicación en esta área se muestra en Hoyos & Pérez, (2007), donde los autores implementan un algoritmo genético para clasificar imágenes de satélite mediante el uso de una arquitectura de Grid y la publicación de los resultados obtenidos en la web, basado en el principio de la Web 2.0 o Web Semántica. Esta implementación se hace para generar una nueva técnica de clasificación, sin la intervención de un experto, aprovechando las bondades de las arquitecturas de Grid y haciendo un aporte a la filosofía de Web inteligente mediante los resultados obtenidos en la clasificación de imágenes reportado en este estudio.

#### **2.4.2 Variable Neighbourhood Search (VNS)**

Propuesto por P. Hansen y N. Mladenovic, su idea básica consiste en realizar exploraciones sucesivas en un subconjunto de vecinos, con el fin de proveer una mejor solución. Explora vecindarios distantes de la solución actual de manera incremental, y se mueve de uno actual a otro si se ha obtenido una mejora. Su idea principal es el cambio dinámico de vecindarios dentro de un algoritmo de búsqueda local (Martínez C. , 2011).

El funcionamiento del VNS se basa en 3 principios:

- Un óptimo local dentro de un vecindario, no necesariamente lo será dentro de otro.
- Un óptimo global es un óptimo local en todos los vecindarios.

- Para muchos problemas de optimización, los óptimos locales están relativamente cerca entre ellos.

Motivado a que el óptimo local y el global tienen valores de algunas variables de decisión parecidos, es posible dirigir el proceso de búsqueda hacia la solución óptima, tomando en cuenta la tendencia de éstos valores, guiando entonces la búsqueda hacia una convergencia más rápida.

Un esquema genérico de funcionamiento del VNS tiene tres fases:

- *Agitación (Shaking)*: En esta fase se obtiene una solución de forma aleatoria, la cual servirá de punto de partida en la exploración del vecindario.
- *Búsqueda Local (Local Search)*: Se aplica un algoritmo de búsqueda local sobre la solución inicial.
- *Movimiento (Move)*: Al comparar el valor de la función objetivo de la solución obtenida en la fase de búsqueda local, con la mejor solución hasta el momento. Si existe una mejora, se mueve como un nuevo mejor y se comienza el proceso de búsqueda nuevamente dentro de un vecindario más pequeño. Si no hay una mejora, entonces se amplía la búsqueda hacia un vecindario mas grande.

Los algoritmos basados en VNS deben incluir varias estructuras de vecindarios, con el fin de alcanzar soluciones cercanas al óptimo global. Una forma de realizar movimientos dentro del espacio de soluciones es cambiando algunos elementos que conforman la solución. Estos movimientos permiten obtener diferentes vecindarios si se parametrizan los aspectos relacionados con los criterios de cambio, por ejemplo fijando el número máximo de elementos que se pueden cambiar dentro de la vecindad (Hansen & Mladenovic, 2003).

### **2.4.3 Hill Climbing (HC)**

Es un algoritmo basado en “ascenso” dentro del espacio de soluciones, de sencilla implementación y gran robustez. La dirección de búsqueda se realiza de forma exhaustiva calculando todas las posibles direcciones y seleccionado aquella que proporcione un mejor ascenso o descenso, según sea la naturaleza del problema. Se elige una solución  $S_n$  al azar y desde ésta se calculan los posibles caminos que pueda tomar el algoritmo, entonces selecciona su movimiento y reemplaza  $S_n$  por un nuevo elemento considerado como mejor que el anterior, entonces realiza nuevamente la evaluación con el fin de conseguir un óptimo local en base a estos movimientos iterativos. El *Hill Climbing* (HC) se detiene cuando ninguna dirección de búsqueda proporciona una solución que mejore a la actual, cuando la mejora es menor que un cierto valor mínimo, o cuando se alcanza un máximo de búsquedas predefinidas (Conesa, Contero, & Company, S.F).

Su nombre se debe a la similitud de un escalador que asciende a través de una colina, en donde cada paso que da (movimiento entre soluciones) lo acerca a su meta que es llegar a la cima (solución óptima). Las soluciones mejores están a mayor altitud, formando colinas y picos; las peores están a menor altitud, formando valles. Un "ascenso a la colina" es, por tanto, un algoritmo que comienza en un punto dado del espacio de soluciones y se mueve hacia arriba en busca de una mejor.

La tendencia a obtener mínimos locales es el principal problema del HC, si los movimientos son muy pequeños podría estancarse en algunas áreas del espacio de soluciones. A continuación algunos ejemplos donde pudiera estancarse el HC:

- Un máximo local, es un estado mejor que sus vecinos pero no es mejor que otros más lejanos.
- Una meseta, es un espacio de búsqueda donde todo un conjunto de estados vecinos tienen igual valor.
- Un risco, es un tipo especial de máximo local, imposible de atravesar con movimientos simples.
- Planicie: son áreas del espacio de estados en donde la función de evaluación básicamente es plana.

## Capítulo 3: Metodología

Para la realización de la investigación, se seguirá una metodología aplicada a la optimización de modelos orientados a metaheurísticas, la cual abarca desde la concepción inicial del problema, hasta su implementación y mejora. A continuación se describe la metodología utilizada en el desarrollo de metaheurísticas (Talbi, 2009):

- *Formulación del problema:* En primer lugar el problema debe ser identificado. Entonces se realiza un primer planteamiento. La formulación puede ser imprecisa pero debe hacer un esbozo de los factores internos, externos y objetivos que apoyaran la resolución del planteamiento. Para esta investigación se tomará el problema planteado por Meng (2009), donde se identifica una necesidad de optimización en el flujo vehicular en una autopista con 21 rampas de acceso (*ramp metering*). También se especifica el comportamiento de todo el sistema en términos de promedios, tasas de entrada y salida.
- *Modelado del problema:* Se utilizan como punto de partida los basamentos obtenidos en la formulación del problema, con el fin de generar un modelo matemático abstracto capaz de representar la situación planteada. Se pueden utilizar modelos similares, de tal forma que se pueda reducir a un modelo de optimización bien estudiado. Usualmente son simplificaciones de la realidad, están compuestos por

aproximaciones y, en ocasiones, pasan por alto procesos complejos para representar en un modelo matemático. En el caso de este trabajo, se toman los datos proporcionados en la descripción del problema y a partir de ellos se crea un modelo matemático que contenga las características del comportamiento del sistema vial en estudio, obteniendo entonces una abstracción de la situación descrita y adecuada al planteamiento de un problema de optimización multiobjetivo.

- *Optimización del problema:* Una vez que el problema es modelado, el procedimiento de resolución genera una “buena” solución al planteamiento, ésta puede ser óptima o sub-óptima. Es de notar que se está buscando solución a un modelo abstracto. Es por esto que la medición de las soluciones obtenidas, pueden determinar si es exacto o no en este contexto. Se pueden utilizar algoritmos similares ya implementados y probados, así como también se puede incluir conocimiento de una aplicación específica. En este paso se procede a la adecuación del modelo matemático obtenido para su implementación en un experimento numérico con un algoritmo híbrido, para esto se toman las funciones objetivo con sus restricciones, se analiza el objetivo en términos de los requerimientos de maximización de flujo vehicular del sistema total y se toman los datos necesarios para configurar el experimento numérico en función de las características del problema modelado, con el fin de reflejar toda la dinámica del sistema en estudio

dentro de la configuración, para así entonces no perder generalidad y alejarse del objeto real del problema durante su implementación numérica.

- *Implementación de la solución:* Luego de la optimización del problema, se obtienen una solución la cual es probada de forma práctica y su resultado se marca como “aceptable”, según un indicador definido previamente para realizar la comparación. Algún conocimiento práctico puede ser incluido en la implementación de la solución. Si es inaceptable, el modelo y/o el algoritmo de optimización debe ser mejorado y el proceso de decisión debe repetirse. Para este problema se realizará la optimización mediante la implementación de un algoritmo híbrido ejecutado en dos fases. Una primera fase se realiza la optimización mediante el NSGA-II, donde se obtiene un primer resultado, el cual sirve de una solución base que alimenta a un segundo algoritmo. En la segunda fase un algoritmo de búsqueda local procura una mejora sobre la base entregada por el NSGA-II, es en este paso donde se implementa una hibridación compuesta de dos metaheurísticas, colaborativa, de bajo nivel, con fuerte acoplamiento y procesamiento secuencial (Raidl, 2006), resultado de la combinación dos métodos para conseguir una solución del problema multiobjetivo de optimización combinatoria.

La metodología descrita comprende el desarrollo de algoritmos de optimización aplicado a cualquier situación, tomando como punto de partida un problema del mundo real que será objeto de investigación, planteado como una abstracción en función de las variables a estudiar en un caso particular, para luego definir un método de optimización, con el fin de obtener un grupo de soluciones posibles y luego evaluarlas con el modelo planteado, determinando así si el algoritmo utilizado es aceptable o si es necesaria una revisión del modelo o de los algoritmos empleados.

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

## Capítulo 4: Desarrollo

Siguiendo los pasos de la metodología planteada en el Capítulo 3 en se propone el estudio del trabajo realizado por Meng (2009), donde se plantea un problema de un sistema vial modelado como un MOP. Tomando como punto de partida el problema tratado por el mencionado autor, se procede de la siguiente manera según la metodología planteada por Talbi (2009):

### 4.1 Formulación del problema

Los autores en el trabajo titulado “*A Pareto-optimization approach for a fair ramp metering.*” (Meng, 2009), hacen un planteamiento sobre la optimización del flujo vehicular de una autopista con 21 rampas de acceso y 18 rampas de salida, a lo largo de un tramo de la autopista interestatal 210W ubicada en Pasadena, California. Las rampas de acceso pueden ser controladas mediante la implementación de semáforos (*ramp metering* por su término en inglés), para regular la incorporación de los vehículos a la vía principal de la autopista, con el fin de establecer una tasa de acceso suficientemente alta sin colapsar el canal de incorporación, satisfaciendo la demanda de cada rampa con una cola mínima, de tal forma que se le proporcione a los usuarios tiempos justos en cuanto espera en la rampa (minimización de la cola de espera) y transito de la autopista (maximización del

flujo en la vía principal), consiguiendo así un desempeño aceptable del sistema vial completo y usuarios de la red vial satisfechos.

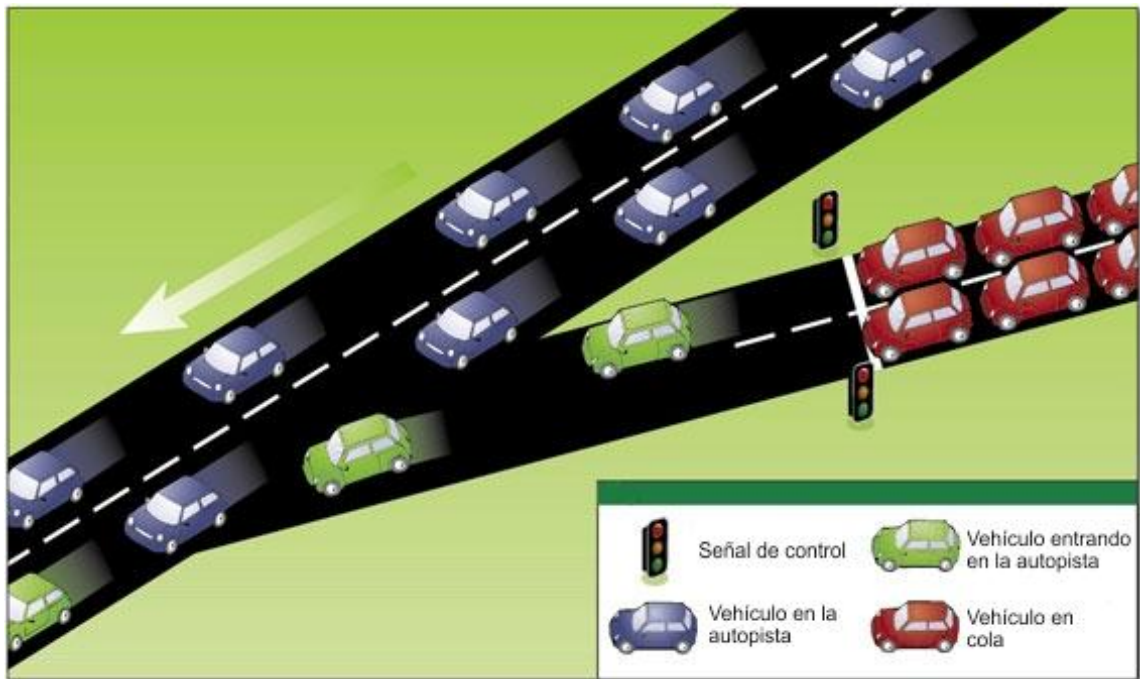
Una vez definido el ámbito del problema, se procede a identificar los elementos que permitirán realizar una abstracción del comportamiento del sistema vial completo en función de las demandas promedio de los distintos componentes en estudio. Para este caso de estudio se identificaron los siguientes elementos:

- *Rampa de acceso*: Permite a los usuarios incorporarse a la vía principal de la autopista desde carreteras de otros lugares que confluyen hacia esta vía como sistema de interconexión a otras localidades. Estas rampas poseen una demanda promedio de vehículos/hora y una capacidad máxima de vehículos/hora. Sin embargo existe una razón entre el 40% y 60% (Meng, 2009), de los vehículos que acceden a través de estas rampas desocupan la vía principal de la autopista una vez ingresan, ya sean por trasladarse a otro carril o descender por una rampa de salida.
- *Vía principal*: Es la autopista en si, donde desembocan las rampas de acceso y permiten el traslado de los conductores de una localidad a otra. Esta posee una demanda promedio en términos de vehículos/hora/carril y una capacidad máxima expresada en vehículos/hora/carril. Esta es una de las principales restricciones que

deben tener en cuenta para el acceso de los automóviles en las secciones de la autopista, puesto que la suma de las demandas promedio de las rampas de acceso superan la capacidad máxima de la autopista.

- *Rampas de descenso*: Son las que permiten salir de la autopista e incorporarse a las vías por donde el conductor alcanza su destino. Estas capacidades no se definen en el problema, sólo se describe una proporción de automóviles que abandonan el carril de incorporación una vez que ingresan a la autopista, ya sea porque descienden por una de estas rampas de salida o porque cambian a un carril interno de la autopista para proseguir su camino.

En la Figura 8 se muestra un esquema de la disposición de los elementos que conforman el *ramp metering* en la autopista. Allí se identifican gráficamente los elementos que intervienen en el sistema como son: los vehículos en tránsito dentro de la autopista, los vehículos en proceso de incorporación una vez que se les ha permitido el paso de la rampa a la vía principal y, finalmente, el grupo de vehículos que esperan la señal de control que les permita ingresar a la autopista. Todos estos elementos son deben ser considerados al momento de plantear el modelo matemático que representará esta situación.



**Figura 8: Esquema de una rampa de acceso**

Para el desarrollo de este trabajo, se tomará parte del principio de modelado planteado por Meng (2009), donde definen un esquema multiobjetivo basándose en la separación de la autopista en estudio en 8 segmentos principales, donde se consideran las rampas de acceso, las rampas de descenso y la vía principal, en la Figura 9 se muestra el esquema original mostrado en el trabajo mencionado anteriormente.

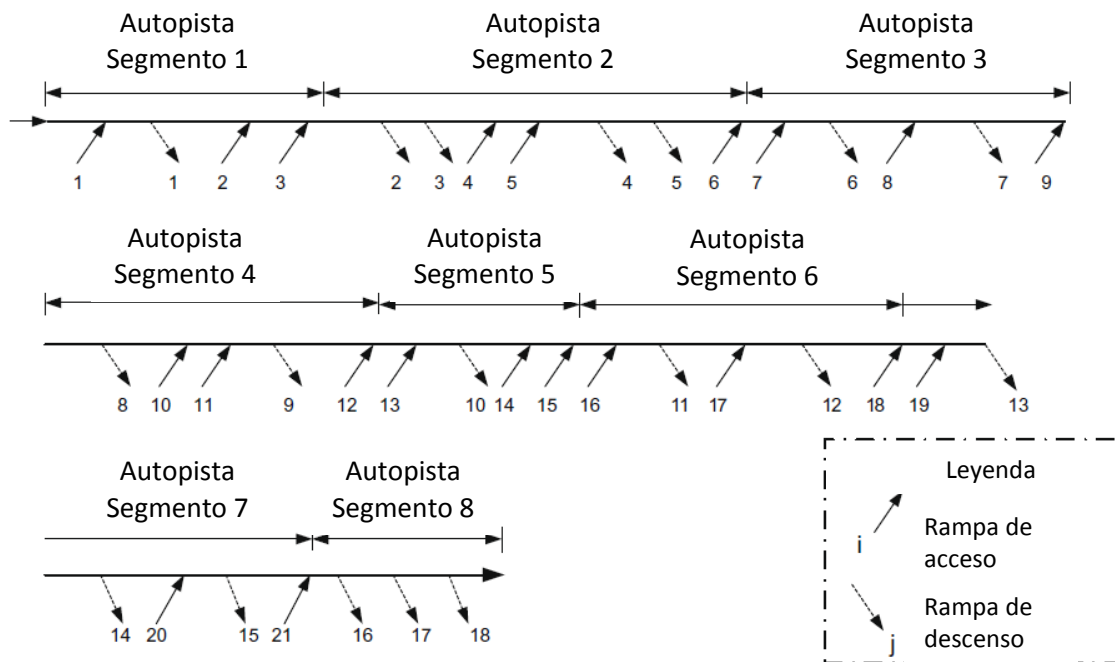


Figura 9: Estructura topológica de la autopista, separada en 8 segmentos. (Meng, 2009)

Para el modelo se considerará sólo los primeros siete segmentos de la autopista, puesto que son los que poseen rampas de acceso (*on-ramp*) y es allí donde se va a intervenir el sistema para su optimización, el octavo segmento sólo tiene rampas de descenso (*off-ramp*). El objeto de este trabajo es maximizar el flujo vehicular de la autopista en función de sus entradas, por lo que el segmento 8 no cumple con las especificaciones del problema en estudio, por lo tanto no se tomará en cuenta para el resto del trabajo desarrollado.

## 4.2 Modelado del problema

La comprensión del problema permite una visión completa que permite realizar una abstracción del comportamiento del sistema y su posterior

modelado matemático. Para el desarrollo de este estudio se utilizará un modelo multiobjetivo basado en la sección descrita de la autopista, siendo los objetivos a evaluar el desempeño de cada uno de los segmentos, tomando en cuenta las entradas y salidas de cada uno de ellos y cómo intervienen en el rendimiento total del sistema.

Para esto se definen los siguientes elementos:

$r_{ni}$ : Tasa de ingreso a la autopista a través de la rampa de acceso  $i$  en el segmento  $n$ .

$d_{ni}$ : Demanda promedio de la rampa  $i$  en el segmento  $n$ .

$s_n$ : Segmento de la autopista  $n$ .

$t_n$ : Tasa de salida del segmento  $n$ .

$q_n$ : Demanda promedio del segmento  $n$ .

En función del problema planteado y sus elementos, se construye un modelo matemático de optimización multiobjetivo.

Sea  $r_{ni}$  la tasa de flujo vehicular de la  $i$ -ésima rampa perteneciente al  $n$ -ésimo segmento de la autopista  $s_n$ , tal que  $0 < r_{ni} \leq 1$ , se toma el segmento  $s_n$  de la rampa como un elemento de recepción de los flujos de vehículos generado por cada una de las rampas  $r_i$  que desembocan en  $s_n$ , estas se

definen como un flujo entrante a la sección  $s_n$  de la autopista, por lo tanto se denota el flujo de la rampa como  $r_{ni}$ :

$$FlujoR(r_{ni}) = \sum d_{ni}r_{ni}$$

Donde  $d_{ni}$  es la demanda promedio de la rampa  $r_{ni}$ , por lo que la multiplicación de cada uno de estos factores determina la proporción de la demanda servida por  $r_{ni}$  a  $s_n$ . Según el modelo inicial (Meng, 2009), existe una porción de los vehículos que abandonan el carril de incorporación ya sea porque utilizan una rampa de descenso o se cambian a un carril interno de la autopista para seguir su camino, denotada por  $t_n$  (varía entre 0,4 y 0,6), de modo tal que la cantidad de vehículos que permanecen en la vía viene dado por  $(1 - t_n)$ , los cuales formaran parte del flujo de entrada de la próxima sección de la autopista. De tal forma que el flujo total de una sección  $s_n$  viene definido por:

$$FlujoS(s_n) = (1 - t_n) \sum d_{ni}r_{ni} - (t_n) \sum d_{ni}r_{ni} + (t_{n-1}) \sum d_{(n-1)k}r_{(n-1)k}$$

Donde  $(1 - t_n) \sum d_{ni}r_{ni}$  representa la porción de vehículos que permanecen en el carril de incorporación de la autopista en esa sección  $s_n$ , la cantidad de vehículos que abandonan el carril de incorporación de la autopista está dada por  $(t_n) \sum d_{ni}r_{ni}$  y finalmente  $(t_{n-1}) \sum d_{(n-1)k}r_{(n-1)k}$  es la cantidad de vehículos que permanecieron en el carril de incorporación desde la sección anterior de la autopista. Sin embargo, respetando el modelo inicial planteado, la suma de estos flujos está restringida a la demanda promedio de la sección  $s_n$

(1500 vehículos/hora), denotada en este caso por  $q_n$ . De modo que la restricción para el flujo de la sección se define:

$$q_n - \left( (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k} \right) \geq 0$$

Con esta restricción se acota el flujo vehicular, impidiendo que el sistema completo adopte una capacidad infinita de ingreso, lo cual pudiera desestabilizar el sistema durante los cálculos.

Una vez planteado los elementos que describen el comportamiento del sistema completo, se unifican en una primera aproximación del modelo a optimizar:

*Maximizar:*

$$F(r) = \sum_{i=1}^k d_n r_i, \text{ donde } k = 21$$

*Sujeto a:*

$$1) 0 < r_{ni} \leq 1$$

$$2) 0,4 < t_n \leq 0,6$$

$$3) d_{ni} > 0$$

$$4) (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k} \leq q_n$$

$$5) (1 - t_n) \sum d_{ni} r_{ni} > 0$$

Donde  $F(r)$  representa el flujo total presente en el canal de incorporación de la autopista,  $d_n r_i$  es la cantidad de vehículos aportados por la rampa  $i$  (donde  $d_n$  es la demanda promedio de la rampa  $i$  y  $r_i$  es la proporción de vehículos que ingresan por esta rampa en función de la demanda asociada). Sin embargo el modelo reflejó un comportamiento poco uniforme, puesto que al colocar todas las variables de decisión evolucionando sin tomar en cuenta su sección de influencia, ocasiona que si las primeras rampas aportan la mayor parte del flujo vehicular, el resto de ellas (variables en este caso), no tendrán oportunidad de maximizar su valor dentro de la función objetivo. Por lo tanto este modelo fue revisado y mejorado en una segunda versión.

En una segunda versión, se observó cierta irregularidad entre los valores de las variables de decisión (rampas), las primeras secciones de la autopista aportan la mayor proporción de la capacidad total de la autopista. Para tratar de mejorar esta situación sin perder la generalidad del problema inicial, se separaron las variables en flujos de entrada y salida para cada una de las secciones de la autopista, logrando entonces siete funciones objetivo, una por cada sección de la autopista, donde se tomaron en cuenta como los flujos de entrada las rampas de acceso asociadas a cada sección y el flujo residual de la sección de la autopista anterior (los vehículos que continúan utilizando el canal de incorporación hasta la próxima sección), como flujo de salida de la sección en estudio se tomó la proporción de vehículos que dejan el carril de

incorporación y aquellos que quedan, forman parte del flujo de entrada de la siguiente sección. El segundo modelo planteado queda como sigue:

*Maximizar:*

$$C = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \\ c_7 \end{bmatrix}$$

*Tal que:*

$$c_n = (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k}$$

*Sujeto a:* [www.bdigital.ula.ve](http://www.bdigital.ula.ve)

$$1) 0 < r_{ni} \leq 1$$

$$2) 0,4 < t_n \leq 0,6$$

$$3) d_{ni} > 0$$

$$4) (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k} \leq q_n$$

$$5) (1 - t_n) \sum d_{ni} r_{ni} > 0$$

Donde  $c_n$  representa la función objetivo del flujo vehicular a maximizar del n-ésimo segmento de la autopista, donde confluyen los vehículos provenientes de las rampas de acceso, las rampas de descenso y la porción de

ellos que permanecieron desde la rampa anterior. La restricción 1 define el rango de variación de la tasa de entrada de cada una de las rampas, la cual es la proporción de vehículos que deben pasar por ésta en un momento determinado. La restricción 2 es la variación de la tasa de salida del segmento  $n$ . La restricción 3 es una definición formal de la demanda en una rampa en particular, se define como un valor mayor que 0 a manera de darle formalismo al modelo planteado. La restricción 4 define el máximo flujo vehicular que puede mantener un segmento de la autopista, en función de sus entradas y salidas. La restricción 5 se refiere al flujo de las rampas de acceso, el cual debe ser un valor mayor que cero, para evitar que el modelo arroje valores negativos de flujo vehicular en cualquier segmento de la autopista.

www.bdigital.ula.ve

Sin embargo se observó una alta correlación entre las funciones objetivo, puesto que parte de las variables de  $c_n$  afectan a  $c_{n+1}$ , por lo cual se puede simplificar el modelo de la versión anterior, sin perder generalidad, sumando todas las funciones objetivo y modificando algunas restricciones. También se considera maximizar el flujo de cada sección, tomando en consideración el aporte que realiza cada rampa (variable de decisión), al flujo vehicular de la sección asociada, agregando entonces una función objetivo por cada sección. Se incluyen siete restricciones más tomando en cuenta el aporte de las rampas por cada sección y acotando dicho aporte a la máxima demanda descrita por el.

De esta revisión y tomando en cuenta las consideraciones descritas surge un tercer y último modelo, basado en el problema original planteado en el trabajo de Meng (2009):

*Maximizar:*

$$1) F(c) = \sum_{n=1}^v c_n$$

$$2) G(r) = \sum_{i=1}^k r_i$$

$$3) s_n = q_n(r_i + r_{i+1} + r_{i+2})$$

*Tal que:*

$$c_n = (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k}$$

*Sujeto a:*

$$1) 0 < r_{ni} \leq 1$$

$$2) 0,4 < t_n \leq 0,6$$

$$3) d_{ni} > 0$$

$$4) (1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k} \leq q_n$$

$$5) (1 - t_n) \sum d_{ni} r_{ni} > 0$$

$$6) (r_{ni} + r_{ni+1} + r_{ni+2}) \leq \frac{q_n}{d_n}$$

Una vez construido el modelo final, se definen los elementos que lo componen, muchos de ellos ya han sido mencionados. Sin embargo se revisan nuevamente a fin aclarar cualquier modificación realizada. En el modelo final  $s_n$  define de manera general el flujo total de las rampas del segmento  $n$  de la autopista, por lo tanto se tienen una función objetivo para cada segmento, en total siete de este tipo para un total de nueve funciones objetivo. En cuanto a las restricciones se agregan siete más (definida de manera general por la restricción siete del modelo), que contemplan una regulación entre el flujo aportado por cada rampa en relación con la demanda total del segmento de la autopista, lo cual proporciona un crecimiento mas uniforme en las variables de decisión.

www.bdigital.ula.ve

### **4.3 Optimización del problema**

Siguiendo el problema planteado inicialmente por Meng (2009), se requiere optimizar el flujo vehicular total en la autopista, con el fin de minimizar la cola en las rampas de acceso y maximizar el volumen del flujo vehicular a través de la vía principal de la autopista, obteniendo así un régimen de tiempo de espera y tiempo de uso más equitativo para los usuarios de la vía principal. Esto se puede lograr incrementando el flujo vehicular en las rampas de acceso, sin colapsar la vía principal de la autopista, de tal forma que la cantidad de vehículos que circule sea máxima.

Para probar el modelo planteado, se configura un experimento numérico basado en el planteamiento y en el último modelo matemático mostrado, se realizan algunas simplificaciones desde el punto de vista algebraico para su implementación experimental. Una vez expandidas las ecuaciones del modelo general en función del caso particular en estudio, algunos términos se simplifican, por lo cual en este paso de la metodología propuesta por Talbi (2009), se realizan los pasos necesarios para adecuar el modelo a las implementaciones numéricas, de acuerdo al algoritmo híbrido diseñado. Las variables de decisión para éste modelo serán las tasas de ingreso a la autopista a través de la rampa de acceso  $i$  en el segmento  $n$ , identificadas en el modelo como  $r_{ni}$ . Una vez acondicionado el modelo planteado para su experimento numérico, se procede a realizar su implementación mediante un algoritmo genético multiobjetivo, en este caso el NSGA-II. Por lo tanto se parte del modelo compuesto de nueve funciones objetivo. El comportamiento de los siete segmentos de la autopista se refleja en la ecuación uno ( $v=7$ ), donde se realiza la sumatoria del aporte de las rampas de cada segmento tomando en cuenta la tasa de salida definida para cada uno de estos. La segunda es una función que determina la maximización de la sumatoria de las 21 variables de decisión ( $k=21$ ), las cuales representan las rampas involucradas en el modelo y las siete funciones objetivo restantes reflejan el flujo de entrada a cada una de las secciones de la autopista.

Al expandir la ecuación de la primera función objetivo, surgen algunas simplificaciones matemáticas y es por ello que la función objetivo del experimento numérico pareciera ser más simple que la propuesta en el modelo matemático, quedando entonces el experimento numérico como sigue:

*En la siguiente ecuación se describe el comportamiento de la sección uno de la autopista:*

$$c_1 = (1 - t_1)(d_{11}r_{11} + d_{12}r_{12} + d_{13}r_{13}) - (t_1)(d_{11}r_{11} + d_{12}r_{12} + d_{13}r_{13})$$

*Siguiendo este mismo comportamiento, la sección dos de la autopista se describe con la siguiente ecuación:*

$$c_2 = (1 - t_2)(d_{21}r_{21} + d_{22}r_{22} + d_{23}r_{23}) - (t_2)(d_{21}r_{21} + d_{22}r_{22} + d_{23}r_{23}) + (t_1)(d_{11}r_{11} + d_{12}r_{12} + d_{13}r_{13})$$

*El modelo planteado indica que se debe maximizar el flujo vehicular total de la autopista, el cual está compuesto por los flujos parciales de cada una de las secciones, por lo tanto el flujo vehicular total del sistema en estudio se describe mediante la siguiente ecuación:*

$$F(\mathbf{c}) = \sum_{n=1}^7 c_n$$

*Realizando la suma de los dos primeros segmentos de la autopista se obtiene la siguiente expresión:*

$$c_2 + c_1 = (1 - t_2)(d_{21}r_{21} + d_{22}r_{22} + d_{23}r_{23}) - (t_2)(d_{21}r_{21} + d_{22}r_{22} + d_{23}r_{23})$$

*Aquí se evidencia la simplificaciones de los términos que definen el flujo vehicular que ha permanecido en la sección uno y sirve de entrada a la sección dos. Esto ocurre con las definiciones de los segmentos que componen la autopista en estudio, entonces el modelo se simplifica a sólo los flujos de entrada identificados, puesto que los términos que intervienen como las salidas de una sección son las entradas de la siguiente. Al realizar la sumatoria, estos términos se simplifican.*

Realizadas las sumatorias pertenecientes a la agrupación de los elementos que componen el modelo matemático, el experimento numérico queda configurado de la siguiente manera:

*Maximizar:*

$$1) (r_1 + r_2 + r_3)q_1 + (r_4 + r_5 + r_6)q_2 + (r_7 + r_8 + r_9)q_3 + (r_{10} + r_{11} + r_{12})q_4 \\ + (r_{13} + r_{14} + r_{15})q_5 + (r_{16} + r_{17} + r_{18})q_6 + (r_{19} + r_{20} + r_{21})q_7$$

$$2) \sum_{i=1}^{21} r_i$$

*Sujeto a:*

$$0 < r_{ni} \leq 1$$

$$0,4 < t_n \leq 0,6$$

$$d_{ni} > 0$$

$$(r_1 + r_2 + r_3) \leq \frac{q_n}{d_n}$$

$$(r_4 + r_5 + r_6) \leq \frac{q_n}{d_n}$$

$$(r_7 + r_8 + r_9) \leq \frac{q_n}{d_n}$$

$$(r_{10} + r_{11} + r_{12}) \leq \frac{q_n}{d_n}$$

$$(r_{13} + r_{14} + r_{15}) \leq \frac{q_n}{d_n}$$

$$(r_{16} + r_{17} + r_{18}) \leq \frac{q_n}{d_n}$$

$$(r_{19} + r_{20} + r_{21}) \leq \frac{q_n}{d_n}$$

Nótese que algunas de las restricciones planteadas no se han tomado en cuenta para este experimento numérico, las razones se explican a continuación:

$$(1 - t_n) \sum d_{ni} r_{ni} - (t_n) \sum d_{ni} r_{ni} + (t_{n-1}) \sum d_{(n-1)k} r_{(n-1)k} \leq q_n$$

La ecuación anterior describe formalmente el comportamiento de cada segmento de la rampa. Está reflejado en las funciones objetivo y las siete últimas restricciones.

$$(1 - t_n) \sum d_{ni} r_{ni} > 0$$

La ecuación anterior evita que el modelo tome valores negativos para la rampa de acceso, es una definición formal.

Las rampas de salida no forman parte de las variables de decisión, puesto que no se controlará su flujo. Dado el volumen de variables y restricciones, se selecciona como método de optimización un algoritmo híbrido multiobjetivo, el cual está compuesto por un algoritmo genético NSGA-II en su variación multiobjetivo y uno de búsqueda local también multicriterio, Hill Climbing para la selección de soluciones óptimas no dominadas.

#### **4.4 Implementación de la solución**

##### **4.4.1 Implementación de NSGA-II**

A continuación se hace una implementación del planteamiento del problema con un método de optimización combinatoria, para éste estudio se ha seleccionado el NSGA-II en su versión multiobjetivo. Se definió una configuración parecida a la utilizada por Meng (2009), esta consistió en definir un cromosoma de 210 bits para cada individuo de la población, destinando 10 bits a cada una de las variables de decisión (las 21 rampas existentes en la autopista en estudio), una probabilidad de 70% de cruce y 0.2% de mutación, el tamaño de población de 420 individuos, según Coello, Lamont, & Van Veldhuizen (2007) donde sugieren que debe ser el doble del tamaño del

cromosoma, se evolucionó a través de 100 generaciones. Las tasas de salida de cada segmento de la autopista varían entre 0,4 y 0,6. Las demandas por cada una de las rampas varían entre 480 y 1300 vehículos/hora, la demanda del canal de incorporación de cada uno de los segmentos es de 1500 vehículos/hora. Se considera un horizonte de tiempo de una hora. Se realizaron 30 ejecuciones con la configuración descrita, dado que cada una de las ejecuciones generó una frontera de Pareto, se calculó un promedio entre los vectores resultantes de cada una de las funciones objetivo, obteniendo entonces una tendencia de los valores. Los resultados se muestran en Tabla 1:

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

**Tabla 1: Resultados del NSGA-II.**

F1	F2	F3	F4	F5	F6	F7	F8	F9
3,0602	6,0652	756,6374	765,8247	729,7554	808,3200	877,4191	708,7995	829,2964
3,1887	6,2518	812,1293	862,8099	737,9072	705,3218	753,8159	879,2315	712,7768
3,0714	6,0694	839,5769	687,6903	777,0884	910,2608	753,9078	764,4520	722,1049
3,0072	6,0763	846,6895	788,4562	792,1236	787,1607	787,8811	865,5554	752,6720
2,8881	5,9428	713,2849	737,6833	740,7795	846,6286	789,8616	727,5663	822,2468
3,1073	6,2746	827,8122	783,4417	818,4190	836,9749	822,2766	795,1882	872,2747
3,2204	6,3232	956,8047	914,9940	772,8748	894,5337	887,1395	838,8023	860,7742
3,1672	6,3796	843,7638	729,6453	780,6992	836,9798	761,7461	879,9856	759,3020
3,0364	6,1004	854,6464	781,9189	756,2302	773,3621	783,0590	871,1578	797,7584
3,2431	6,5558	838,5503	872,9467	775,2099	731,0025	758,2215	900,2561	866,7190
2,9564	6,0592	793,4841	828,5614	755,4193	659,8009	756,3909	889,7477	795,6982
3,1347	6,1680	828,9780	863,5122	822,6473	889,5863	893,8029	723,5126	828,6137
3,0289	6,0822	788,0610	889,6287	705,9750	744,5725	773,1180	721,6202	778,7481
2,8856	5,9603	875,2162	767,6537	780,6149	817,2587	850,8005	810,3736	855,7752
2,9762	6,0433	818,2882	783,4160	763,6387	758,1642	703,1304	816,7356	844,8369
2,9964	6,0333	734,4047	854,8751	794,2873	765,8952	822,0733	813,7874	837,2913
3,0628	6,0129	772,7449	874,9545	701,6129	810,4107	831,8413	729,7209	794,2071
3,0663	6,0713	834,7299	667,0263	751,7306	794,7306	753,8192	821,7472	872,3958
3,1816	6,1834	803,0702	802,7027	696,2254	836,3347	730,7771	753,7502	675,7413
2,9513	6,0250	858,1175	735,1180	752,1083	808,8638	785,7840	741,1344	881,7234
3,0545	6,1392	849,3480	746,4172	692,3418	807,0841	886,3830	741,4375	778,8872
3,0893	6,0646	750,0053	672,7122	795,2525	823,2604	750,6069	800,7707	788,2721
3,0958	6,1468	830,5456	778,7255	853,1746	790,1180	785,7320	827,0982	810,5095
3,0808	6,1427	774,7578	837,0031	769,9458	925,6518	910,5549	726,2953	851,5846
2,9896	6,0793	803,3968	778,4327	748,1149	900,8231	787,9869	701,0946	786,2407
2,9717	5,9336	712,9167	666,2848	807,4049	784,6567	804,2629	737,8422	720,1228
2,8954	5,9578	802,0756	815,2623	823,7894	890,2090	825,8610	761,5398	723,0141
3,1517	6,1897	797,3348	780,2388	848,1980	758,5556	678,0334	714,3920	739,9610
3,0123	6,0382	884,3028	824,5759	780,5360	790,5485	769,6511	770,6489	692,3576
3,1735	6,2564	811,7824	862,7573	889,8674	887,1636	833,4500	811,4459	836,6125

Donde F1 hasta F9, representan cada una de las funciones objetivos presentes en el modelo planteado. Las dos primeras (F1 y F2), definen la maximización de la sumatoria de las variables de decisión, concentrando así toda la dinámica del sistema en una primera función, sin tomar en cuenta sus restricciones, es decir, sólo los valores máximos de estas, entonces cuanto mayor sean estos valores, reflejaran soluciones con un mayor rendimiento total del sistema vial. F1 corresponde a la sumatoria de las 21 variables de decisión tomando en cuenta la tasa de salida en cada sección de la autopista, mientras

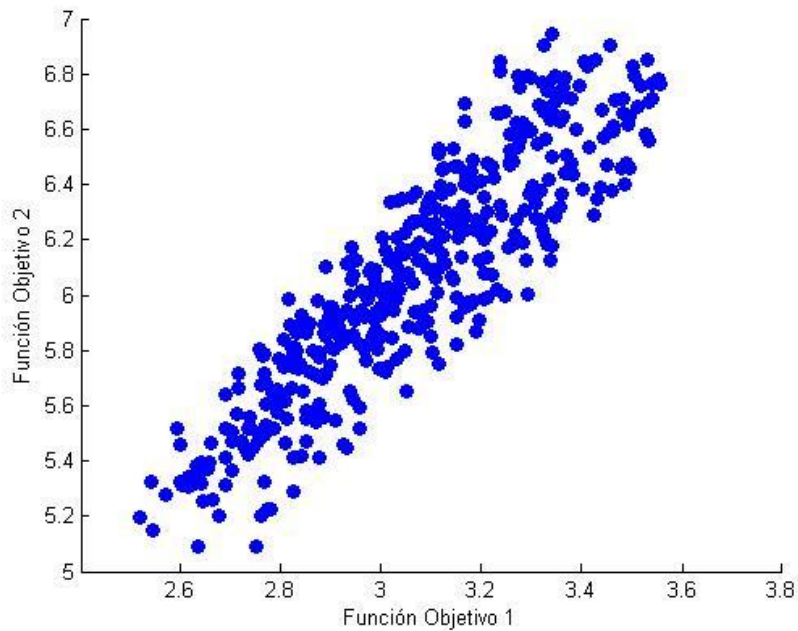
que F2 representa la simple suma de las variables de decisión. Las 7 funciones objetivo restantes representan la optimización del flujo vehicular en cada una de los segmentos de la autopista, en función de las rampas involucradas en ese segmento y los valores mostrados están expresados en vehículos por hora en cada sección de la autopista. De los resultados obtenidos en la Tabla 1, se calculan los promedios mostrados en la Tabla 2 a manera de un análisis cualitativo de los datos:

**Tabla 2: Promedio obtenido**

Promedios								
F1	F2	F3	F4	F5	F6	F7	F8	F9
3,0582	6,1209	813,7819	791,8423	773,7991	812,4744	796,9796	788,1897	796,2839

Con esto se observa una tendencia de los valores de cada una de las funciones objetivo que representan los segmentos de la autopista. Cabe destacar que cuanto mayor sea el valor de las dos primeras funciones objetivo, mayor será el flujo de vehículos que entran por cada rampa.

En la Figura 10 se muestran gráficamente las dos primeras funciones objetivo de una de las ejecuciones, donde se puede observar la tendencia de los valores obtenidos y la dispersión (de forma gráfica) de las soluciones. Se grafican las dos primeras funciones objetivo puesto que son éstas donde se define la sumatoria de todas las variables de decisión, entonces cuanto mayor sea su valor (máximos de la función F1 y F2), mayor será el flujo vehicular representado por la solución evaluada.



**Figura 10: Gráfica de F1 contra F2**

Los resultados observados del NSGA-II para este problema arrojan resultados aparentemente consistentes, pese a las consideraciones dinámicas del modelo tomadas para la implementación (variabilidad en las tasas de salidas de los segmentos de la autopista y variabilidad en las tasas de entrada de cada una de las rampas), sin embargo no es posible hacer una comparación en función de la efectividad del método, puesto que no se tiene otro resultado para contrastar la calidad del obtenido del NSGA-II.

En la siguiente parte del desarrollo se implementará un algoritmo que proporcionará otro mecanismo de selección sobre el resultado obtenido en el NSGA-II, es aquí donde se procederá a la hibridación, puesto que se combinarán dos mecanismos de optimización. Enmarcado en esta

investigación, según Raidl (2006), se realizará una hibridación del tipo metaheurística combinada con otra metaheurística, en este caso NSGA-II con HC, de bajo nivel con fuerte acoplamiento. En cuanto a su orden de ejecución se cataloga como un proceso en lote, puesto que primero es ejecutado el NSGA-II y al finalizar con la respuesta obtenida comienza la ejecución del HC. Por sus mecanismos de control es considerado colaborativo homogéneo, puesto que se integran esfuerzos computacionales de diferentes técnicas para la resolución de un mismo problema de optimización combinatoria en una misma ejecución.

Luego se realizarán pruebas comparativas a fin de contrastar ambos resultados, NSGA-II puro y la hibridación.

#### **4.4.2 Implementación del Hill Climbing (HC)**

A continuación se explica la implementación de una técnica de búsqueda heurística, con el objeto de conseguir una frontera de Pareto definida, dentro del conjunto de soluciones generadas por el NSGA-II, obteniendo así unas óptimas no dominadas, completando así la construcción del algoritmo híbrido objeto de este trabajo.

Dentro de los métodos heurísticos disponibles se utilizó el método Hill Climbing (HC) en su versión multiobjetivo. El algoritmo consiste compara un resultado seleccionado con sus vecinos dentro del espacio de soluciones

generados por el NSGA-II, con el fin de conseguir mejores de estas en cada iteración, basado en un criterio de mejora. Para este caso en particular de maximización de flujo, se seleccionan las funciones objetivo F1 y F2 como criterio de mejora, las cuales representan la suma de todas las variables de decisión (rampas de acceso a la autopista), de tal forma que son estas funciones las que concentran el fin del problema en cuestión, consiguiendo entonces aquellas que posean una mayor cantidad de vehículos ingresando a la autopista.

Para conseguir mejoras el algoritmo selecciona una solución aleatoria del espacio generado por el NSGA-II, a partir de allí comienza a buscar entre sus vecinos alguna de ellas que sea mejor, basado en los valores de F1 y F2. Busca entre sus vecinos inmediatos una solución mejor, de no conseguirla amplía su vecindad hasta conseguir una mejor o llegar al máximo de soluciones exploradas, evitando así estancarse en un mínimo local y teniendo la posibilidad de explorar, de manera exhaustiva, el espacio de soluciones generado por el NSGA-II. En la Figura 11 se muestra un pseudocódigo del HC:

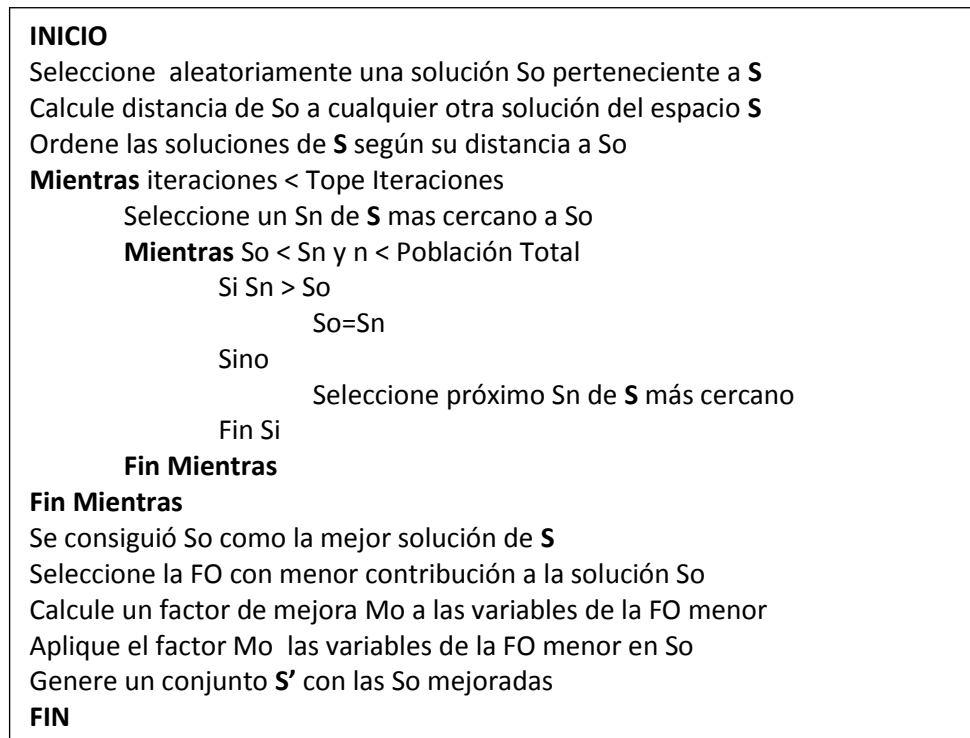


Figura 11: Pseudocódigo del Hill Climbing utilizado

El algoritmo mostrado en la Figura 11, selecciona como base de búsqueda una solución  $S_n$  aleatoria del espacio total  $S$ , generado por el NSGA-II, de allí calcula la distancia de la solución  $S_n$  seleccionada a las demás soluciones del espacio de búsqueda. Luego comienza a revisar los elementos, desde el más cercano hasta uno mejor a él o hasta que revise todos los individuos de la población, en cuyo caso será  $S_n$  la mejor solución del espacio  $S$ . De tal manera que ante un mejor elemento, toma éste último como un nuevo  $S_n$  y comienza la búsqueda nuevamente. Esto evita un estancamiento en un mínimo local y permite hacer una búsqueda pseudo-exhaustiva dentro de todo el espacio definido, entonces se puede obtener soluciones máximas dentro de éste espacio  $S$ .

Una vez que el HC ha conseguido una serie de soluciones no dominadas, se procede a realizar algunas mejoras aleatorias de acuerdo a un factor  $M$  calculado a partir de la diferencia entre las variables de decisión seleccionadas y el máximo posible de la restricción asociada, esto es:

$$M = \frac{1500}{1300} - (r_{nm} + r_{(n+1)m} + r_{(n+2)m})$$

La cual produce la mejora máxima aplicable a la función objetivo seleccionada, luego se genera un valor  $C_m$  aleatorio entre 0 y  $M$  el cual será la mejora total aplicada a la función objetivo seleccionada, se evalúa una proporción de contribución de cada  $r_{nm}$  y se aplica un criterio de incremento tomando en cuenta la  $r_{nm}$  con menor contribución como sigue:

$$r_{nm} = C_m \left( 1 - \left( \frac{r_{nm}}{r_{nm} + r_{(n+1)m} + r_{(n+2)m}} \right) \right)$$

De esta manera las  $r_{nm}$  con menor valor dentro de la función objetivo, recibirán una mejora mayor y aquellas con un valor mas alto, recibirán una menor. Se evalúa nuevamente la solución para verificar su validez en términos de las restricciones del problema y se incorpora al conjunto de soluciones mejoradas. Si por alguna razón la solución obtenida luego de la mejora no es válida por violar alguna de las restricciones, se genera una nueva solución con otro valor  $C_m$ , se realizan los cálculos de mejora, se aplican y se vuelve a evaluar la nueva solución, con el fin de no incluir en el análisis algún elemento

no válido y generar una población degenerada. Se realiza con cada una de las soluciones obtenidas del NSGA-II, con el fin obtener individuos más aptos en un tiempo computacional aceptable, razón principal para la implementación de algoritmos híbridos.

En la Tabla 3 se muestra el resultado 30 ejecuciones del algoritmo híbrido desarrollado:

**Tabla 3: Resultados del algoritmo híbrido**

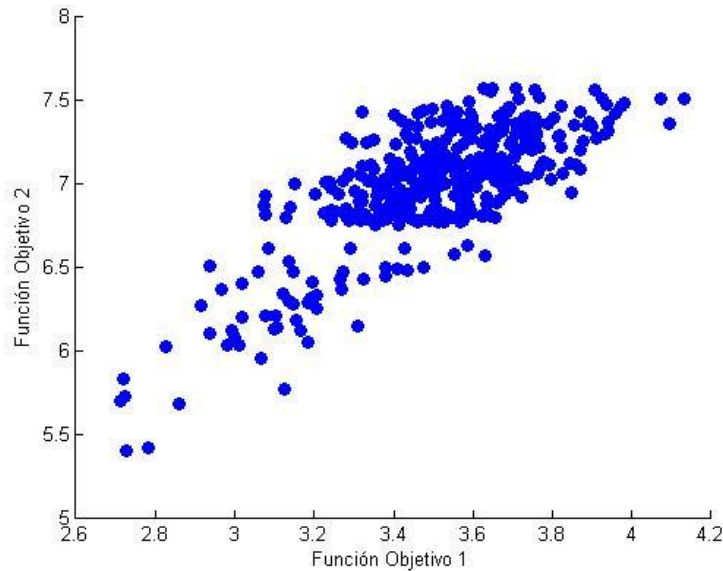
F1	F2	F3	F4	F5	F6	F7	F8	F9
3,6229	7,2429	1410,5057	1471,5108	1383,2299	1226,5558	1399,4617	1096,9365	1427,5542
3,5186	7,0409	1356,0909	1462,5197	1249,9314	1054,8388	1233,0730	1403,2824	1393,4403
3,5493	7,0913	1485,7856	1259,4229	1274,6936	1280,9010	1087,8486	1386,6222	1443,4346
3,4924	7,0133	1443,0709	1305,7277	1403,3856	982,8997	1375,9447	1216,3376	1389,9466
3,4495	6,9163	1295,2903	1219,1749	1317,0980	1230,7324	1258,0216	1336,8632	1333,9922
3,5019	6,9901	1326,1656	1362,6185	1307,0859	1277,9174	1294,7047	1270,0544	1248,5465
3,6711	7,3480	1341,2439	1405,8313	1445,2278	1361,4281	1309,4005	1246,9910	1442,2772
3,5533	7,0996	1172,3086	1377,0468	1321,5959	1370,1115	1253,1917	1333,7365	1401,4459
3,5281	7,0781	1406,7217	1159,6311	1409,7535	1437,2416	1475,2876	1044,0405	1268,8139
3,5489	7,0915	1136,0130	1310,1023	1433,7441	1267,7814	1414,4051	1305,9659	1350,9068
3,5077	7,0530	1386,2661	1245,3197	1211,6918	1249,5671	1319,0870	1297,7783	1459,2023
3,4715	6,9498	1410,2682	1344,6411	1302,6927	1289,9590	1140,7766	1265,0890	1281,3084
3,5770	7,1351	1182,9344	1320,5623	1241,2485	1415,1462	1448,3716	1354,4739	1312,8970
3,5617	7,1328	1377,0641	1279,0679	1459,4186	1397,2852	1357,4704	1304,4858	1097,7921
3,5323	7,0595	1399,1325	1254,4574	1365,9447	1282,4161	1356,6383	1302,2163	1216,5749
3,5133	7,0302	1227,9620	1270,5375	1360,3878	1407,9130	1319,3921	1144,5610	1408,5540
3,4695	6,9190	1365,6361	1451,0730	1460,2167	1053,1213	1146,3721	1249,6662	1268,6024
3,5438	7,0854	1285,8802	1461,0693	1361,0403	1200,3290	1492,5343	1102,2782	1307,8429
3,4462	6,9113	1147,3228	1311,0254	1264,1965	1226,3685	1237,2592	1402,9026	1395,6509
3,5205	7,0435	1381,7600	1324,9872	1382,2130	1328,2140	1254,4942	1299,5863	1185,2636
3,5432	7,0958	1329,8335	1353,9870	1082,3311	1220,7845	1481,9241	1380,2832	1375,3776
3,5131	7,0217	1166,2497	1414,5508	1378,3058	1307,4202	1328,3776	1212,2926	1321,0642
3,5186	7,0279	1361,2996	1229,6453	1244,4796	1309,4717	1353,2801	1217,0614	1421,0230
3,5187	7,0173	1354,4196	1329,9006	1320,1943	1200,7517	1190,1805	1398,6170	1328,4287
3,5629	7,1487	1456,2807	1471,8873	1244,4138	1231,8518	1106,8632	1436,7641	1345,3130
3,5317	7,0732	1371,8445	1343,2840	1353,5216	1364,5423	1142,3901	1414,2911	1205,3365
3,4997	6,9977	1122,3936	1340,2764	1340,5107	1315,3299	1179,9992	1464,7328	1333,7924
3,5334	7,0657	1210,9780	1327,8171	1276,3447	1475,0368	1386,8701	1233,7441	1274,6306
3,5444	7,0932	1482,3525	1485,8580	1428,3643	1203,3616	1207,1022	1253,5208	1160,6519

Nuevamente F1 hasta F9, representan cada una de las funciones objetivo, F1 es la sumatoria de las variables de decisión en tomando en cuenta la tasa de salida asociada a cada sección de la autopista, F2 es el valor obtenido por la suma de los valores de las 21 variables de decisión, las 7 restantes representan la optimización del flujo vehicular en cada una de los segmentos de la autopista y su valor está expresado en vehículos por hora en cada sección. En la Tabla 4 se muestra los promedios de los resultados obtenidos.

**Tabla 4: Promedios de las funciones objetivo del algoritmo híbrido**

Promedios									
F1	F2	F3	F4	F5	F6	F7	F8	F9	
3,5313	7,0649	1324,1880	1341,7696	1329,7554	1280,9262	1300,8319	1282,0317	1324,8459	

En la Figura 12 se muestra un gráfico de dispersión con las dos primeras funciones objetivo, las cuales, como ya se ha mencionado, representan la sumatoria de las variables reguladoras del flujo vehicular de las rampas de acceso a las secciones de la autopista, por lo tanto cuanto mayor sea el valor de éstas dos funciones, mayor flujo vehicular estará circulando por toda la autopista.



**Figura 12: Gráfico de dispersión de las funciones objetivo 1 y 2 luego del HC**

[www.bdigital.ula.ve](http://www.bdigital.ula.ve)

Tal como se muestra en la Figura 12, a simple vista se puede apreciar una dispersión un poco más baja que en el NSGA-II, mostrada en la Figura 10, sin embargo las soluciones se ubican en un rango de valores un poco más altos y se agrupan más fuerte en la parte superior de la gráfica, lo que indica que la mayoría de estas se han desplazado a un espacio de soluciones más favorable para el problema. La irregularidad en la agrupación de las soluciones se debe a la naturaleza dinámica del modelo, puesto que dos de las variables consideradas son valores que obedecen a un comportamiento aleatorio dentro de un rango de valores definidos por el problema (demanda por rampa: 480 – 1330 v/h y tasa de salida de sección de la autopista: 0,4 – 0,6), lo cual le agrega cierta dispersión en distribución del grupo de soluciones por la naturaleza

dinámica de estas variables, sin embargo son las mejores que ha conseguido el algoritmo de búsqueda local a partir de la población generada por el NSGA-II, para este caso del problema.

Con el fin de comprender mejor el significado de los resultados obtenidos, a continuación se describe un ejemplo de un resultado numérico de una solución. Dado el conjunto de soluciones mostrado en la Figura 12, se selecciona una solución al azar perteneciente a este conjunto, la cual está compuesta por 21 variables, los cuales representan la proporción de la demanda de vehículos por hora que pueden pasar a través de una rampa. En la Tabla 5 se muestra las variables involucradas en la solución seleccionada:

www.bdigital.ula.ve  
**Tabla 5: Variables involucradas en una solución**

r1	r2	r3	r4	r5	r6	r7
0.196284	0.039062	0.744123	0.221674	0.324211	0.525378	0.126950
r8	r9	r10	r11	r12	r13	r14
0.120114	0.826153	0.018554	0.296868	0.681625	0.061522	0.162106
r15	r16	r17	r18	r19	r20	r21
0.713850	0.244532	0.411019	0.323174	0.431630	0.231440	0.393545

Los valores de las variables en sí son adimensionales, puesto que no se mide en alguna escala, sólo representan la proporción o razón de vehículos que pasarán por cada una de las rampas, de la demanda total calculada para esta solución. Por ejemplo para r4 sólo pasará el 22,16% de la demanda total por hora calculada. Al introducir estos valores en las funciones objetivo del problema, se obtiene en las dos primeras funciones objetivo, una sumatoria de

todos los valores de las rampas, la primera función una suma individual y la segunda una suma asociada a la capacidad de cada uno de los segmentos de la autopista, las otras siete funciones representan la cantidad de vehículos, expresada en vehículos/hora, que tendrán acceso a cada uno de las secciones de la autopista, es entonces cuando las variables adimensionales toman significado en términos reales para el modelo planteado. En la Tabla 6 se muestra los resultados obtenidos de las funciones objetivo luego de aplicar las variables al modelo planteado.

**Tabla 6: Resultado de las funciones objetivo**

F1	F2	F3	F4	F5	F6	F7	F8	F9
3,555709	7,093813	1273,31	1392,642	1395,182	1296,161	1218,721	1272,341	1373,599

Se observa en la Tabla 6 que los valores de F1 y F2 son bastante menores a los obtenidos en las funciones F3 a F9, esto obedece a que en F1 y F2 sólo se suman las variables reguladoras del flujo de las rampas (identificadas como  $r_1 \dots r_{21}$  en la Tabla 5), las cuales varían entre 0 y 1, representando la proporción de vehículos por hora que acceden a una rampa en particular. En F1 se toma en cuenta la tasa de salida de cada una de las secciones de la autopista y en F2 sólo la sumatoria total de las 21 variables. Al realizar la sumatoria se obtiene un valor mucho menor por los valores de las variables.

Los valores arrojados por F1 y F2 representan un indicador de flujo máximo del sistema completo, es decir entre más alto sea su valor, mayor será

el flujo vehicular total entregado por la solución, de tal forma que se puede determinar inferir que un valor alto en estas funciones, significa que es una mejor solución. F1 representa la relación entre el flujo en las rampas y el flujo en cada sección. F2 es un indicador del grado de acceso de todas rampas del sistema. Por ello, tomé a F1 y F2 como parámetros para medir la calidad de las soluciones obtenidas. Los valores obtenidos de las funciones objetivo F3 a F9 representan los flujos vehiculares en cada una de las secciones de la autopista. Por ejemplo el valor de F3 representa que el tráfico regulado en las rampas de acceso, según los resultados de las variables y la demanda, es de 1273,64 vehículos/hora, lo que quiere decir que cada uno de los controles instalados en las rampas permiten que la demanda de cada una de ellas establezcan un flujo hasta ese valor.

## Capítulo 5: Discusión y análisis de resultados

Una vez implementado y probado, en una primera versión, el NSGA-II y luego el algoritmo híbrido, se procede a realizar una serie de pruebas comparativas, con el fin de verificar cual de los métodos produce mejores soluciones a partir del mismo contexto del problema. Así una mejor solución será aquella que proporcione valores máximos entre las dos funciones objetivo en comparación.

Como una primera comparación se analizan las fronteras de Pareto obtenidas por ambos algoritmos mostradas en la Figura 13:

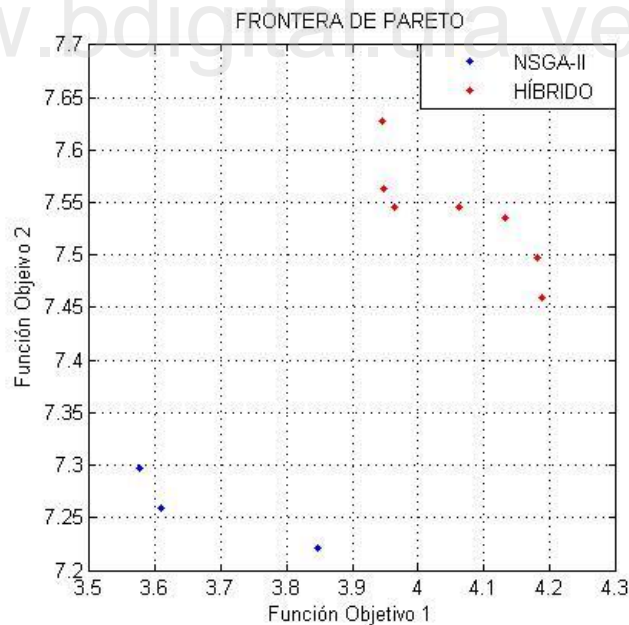
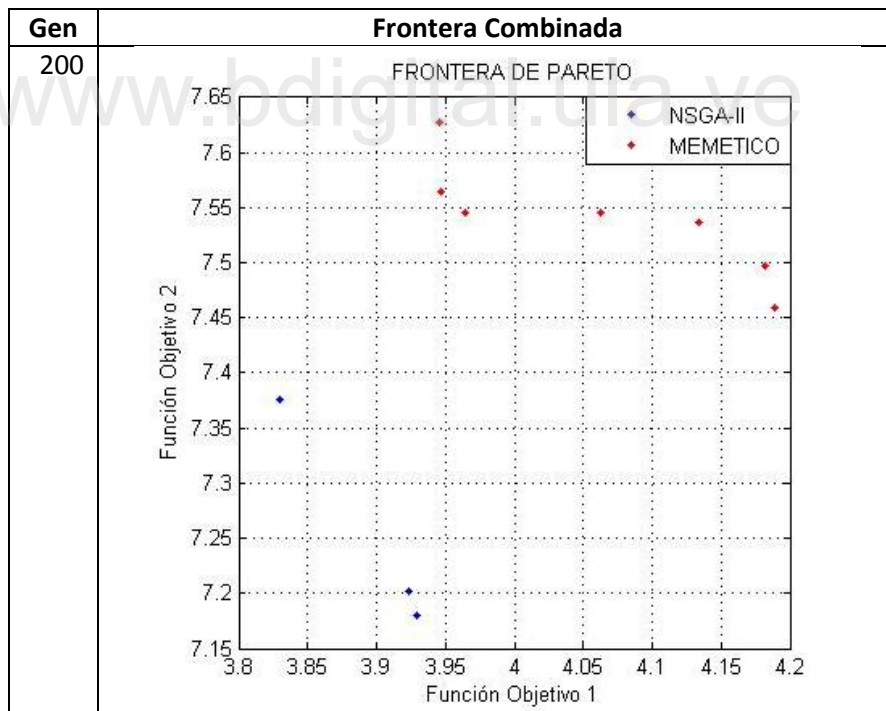
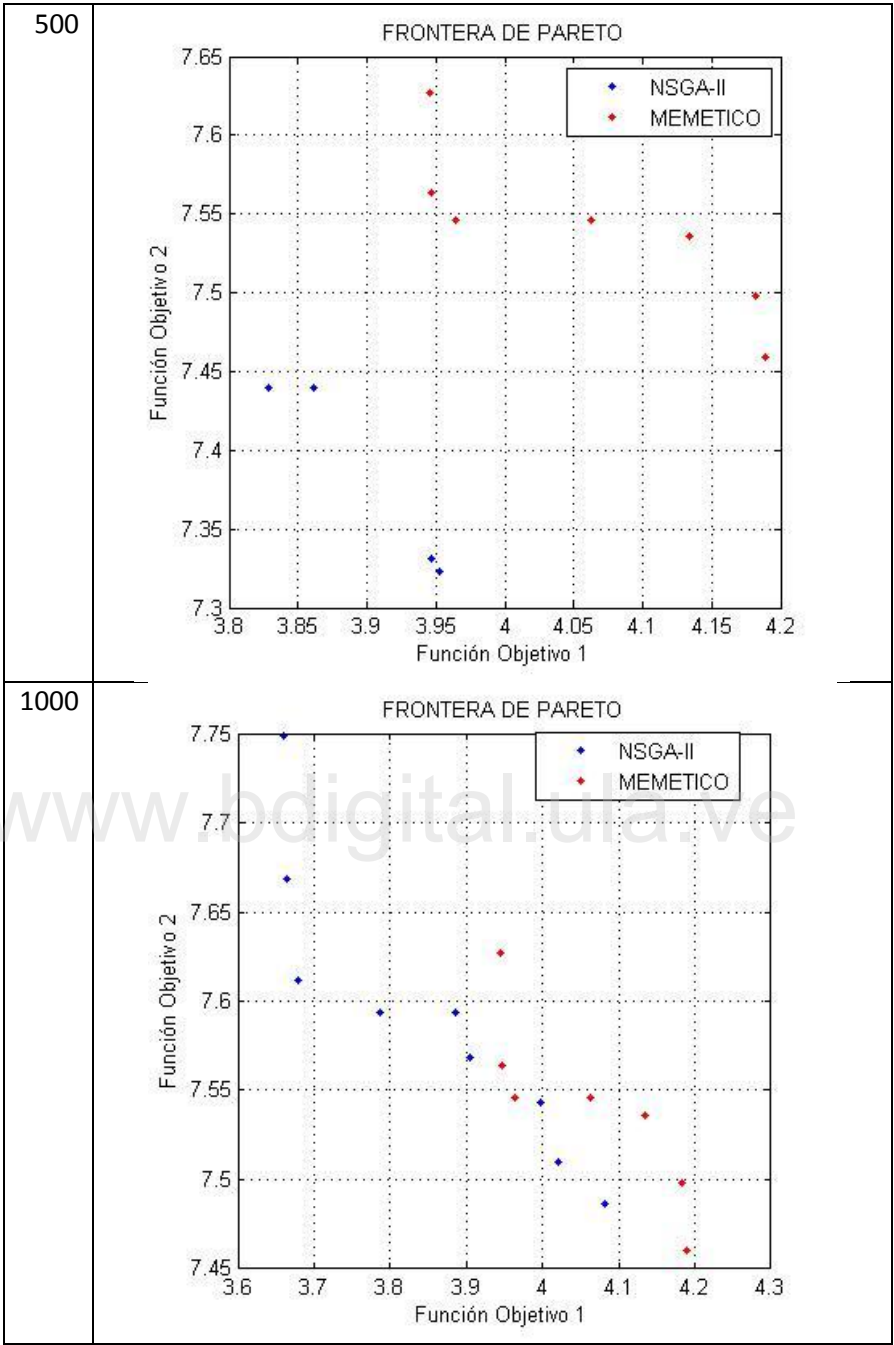


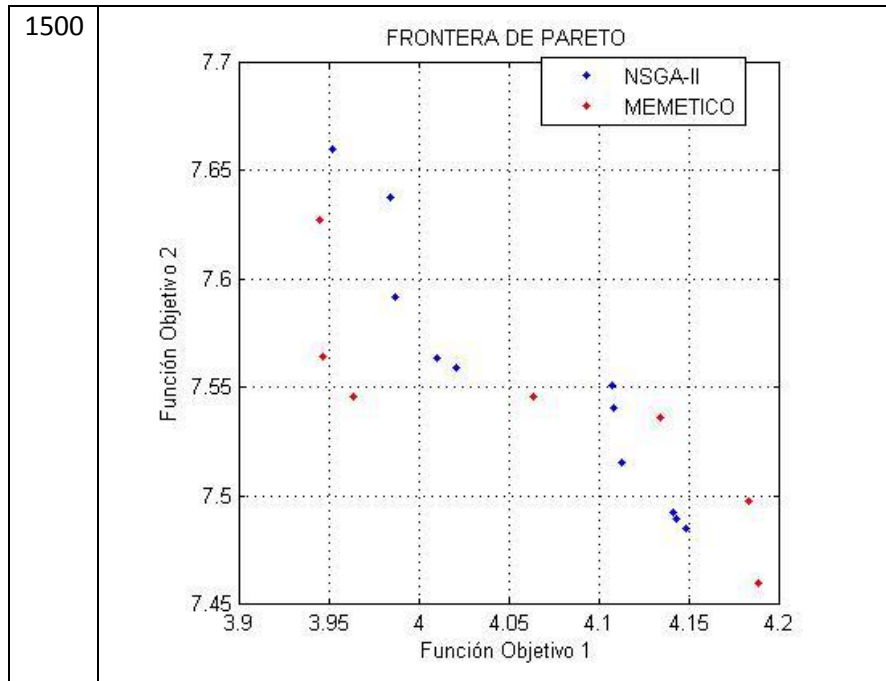
Figura 13: Frontera de Pareto generada por el NSGA-II y el Algoritmo Híbrido comparado en 100 generaciones

En una primera observación se puede notar la diferencia entre la frontera del NSGA-II y el híbrido implementado, éste último ha trasladado las soluciones a una zona donde sus valores son más altos, es decir que el algoritmo híbrido ha logrado, aparentemente, mejorar las soluciones generadas por el NSGA-II. Se verifica la solución obtenida por el algoritmo híbrido con 100 generaciones en contraste con el NSGA-II en 200, 500, 1000 y 1500 generaciones, con el fin de revisar las mejoras introducidas por la hibridación con un costo computacional menor. En la Tabla 7 se muestra este contraste.

**Tabla 7: Frentes de Pareto. Híbrido 100 generaciones contra NSGA-II en 200, 500, 1000 y 1500 generaciones**







Se puede observar gráficamente, que las soluciones obtenidas por el algoritmo híbrido pudieran ser mejores que las generadas por el NSGA-II, puesto que la frontera obtenida por éste último necesita cerca de 1500 generaciones para alcanzar los valores obtenidos en 100 iteraciones por el algoritmo híbrido. Entonces se puede afirmar que en una primera observación los resultados obtenidos por el algoritmo híbrido son mucho mejores en cuanto a calidad y esfuerzo computacional empleado. Es entonces cuando la combinación del NSGA-II y el HC ofrecen una potencia mucho mayor que cada uno de ellos por separado, para el experimento numérico implementado de la instancia del problema en estudio.

Para verificar esta apreciación gráfica, se procede a realizar dos medidas de calidad *Hypervolume* y *R2*, las cuales son dos indicadores que miden el rendimiento de un algoritmo multiobjetivo en términos de calidad de resultados, medida necesaria para comprar al NSGA-II con el HC en este trabajo. Existen otros métodos o indicadores para evaluar estas medidas, sin embargo se utilizarán las dos anteriormente mencionadas puesto que son las más recomendadas para realizar la comparación de dos MOEA (Basseur, Talbi, Nebro, & Alba, 2006).

A continuación se hace una breve reseña de los métodos comparativos utilizados:

## **5.5 Métodos comparativos**

### **5.5.1 Indicador de Hypervolume**

El indicador de hypervolume fue propuesto por primera vez por Zitzler y Thiele (Zitzler & Thiele, 1999). Es una medida utilizada en resultados de problemas de optimización multiobjetivo, para evaluar el rendimiento de los algoritmos en términos de la calidad de sus resultados y capacidades de búsqueda. Dados dos conjuntos de Pareto A y B, los cuales representan dos conjuntos solución a un problema multiobjetivo dado, se dice que A es mejor que B si A domina a B. Desde un punto de vista geométrico, el indicador de hypervolume mide el volumen cubierto, en el espacio objetivo, por los

elementos de la aproximación. Para cada solución  $i \in A$  se construye un hipercubo de volumen  $v_i$ . La métrica calculada mide el volumen de la unión de todos los hipercubos. Se considera que mientras más grande es el valor del hipervolumen, mejor será la aproximación (Sánchez, 2008). Un ejemplo se muestra en la Figura 14:

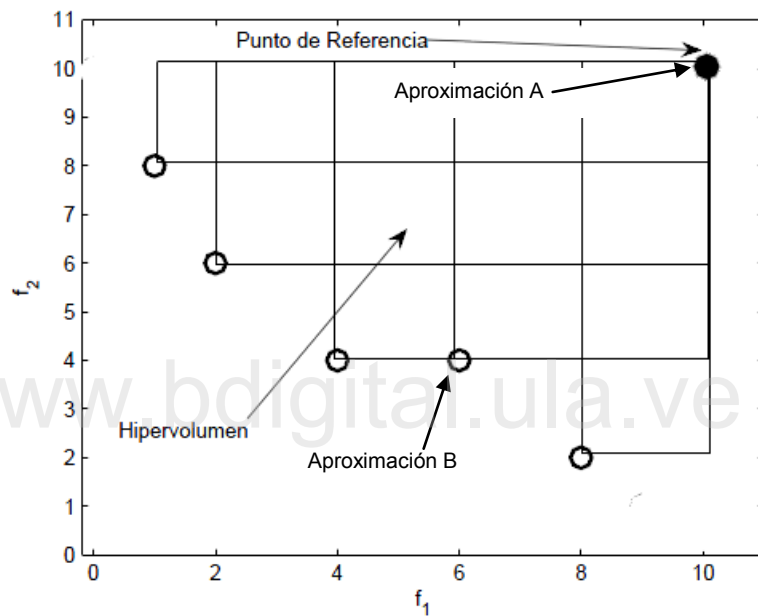


Figura 14: Ejemplo de cálculo para el indicador de hypervolume

En la Figura 14 se muestra dos aproximaciones A y B, en un problema de dos objetivos  $f_1$  y  $f_2$ . La aproximación de A es mejor que la de B con respecto al indicador de hypervolume, puesto que el volumen de los hipercubos que cubren el espacio de solución B, es mayor para el punto de referencia tomado de A.

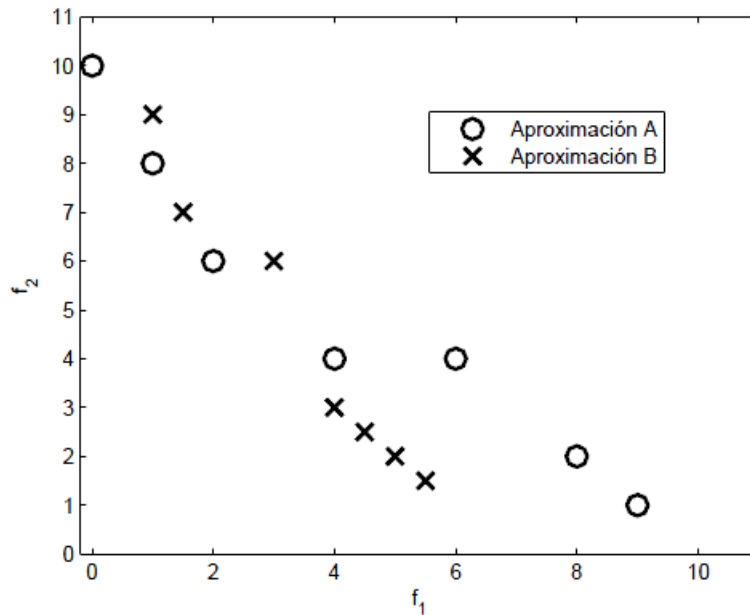


Figura 15: Dos aproximaciones a un problema multiobjetivo

### 5.5.2 Indicador R2

Propuesto por (Pilegaard & Jaszkiwicz, 1998), donde definen varios indicadores de calidad basados en conceptos de probabilidad, denominados métricas R (R-metrics por su nombre en inglés). Dado un conjunto de funciones  $U$  utilizadas para modelar todas las preferencias posibles del tomador de decisiones de un problema multiobjetivo, para cada  $u \in U$ , entonces  $u^*(A)$  es el máximo valor alcanzado por  $u$  en  $A$ . Entonces  $p(u)$  es la probabilidad de que el tomador de decisiones escoja a  $u$  como una buena solución. Igualmente se define otro conjunto de soluciones  $B$  que proporciona soluciones  $v \in U$ . Suponga que las funciones objetivo o de utilidad están construidas de tal forma

que todas ellas pueden sumarse, entonces el beneficio esperado de una aproximación puede calcularse como:

$$E(u^*(A)) = \int_{u \in U} u^*(A)p(u)du$$

El valor de utilidad esperado de un conjunto aproximado es, por si mismo un buen indicador de calidad unario. Suponga que todas las funciones  $u(z)$  son formuladas para medir el ingreso monetario obtenido de una solución candidata  $z$ . Entonces  $E(u^*(A))$  representa el ingreso esperado que es posible obtener utilizando a  $A$ .

Se definen un indicador de calidad binario conocido como R2, basado en la formula de valor esperado planteada anteriormente:

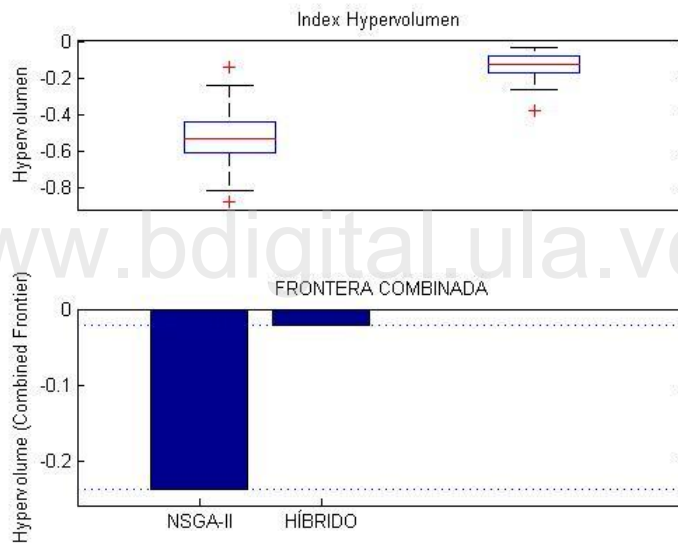
$$R2(A, B, U, p) = E(u^*(A)) - E(u^*(B))$$

Entonces  $A$  es mejor que  $B$  si  $R2(A, B, U, p) > 0$ . Es decir que el valor de la función de probabilidad de valor esperado de  $A$  es mayor que el valor de la función de probabilidad de  $B$ .

## 5.6 Comparación

Como resultado de la comparación, será el mejor algoritmo el que tenga el frente con mayor valor de *Hypervolume*. Para este caso en específico se escoge un punto con una sumatoria de variables de decisión mínimo, a fin de

establecer el área dominada desde un punto no-dominado, puesto que se refiere al estudio de un problema de maximización. El segundo indicador, índice R2, estima la cercanía de la solución generada por el algoritmo sobre el frente real o alguna referencia, por lo tanto cuanto mas pequeño es el indicador de R2, mejor es el algoritmo (Colomine, Cotta, & Fernández, 2012). En la Figura 16 se muestran las gráficas comparativas de los indicadores de *Hypervolume* y R2, de los resultados del NSGA-II y el algoritmo híbrido:



**Figura 16: Índice de Hypervolumen. Híbrido 100 generaciones contra NSGA-II 100 generaciones**

Como se mencionó anteriormente, un valor grande de *hypervolume* indica que ese algoritmo es mejor, por lo tanto según el resultado de ésta medida se puede verificar que las soluciones generadas por el algoritmo híbrido son mejores que las generadas por el NSGA-II, puesto que en la Figura 16 se

puede ver un valor más alto por parte del algoritmo híbrido. Luego de esto se realiza el cálculo del índice R2, el cual se muestra en la Figura 17.

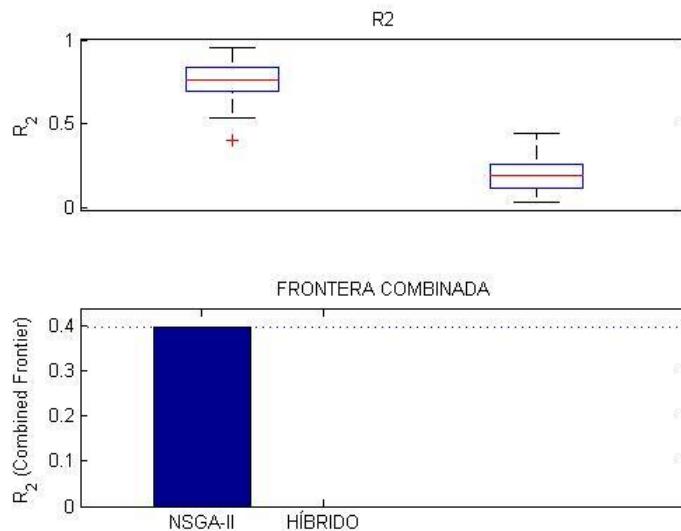
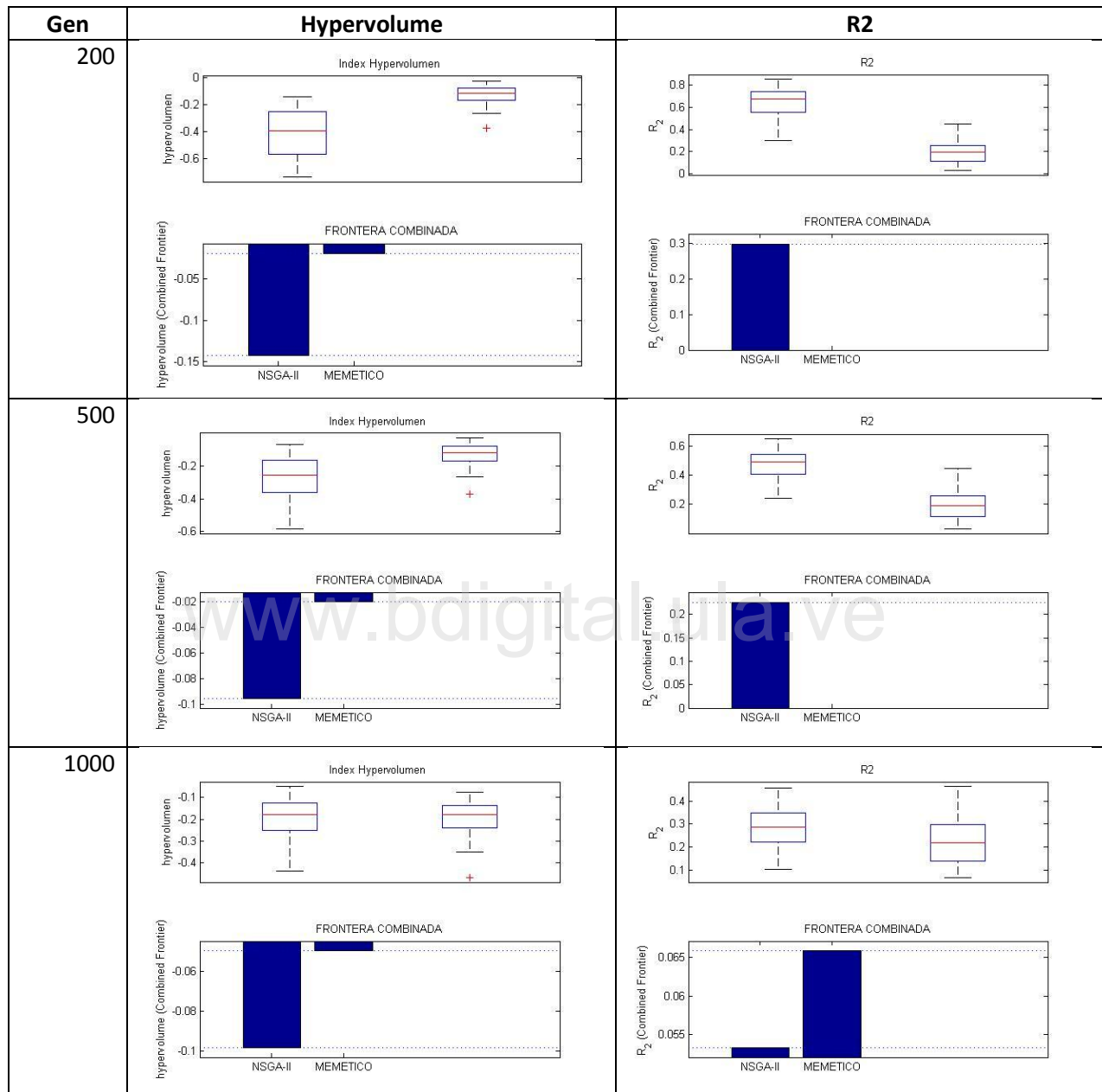


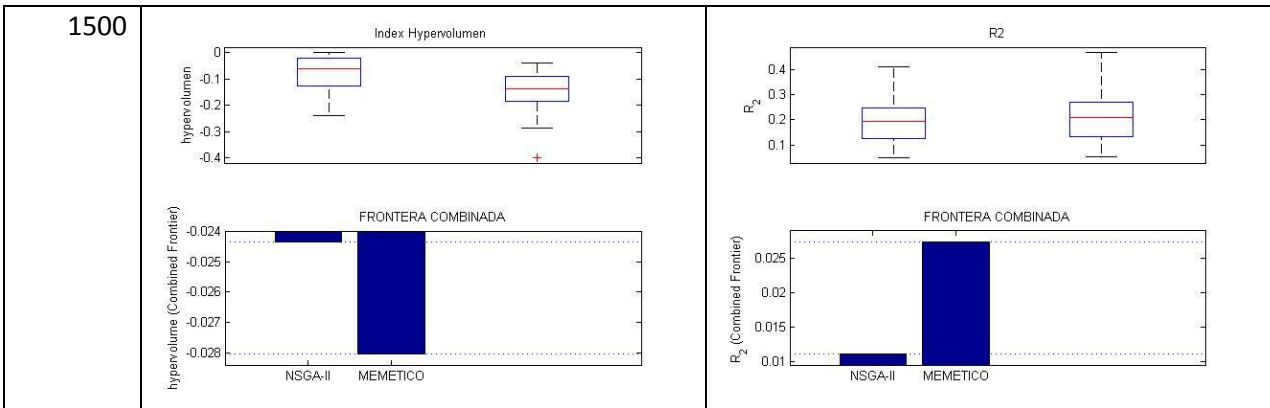
Figura 17: Índice R2. Híbrido 100 generaciones contra NSGA-II 100 generaciones

Según la Figura 17 se puede observar que el algoritmo híbrido nuevamente tiene un mejor valor que el NSGA-II, puesto que su índice R2 tiene un valor más pequeño, es decir que sus resultados son más cercanos a una solución óptima de referencia.

Una comparación entre los resultados del NSGA-II y el algoritmo híbrido se muestra en la Tabla 8 donde se observa un contraste entre el resultado del algoritmo híbrido de 100 generaciones y el NSGA-II con 200, 500, 1000 y 1500 generaciones. Se evidencia la diferencia entre la calidad de los resultados.

**Tabla 8: Índices de Hypervolume y R2. Híbrido de 100 generaciones contra NSGA-II de 200, 500, 1000 y 1500**





Como puede verse en la Tabla 8, el algoritmo híbrido produce un resultado sólo superable por el NSGA-II cuando trabaja sobre 1000 generaciones, sin embargo el esfuerzo computacional para llegar a soluciones de una calidad similar es muy superior, puesto que el algoritmo híbrido sólo le toma 100 generaciones para alcanzar los mismos valores que el NSGA-II puro en 1000, evidenciando entonces la eficiencia introducida por la hibridación en cuanto a la convergencia de las soluciones.

De lo anterior se puede comprobar la mejora significativa y la enorme utilidad de los algoritmos híbridos en el área de solución de problemas multiobjetivo. Son capaces de conseguir soluciones de buena calidad con un esfuerzo computacional aceptable, tal como se puede observar en las pruebas realizadas sobre el experimento numérico del problema objeto de esta investigación.

## Conclusiones

Se obtuvo el fundamento teórico para construir el resto de la investigación y los lineamientos sobre el cual se desarrolló el resto de las actividades llevadas a cabo, luego de planteado el problema y comprendido los elementos claves para su desarrollo, se construyó un modelo matemático que abarca una abstracción precisa del comportamiento definido, siguiendo una metodología adecuada para este fin, lo que permitió el desarrollo de diferentes modelos que fueron evolucionando hasta el utilizado finalmente. Las modificaciones se hicieron en función de la observación y pruebas preliminares, logrando entonces un modelo que representara el comportamiento del sistema en un horizonte de tiempo determinado, para este caso se tomó una hora, tal como se describe en Meng (2009).

El modelo y la documentación revisada permitieron realizar comparaciones de algunos algoritmos empleados en esta área, apoyándose en experiencias previas, se consiguió el diseño de la metaheurística utilizando dos algoritmos ampliamente implementados en este tipo de problemas, así como también orientar las pruebas en función de las características y requerimientos de información del experimento numérico implementado para hacer la comparativa de los resultados.

La implementación se llevó a cabo construyendo el diseño propuesto, basado en las variables de decisión que conforman el modelo planteado. Se

utilizó el algoritmo NSGA-II y un procedimiento *Hill Climbing*, el cual utilizaba como punto de partida para su búsqueda la población final alcanzada por el NSGA-II, de allí comienza a escalar hacia las mejores soluciones disponibles dentro de los individuos de esta, una vez que los ha conseguido comienza a introducir mejoras graduales en las soluciones, respetando las restricciones propias del modelo, para entonces conseguir una nueva población con mejores valores en sus funciones objetivo.

Para el planteamiento del problema hecho y el experimento numérico configurado, se logró una mejora sustancial de la calidad de los resultados mediante el algoritmo híbrido, con tan solo 100 generaciones, la cual pudo ser superada hasta las 1000 generaciones con el NSGA-II, consiguiendo entonces una convergencia mucho más eficiente en términos de calidad de resultados y esfuerzo computacional, logrando un algoritmo híbrido mucho más eficiente que el NSGA-II aplicado para la instancia numérica del problema planteado. Esto se fundamenta en pruebas comparativas para éste tipo de algoritmos, como lo son el índice de *Hypervolume* y  $R_2$ , los cuales analizan la calidad de los resultados de ambos algoritmos. Sin embargo los resultados obtenidos no concluyen que para todas las instancias del problema o para todos los casos de este tipo de aplicaciones multiobjetivo, la hibridación sea mejor que el algoritmo NSGA-II.

## Recomendaciones

Como parte de la continuación del proyecto, se recomienda realizar pruebas con otros algoritmos evolutivos y contrastar los resultados con el propuesto en este trabajo, así como definir una hibridación de tipo integrativa, es decir implementar el mecanismo de búsqueda dentro del NSGA-II como un operador más del algoritmo evolutivo.

El diseño propuesto es suficientemente flexible para incluir más variables o experimentar con más generaciones. Una prueba de este tipo podría realizarse para medir la efectividad del método propuesto en otras instancias del problema.

Otro elemento interesante, corresponde a la evaluación de otros problemas con la metodología propuesta, de forma tal de validar la generalidad de la misma en diferentes contextos.

## Bibliografía

- Auger, A., Bader, J., Brockhoff, D., & Zitzler, E. (2009). Theory of the hypervolume indicator: optimal  $\mu$ -distributions and the choice of the reference point. *FOGA '09 Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms* (págs. 87-102). New York: ACM.
- Basseur, M., Talbi, E., Nebro, A., & Alba, E. (2006). *Metaheuristics for Multiobjective Combinatorial Optimization Problems: Review and recent issues*. Orsay Cedex, Francia: INRIA.
- Benayoun, R., Roy, B., & Sussman, B. (1966). Electre: Une méthode pour guider le choix en présence de points de vue multiple. *Direction Scientifique No 49*.
- Bringmann, K., & Friedrich, T. (2010). Tight Bounds for the Approximation Ratio of the Hypervolume Indicator. *PPSN'10 Proceedings of the 11th international conference on Parallel problem solving from nature: Part I* (págs. 607-616). Berlin, Heidelberg: Springer-Verlag.
- Canales, S. (2004). *Métodos Heurísticos en Problemas Geométricos. Visibilidad, Iluminación y Vigilancia*. Universidad Politécnica de Madrid, Facultad de Informática, Madrid.
- Chankong, V., & Haimes, Y. (1983). *Multi-objective Decision making Theory and Methodology*. Elsevier Science Publishing Co.
- Charnes, A., & Cooper, W. W. (1961). *Management Models and Industrial Applications of Linear Programming, volume 1*. New York: John Wiley.
- Coello, C. (2010). *Introducción a la Computación Evolutiva*. Mexico DF: Departamento de computación. CINVESTAV-IPN.
- Coello, C. A. (1999). A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. *Knowledge and Information Systems. Vol. 1, No. 3*, 269-308.

- Coello, C. A. (1999). An Updated Survey of Evolutionary Multiobjective Optimization Techniques: State of the Art and Future Trends. *Congress on Evolutionary Computation, volume 1*, 3-13.
- Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer Science+Business Media, LLC.
- Cohon, J. L., & Marks, D. H. (1975). *A Review and Evaluation of Multiobjective Programming Techniques*. Water Resources Research.
- Colomine, F., Cotta, C., & Fernández, A. (2012). A Comparative Study of Multi-objective Evolutionary Algorithms to Optimize the Selection of Investment Portfolios with Cardinality Constraints. *EvoApplications 2012*, 165-173.
- Conesa, J., Contero, M., & Company, P. (S.F). *Comportamiento de los algoritmos de optimización en la reconstrucción geométrica de sólidos*. Univer.Politécnica de Cartagena: Dpto. de Expresión Gráfica.
- De la Cruz, J. (2003). *Análisis comparativo de las aproximaciones heurísticas Ant-Q, recocido simulado y búsqueda tabú en la solución del problema del agente viajero*. Colombia: Grupo de investigación en Sistemas Inteligentes. Universidad del Norte.
- Deb, K., & Goel, T. (2001). Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence. *First International Conference on Evolutionary Multi-Criterion Optimization*, 67-81.
- Deb, K., Pratab, A., & Meyarivan, T. (2000). *A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II*. Kanpur, India: Indian Institute of Technology.
- Dorigo, M., Maniezzo, V., & Colorni, A. (1996). The Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, 29-41.

- Edgeworth, F. Y. (1881). *Mathematical Psychics*. Londres: P. Keagan.
- Fonseca, C. M., & Fleming, P. J. (1993). Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. *Proceedings of the Fifth International Conference on Genetic Algorithms*, 416-423.
- Fonseca, C. M., & Fleming, P. J. (1993). Multiobjective Genetic Algorithms. *IEE Colloquium on Genetic Algorithms for Control Systems Engineering*, 6/1-6/5.
- Fonseca, C. M., & Fleming, P. J. (1994). *An Overview of Evolutionary Algorithms in Multiobjective Optimization*. Sheffield, UK: Technical report, Department of Automatic Control and Systems Engineering, University of Sheffield.
- Fourman, M. P. (1985). Compaction of Symbolic Layout using Genetic Algorithms. *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, 141-153.
- Gestal, M., & Rivero, D. (2010). *Introducción a los Algoritmos Genéticos y la Programación Genética*. La Coruña: Universidad da Coruña.
- Glover, F. (1986). *Tabu Search: A Tutorial*. University of Colorado, Center of Applied Artificial Intelligence.
- Glover, F., & Laguna, M. (1997). *Tabu Search*. Boston: Kluwer.
- Goicochea, A., Duckstein, L., & Fogel, M. (1976). Multi-objective programming in watershed management: A study of the Charleston watershed. *Water Resources Research*, 1085-1092.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Massachusetts: Addison-Wesley Publishing Company.
- Gomes, G. (2004). Optimization and Microsimulation of On-ramp Metering for Congested Freeways. *California PATH Research Report*.
- Gomes, G. H. (2006). Optimal freeway ramp metering using the asymmetric cell transmission model. *Transportation Research Part C* 14, 244–262.

- Hansen, P., & Mladenovic, N. (2003). Variable Neighborhood Search. En F. Glover, & G. Kochenberger, *Handbook of Metaheuristics* (págs. 145-184). Kluwer Academic Publishers.
- Horn, J. (1997). Multicriterion Decision making. *Handbook of Evolutionary Computation*, F1.9:1–F1.9:15.
- Horn, J., Nafpliotis, N., & Goldberg, D. E. (1994). Niche Pareto Genetic Algorithm for Multiobjective Optimization. *IEEE World Congress on Computational Intelligence, volume 1*, 82-87.
- Hoyos, J., & Pérez, J. (2007). Aplicación de algoritmos genéticos a la clasificación de imágenes de satélite en el marco de los servicios grid inteligentes. *Ciencia e Ingeniería Neogranadina*, 95-109.
- Hoyos, J., & Pérez, J. (2007). Aplicación de algoritmos genéticos a la clasificación de imágenes de satélite en el marco de los servicios grid inteligentes. *Ciencia e Ingeniería Neogranadina*, 95-109.
- Hurwicz, L. (1964). *Programming in Linear Spaces*. Londres: Oxford University Press.
- Ijiri, Y. (1965). *Management Goals and Accounting for Control*. North-Holland, Amsterdam.
- Jaziri, W. (2008). *Local Search Techniques: Focus on Tabú Search*. Vienna: I-Tech Education and Publishing KG.
- Jutler, H. (1967). Liniejnaja model z nieskolkimi celevymi funkcijami (Linear Model with Several Objective Functions). *Ekonomika i matematiceckije Metody*, 397-406.
- Kellerer, H., Pferschy, U., & Pisinger, D. (2003). Knapsack Problems. *Mathematics - Monograph (English)*.
- Kuhn, H. W., & Tucker, A. W. (1951). Nonlinear Programming. *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, 481-492.

- Kumar, R. (2005). Multiobjective EA Approach for Improved Quality of Solutions for Spanning Tree Problem. *Lecture Notes in Computer Science*, 811-825.
- Liu, D. (2007). A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization. *IEEE Transactions on systems, man, and cybernetics—Part B: Cybernetics*, vol. 37, 42.
- Marczyz, A. (2004). *Genetic Algorithms and Evolutionary Computacion*.
- Maroszek, M., & Rettig, C. (2008). *A Tabu Search for the p-Median Problem*. Hamburgo: Institute of Information Systems, School of Business, University of Hamburg.
- Martínez, C. (2011). *Metaheurísticas híbridas aplicadas al Problema de Ruteo de Arcos Capacitados*. Buenos Aires: Universidad de Buenos Aires.
- Martínez, D., Toro, E., & Gallego, R. (2008). Algoritmo de recocido simulado aplicado al problema de secuenciamiento de tareas en sistemas de producción lineal flow-shop. *Scientia et Technica no 40*.
- Meng, Q. K. (2009). A Pareto-optimization approach for a fair ramp metering. *Transport Res. Part C*.
- Monarchi, D., Kisiel, C., & Duckstein, L. (Agosto 1973). Interactive multiobjective programming in water resources: a case study. *Water Resources Research*, 837-850.
- Osyczka, A. (1984). *Multicriterion Optimization in Engineering with FORTRAN programs*. Ellis Horwood Limited.
- Pareto, V. (1896). *Cours D'Economie Politique, volume I and II*. Lausanne: F. Rouge.
- Pértega, S. (2001). Complejo Hospitalario Universitario de A Coruña. *Unidad de Epidemiología Clínica y Bioestadística*, 37-41.
- Pilegaard, M. (1998). Metaheuristics for multiple objective combinatorial optimization. *Institute of Mathematical Modelling*.

- Pinnell, C., Drew, D., McCasland, W., & Wattleworth, J. (1967). Evaluation of entrance ramp control on a six-mile freeway section. *Highway Research Record 157*.
- Raidl, G. (2006). A Unified View on Hybrid Metaheuristics. *Springer-Verlag*, 1-12.
- Rao, S. (1986). Game Theory Approach for Multiobjective Structural Optimization. . *Computers and Structures 25*, 119-127.
- Reuben, E. (S.F.). *Recocido Simulado (Simulated Annealing)*. Colombia: Universidad Nacional de Colombia.
- Rudenko, O., Schoenauer, M., Bosio, T., & Fontana, R. (2001). A Multiobjctive Evolutionary Algorithm for Card Front End Design. *Artificial evolution: 5th international conference*, (págs. 205-216). Le Creusot, France.
- Sánchez, G. (2008). *Diseño de controladores lineales mediante algoritmos de optimización multi-objetivo evolutiva*. Caracas: Universidad Simón Bolívar.
- Sarma, G. V., Sellami, L., & Houam, K. D. (1993). Application of Lexicographic Goal Programming in Production Planning—Two case studies. *Opsearch*, 30, 141-162.
- Scariza. (2003). Evaluation of Coordinated and Local Ramp Metering Algorithms using Microscopic Traffic Simulation. *MIT*.
- Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 221-248.
- Stadler, W. (1986). *Initiators of Multicriteria Optimization*. Berlin: Springer-Verlag.
- Talbi, E. (2009). *Metaheuristics: from design to implementation*. New Jersey: John Wiley & Son.
- Van Veldhuizen, D. A. (1999). *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. Air Force Institute of Technology, Department of Electrical and Computer Engineering. Ohio: Wright-Patterson AFB.

- Van Veldhuizen, D. A., & Lamont, G. B. (2000). Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. *Evolutionary Computation*.
- Von Lüken M., C. D. (2003). *Algoritmos Evolutivos para la Optimización Multiobjetivo: Un estudio comparativo en un ambiente paralelo asíncrono*. La Asunción: Universidad Nacional de Asunción.
- Wolpert, D. H., & Macready, W. G. (1997). No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*, 67-82.
- Xevi, E., & Khan, S. (s.f.). *A STEP Method based multiple objective methodology for irrigation water management to model preferences and tradeoffs*.
- Zadeh, L. A. (1963). Optimality and Nonscalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, AC-8, 59-60.
- Zhang, L., & Levinson, D. (2005). Balancing Efficiency and Equity of Ramp Meters ASCE. *Journal of Transportation Engineering* 131(6) , 477-481.
- Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans. Evolutionary Computation*, 257-271.
- Zitzler, E., & Thiele, L. (1999). Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 257-271.