



Universidad Nacional Experimental del Táchira
Vice-Rectorado Académico
Decanato de Postgrado
Maestría en Informática
Trabajo de grado

Algoritmo híbrido para la solución del control de flujo vehicular en autopistas

Autor: Hernández Hernández, Juan Carlos
Cédula de Identidad: V-15.079.470
Teléfonos: 0414-7118579 / 0276-3439079
Correo Electrónico: jhernandez@unet.edu.ve
Tutor: Maldonado Carmona, Javier Alexis
Correo Electrónico: jmaldo@unet.edu.ve

Trabajo de Grado presentado como requisito para optar al Título de Magister en Maestría en Informática

San Cristóbal, 16 de Junio de 2017



UNIVERSIDAD NACIONAL EXPERIMENTAL DEL TÁCHIRA
VICERRECTORADO ACADÉMICO
DECANATO DE POSTGRADO
R.I.F. G-20004886-7



A-760.2017.609



ACTA

En el día de hoy 16 de Junio de 2017, siendo las 10:30 am y estando presentes los ciudadanos **Prof. Javier Alexis Maldonado Carmona**, **Prof. Jose Orlando Figueroa Linares** y **Prof. Jhon Edgar Amaya Salazar**, miembros designados para evaluar el Trabajo de Grado "**ALGORITMO HIBRIDO PARA LA SOLUCIÓN DEL CONTROL DE FLUJO VEHICULAR EN AUTOPISTAS**" presentado por el ciudadano, **Juan Carlos Hernández Hernández**, titular de la Cédula de Identidad **V.-15079470**, para optar al título de **Magister En Informática**, se procedió a la discusión del mismo.

El Jurado decidió en un todo de acuerdo con lo establecido en Artículo 44 de las Normas para el Desarrollo del Componente de Investigación en Programas de Doctorado, Maestría, Especialización y Especialización Técnica de la Universidad Nacional Experimental del Táchira, otorgar el veredicto de APROBADO, en fe de lo cual firmamos.

Prof. Jose Orlando Figueroa Linares
C.I. N° V-15565701
MIEMBRO



Prof. Jhon Edgar Amaya Salazar
C.I. N° V-18565595
MIEMBRO

Prof. Javier Alexis Maldonado Carmona
C.I. N° V-13973642
MIEMBRO-TUTOR

**Autorización para publicación de versión digital del Trabajo de Grado –
Trabajo Especial de Grado**

San Cristóbal, 22 / 06 / 2017

Dirección

Biblioteca Central “Dr. Lorenzo Monroy”

Su Despacho.

El suscrito, **Juan Carlos Hernández Hernández**, titular de la C.I.N° **V-15.079.470** domiciliado en la calle 16 sector Barrio Obrero, San Cristóbal – Estado Táchira. Con el fin de optar al título de Magister en Maestría en Informática, he presentado ante la Biblioteca “Dr. Lorenzo Monroy” de la Universidad Nacional Experimental del Táchira el Trabajo de Grado Titulado: **Algoritmo híbrido para la solución del control de flujo vehicular en autopistas**. Por medio del presente, autorizo por tiempo indefinido a la Biblioteca “Dr. Lorenzo Monroy”, para publicar en forma electrónica y bajo libre acceso el documento digital de mí Trabajo, con el fin de que este pueda ser consultado públicamente vía Internet. Toda persona que consulte ya sea el formato digital podrá copiar partes del texto, **citando siempre la fuente**, es decir el autor, título del trabajo y año.

Esta autorización no implica renunciar a la facultad que tengo de publicar total o parcialmente la obra en otras instancias.

La UNET no será responsable de ninguna reclamación que pudiere surgir de terceros que invoquen autoría de la obra que presento.

La publicación electrónica será:

A Texto completo



Inmediata



Después de 2 años



(Para casos donde se requiera)

Atentamente;

Juan Carlos Hernández H.

C.I.: 15.079.470

DEDICATORIA

Esta tesis está dedicada principalmente a mi madre Nancy Hernández, a mis tías, tíos y primos por sus motivaciones, los cuales fueron determinantes para la culminación del mismo.

En memoria de mi abuela Ana Teresa y mi tía Olga Teresa.

AGRADECIMIENTOS

En primer lugar quiero agradecer de manera especial a Dios y a la Virgen por permitirme concluir con éxito una etapa más en mi vida.

Al tutor Ing. MSc. Javier Alexis Maldonado Carmona por haber aceptado trabajar conmigo, por todo su apoyo y su valiosa orientación en el desarrollo de la Tesis.

A los profesores que formaron parte del jurado, Dr. Jhon Edgar Amaya Salazar y al Ing. MSC. José Orlando Figueroa por sus valiosas aportaciones y comentarios para mejorar la Tesis.

Al Dr. Feijoo Colomine, por su valiosa orientación en el análisis de los resultados obtenidos en la investigación.

ÍNDICE

	Pág.
Dedicatoria	ii
Agradecimientos	iii
Índice.....	iv
Lista de tablas.....	vii
Lista de figuras.....	viii
Resumen.....	x
Introducción	xi

CAPITULO I

EL PROBLEMA

1.1 Planteamiento y formulación del problema.....	1
1.2 Objetivos	4
1.2.1 Objetivo General	4
1.2.2 Objetivos Específicos	4
1.3 Justificación de la Investigación.....	4
1.4 Limitaciones	5

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes de la Investigación	6
2.2 Bases Teóricas	9
2.2.1 Optimización multiobjetivo	9
2.2.1.1 Definición de problemas de optimización multiobjetivo.....	10
2.2.1.2 Óptimo de Pareto	11
2.2.1.3 Dominancia de Pareto	11
2.2.1.4 Frente de Pareto	12
2.2.1.5 Algoritmos para problema de optimización multiobjetivo	12
2.2.1.5.1 Técnicas a priori.....	12
2.2.1.5.1.1 Funciones de agregación lineal	12

	Pág.
2.2.1.5.1.2 Funciones de agregación no lineales.....	13
2.2.1.5.2 Técnicas progresivas	13
2.2.1.5.3 Técnicas a posteriori	13
2.2.1.5.3.1 Muestreo independiente	14
2.2.1.5.3.2 Selección por criterio	14
2.2.1.5.3.3 Selección agregativa	14
2.2.1.5.3.4 Basados en pareto.....	14
2.2.2 Métodos tradicionales de optimización.....	15
2.2.2.1 Método de suma ponderada	15
2.2.2.2 Programación de metas	17
2.2.2.3 Ordenamiento lexicográfico.....	18
2.2.2.4 Restricciones ϵ	19
2.2.2.5 Métodos clásicos	20
2.2.3 Algoritmos Evolutivos Multiobjetivo	22
2.2.4 Técnicas de algoritmos evolutivos multiobjetivo	23
2.2.4.1 Nondominated Sorting Genetic Algorithm (NSGA)	23
2.2.4.2 Strength Pareto Evolutionary Algorithm (SPEA).....	24
2.2.4.3 Nondominated Sorting Genetic Algorithm II (NSGA-II).....	26
2.2.5 Heurísticas.....	26
2.2.5.1 Cuando utilizar el procedimiento heurístico	27
2.2.5.2 Problemas de las técnicas heurísticas.....	29
2.2.6 Metaheurística.....	30
2.2.6.1 Clasificación Metaheurísticas Híbridas	31
2.2.6.2 Algoritmos Genéticos	33
2.2.6.2.1 Operaciones de un algoritmo genético.....	35
2.2.6.3 Recocido simulado.....	37
2.2.6.3.1 Origen.....	37
2.2.6.3.2 Principios del método.....	38

	Pág.
2.2.6.3.3 Algoritmo de recocido simulado.....	39
2.2.6.4 Hill Climbing	42
2.2.6.5 Tabu Search.....	43
 CAPITULO III MARCO METODOLÓGICO	
3.1 Naturaleza de la investigación	45
3.2 Tipo de investigación.....	45
3.3 Diseño de investigación	46
3.4 Fases de la investigación.....	46
 CAPITULO IV ANALISIS E INTERPRETACIÓN DE LOS RESULTADOS	
4.1 Implementación del NSGA II	54
4.2 Implementación del Recocido Simulado	56
4.3 Indicador Hypervolume	63
4.4 Indicador R2.....	64
4.5 Comparación de resultados entre los algoritmos	64
 CAPITULO V CONCLUSIONES	
 CAPITULO VI PROPUESTA	
 REFERENCIAS BIBLIOGRÁFICAS	

LISTA DE TABLAS

	Pág.
Tabla 1: Recursos computacionales utilizado por el algoritmo hibrido y el del autor (Maldonado, 2012) en diferentes ciclos generacionales	63

LISTA DE FIGURAS

	Pág.
Figura 1: Clasificación resumida de las metaheurísticas híbridas	32
Figura 2: Operador de cruce basado en un punto	36
Figura 3: Operador de mutación	36
Figura 4: Pseudocódigo del algoritmo genético simple.....	37
Figura 5: Algoritmo recocido simulado.....	40
Figura 6: Estructura topológica de la autopista, separada en 8 segmentos. (Meng, 2009).....	50
Figura 7: Esquema de una rampa de acceso	52
Figura 8: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 100 generaciones	60
Figura 9: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 500 generaciones	61
Figura 10: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 1000 generaciones..	62
Figura 11: Ejemplo del hipervolumen para un problema de dos objetivos. El área sombreada es el valor del indicador	64
Figura 12: Índice del indicador Hypervolume. Híbrido 100 generaciones contra algoritmo autor (Maldonado, 2012)	65
Figura 13: Índice del indicador R2. Híbrido 100 generaciones contra algoritmo autor (Maldonado, 2012).....	66
Figura 14: Índice del indicador Hypervolume. Híbrido 500 generaciones contra algoritmo autor (Maldonado, 2012)	66
Figura 15: Índice del indicador R2. Híbrido 500 generaciones contra algoritmo autor (Maldonado, 2012)..	67
Figura 16: Índice del indicador Hypervolume. Híbrido 1000 generaciones contra algoritmo autor (Maldonado, 2012)	67
Figura 17: Índice del indicador R2. Híbrido 1000 generaciones contra algoritmo autor (Maldonado, 2012).....	68

Figura 18: Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 100 generaciones	69
Figura 19: Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 500 generaciones	70
Figura 20: Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 1000 generaciones.....	71
Figura 21: Índice del indicador Hypervolume generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)100 generaciones.....	72
Figura 22: Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)100 generaciones	72
Figura 23: Índice del indicador Hypervolume generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)500 generaciones	73
Figura 24: Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)500 generaciones.....	74
Figura 25: Índice del indicador Hypervolume generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)1000 generaciones	74
Figura 26: Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)1000 generaciones	75

Universidad Nacional Experimental del Táchira
Vice-Rectorado Académico
Decanato de Postgrado
Maestría en Informática

Algoritmo híbrido para la solución del control de flujo vehicular en autopistas

Autor: Hernández Hernández, Juan Carlos

Tutor: Ing. MSc. Maldonado Carmona, Javier Alexis

Fecha: Junio 2017

RESUMEN

El trabajo de investigación tiene por objetivo proponer un algoritmo de resolución que optimice el control de acceso (ramp metering) a las autopistas mediante un modelo de algoritmos metaheurísticos híbridos, aplicada a la autopista I-210 W, ubicada en Pasadena, California, en función del comportamiento del sistema vial en estudio. Esta investigación por sus cualidades se considera del tipo descriptiva, tecnológica y cuantitativo no experimental. De la misma manera, para su modelado se realizará una serie de fases basadas en la metodología propuesta por Talbi (2009), la cual abarca desde la formulación hasta la implementación de la solución en los modelos de optimización metaheurísticos. Dado el modelo matemático, se realizará un experimento numérico, mediante el uso de dos algoritmos evolutivos: Non-dominated Sorting Genetic Algorithm II (NSGA-II) y un híbrido, donde se realizarán comparaciones en términos de calidad de los resultados obtenidos, tomando como referencia la maximización del flujo vehicular.

Palabras claves: metaheurística, multiobjetivo, ramp metering, control tráfico vehicular, algoritmos evolutivos, algoritmos híbridos.

INTRODUCCIÓN

Los problemas de optimización multiobjetivo tienen amplias aplicaciones en diversas ramas del conocimiento, en donde el ser humano debe tomar decisiones para resolverlos de manera intuitiva y lograr así un balance entre las distintas condiciones que se plantean. No obstante, existen situaciones en donde no se puede resolver de forma analítica, para ello se ha desarrollado una gran cantidad de algoritmos y métodos para llegar a soluciones más aproximadas. Una alternativa para resolver dichos problemas son los Algoritmos Genéticos y la Metaheurística, las cuales requieren en algunos casos un costo computacional elevado o incluso imposible, es entonces donde aparece la combinación de ambos métodos para generar algoritmos híbridos, produciendo, en términos de rendimiento y calidad, muy buenos resultados en un tiempo razonable, logrando resolver los problemas de una manera aceptable.

Esta investigación consiste en la creación de un algoritmo híbrido, capaz de solucionar el problema de flujo vehicular que se presenta en las rampas de entradas (*ramp metering*) de una autopista, maximizando el rendimiento de la red vial y mejorando así los tiempos de espera de los usuarios de la vía.

En las grandes ciudades del mundo, en las últimas décadas, las vías de alta velocidad han sido diseñadas con el fin de aumentar las tasas del flujo vehicular en los sistemas de transporte vial, debido al crecimiento constante del parque automotor. 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 de las ciudades a las autopistas. Los accesos ocasionalmente se saturan por la afluencia vehicular, la cual puede superar la capacidad de dichas rampas. Teniendo en cuenta esto, se hace cada vez más necesario poder introducir métodos de control de tráfico que permitan un uso óptimo de la infraestructura vial, a fin de disminuir o eliminar la congestión que se produce en estas.

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 que la cantidad de vehículos que entra por la rampa sumados a los que van por la autopista sea mayor a la capacidad de esta, tratando en esta forma de mantener la demanda en su punto óptimo, disminuyendo los tiempos de espera de los usuarios y mejorando el desempeño total del sistema de transporte (Zhang y Levinson, 2005).

Este es un método que se ha usado desde hace tiempo en algunas de las grandes ciudades del mundo. La primera implementación fue hecha en Chicago en la autopista Eisenhower Expressway en el año 1963, (McDermott, Kolenko y Wojcik, 1979). Dado el éxito de la medida, su uso se propagó a lo largo del país. También se ha implementado en otros países, como Inglaterra, Francia, Alemania, Holanda, Japón, Israel y Australia, entre otros, (Richetta, 2002).

CAPITULO I

EL PROBLEMA

1.1 Planteamiento y formulación del problema

La congestión del tránsito vehicular ha ido en aumento en las grandes ciudades del mundo, debido a que el parque automotor ha experimentado un aumento significativo, esto ha ocasionado que se produzcan, tanto en el flujo vehicular de las autopistas como también en las rampas de acceso, una serie de problemas como accidentes viales, incremento en el tiempo del viaje, entre otros (Benavente, 2011).

Para mitigar los problemas generados por los vehículos en los accesos, líneas principales, rampas de ascenso y descenso, presentes en la mayoría de las autopistas de las grandes ciudades del mundo, se ha propuesto implementar para el caso de estudio de la autopista I-210 W en Pasadena, California, USA, (Meng, 2009), técnicas que permitan controlar el flujo vehicular en las rampas de acceso.

Una de las técnicas utilizadas para la solución de problemas de optimización multiobjetivo es la metaheurística, la cual es un procedimiento iterativo de alto nivel para el desarrollo de una heurística que explora eficientemente el espacio de búsqueda con la finalidad de encontrar la solución más óptima (Blum y Roli, 2003). La heurística es una técnica predominante en los problemas de optimización que se puede utilizar como ayuda para buscar la mejor solución a un problema en particular, partiendo de un grupo de soluciones y escogiendo la mejor según los criterios propios de su naturaleza (Reeves, 1993).

De acuerdo con (Laarhoveen y Aarts, 1989), existen diferentes técnicas de búsquedas heurísticas para la resolución de problemas, algunas de ellas son: tabu search, simulated annealing, algoritmo voraz, A*, 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, en la toma de decisiones del siguiente punto de búsqueda, permitiendo la exploración del espacio de soluciones evitando estancarse en óptimos locales (Glover, 1986). El simulated annealing se basa en el lento enfriamiento de una sustancia desde su estado líquido hasta su estado sólido. Durante este proceso sus átomos forman un compuesto sólido puro que corresponde al estado de mínima energía de la sustancia. Luego, la sustancia es calentada hasta alcanzar su estado líquido y gradualmente su temperatura es disminuida hasta alcanzar su estado sólido. La analogía entonces se realiza interpretando una solución del problema de optimización como un estado sólido-líquido en equilibrio, entendiendo que la energía que está siendo minimizada en el proceso termodinámico, corresponde con la función objetivo del problema a resolver (Moins, 2002). El algoritmo voraz trata de elegir la opción óptima en cada paso local, con la esperanza de llegar a una solución general óptima. El nombre “voraz” se debe a que, en cada paso, el algoritmo selecciona el mejor “pedazo” que es capaz de “comer” sin preocuparse de los pasos que restan hasta encontrar la solución (Brassard y Bratley, 2000). La técnica A* es un algoritmo de búsqueda que encuentra la ruta de menor costo entre dos puntos siempre y cuando se cumplan una serie de condiciones. Es ampliamente usado para encontrar rutas y que tan transitable es una gráfica, es decir, se refiere al problema de visitar todos los nodos en una gráfica dada de forma particular, esto no es más que el proceso de trazado de la ruta más eficiente entre unos puntos llamados nodos (Concepción, Soria, y Ruben, 2002).

Según (Goldberg, 1989) los problemas multiobjetivos se pueden resolver mediante el uso de algoritmos genéticos (AG) tales como el NSGA II, SPEA, PAES, PESA II, entre otros. El poder de los AG proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos analíticos o deterministas encuentran dificultades. Si bien no se garantiza que el AG encuentre la solución óptima del problema, existe evidencia empírica de que se encuentran soluciones de un nivel aceptable, en un tiempo competitivo con el resto de algoritmos de optimización combinatoria.

Para (Liu, 2007) este tipo de técnicas híbridas también se le 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. Algunas de las heurísticas utilizadas en combinación con los AG son optimización por colonia de hormigas, synchronous particle local search (SPLS), entre otras (Alonso, 2004).

Bajo este marco referencial, 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 que tenga rampas de acceso.

1.2 Objetivos

1.2.1 Objetivo General

Diseñar una metaheurística híbrida para la solución de un problema de optimización multiobjetivo, en específico para el control del flujo vehicular en autopistas.

1.2.2 Objetivos Específicos

- 1) Revisar metaheurísticas híbridas aplicadas a la solución de un problema de optimización multiobjetivo.
- 2) Analizar el modelo del problema de optimización multiobjetivo del caso de estudio.
- 3) Diseñar la metaheurística híbrida tomando en cuenta las variables del problema de optimización multiobjetivo en estudio.
- 4) Implementar el algoritmo diseñado de acuerdo al modelo del problema propuesto.
- 5) Evaluar los resultados obtenidos en comparación con otro método de optimización empleado.

1.3 Justificación de la Investigación

Debido al aumento del flujo vehicular que se presenta en las ciudades y a la imposibilidad económica en gran parte de los casos de ampliar las vías de circulación, se hace necesario crear sistemas avanzados de control de tráfico que nos permitan llegar

a tiempo y de manera segura al destino. Es por ello que el trabajo de investigación servirá como modelo para la solución de problemas relacionados con la optimización del tráfico automotor en vías de alta demanda vehicular, específicamente en las autopistas, permitiendo hacer un uso eficiente de la capacidad de la vía así como mejorar las condiciones globales de tráfico.

El presente trabajo aportará al campo de las metaheurísticas, utilizadas para los problemas multiobjetivos Pilegaard (1998), concretamente, al comportamiento de los controles instalados en las diferentes rampas de acceso de la autopista. Siendo el algoritmo desarrollado en este trabajo investigativo, una base para estudios futuros en donde se pueda aplicar dichas técnicas a soluciones complejas.

1.4 Limitaciones

1) Para poder encontrar un conjunto de soluciones de óptimo pareto se debe correr el algoritmo de optimización multiobjetivo un determinado número de veces, hasta llegar al punto, en el que al tratar de mejorar uno de los objetivos de las diferentes soluciones encontradas no afecte negativamente al resto de los mismos, cabe mencionar que el proceso de encontrar dichas soluciones tomaría mucho recurso computacional, afectando el tiempo de respuesta debido a la velocidad de procesamiento así como la capacidad de memoria disponible en el equipo.

2) La convergencia prematura es un caso que se puede presentar en los Algoritmos Genéticos, esto ocurre cuando los genes de un individuo relativamente adaptado o bien adaptado en comparación con los demás individuos de la población comienza rápidamente a reproducirse dominando al resto de la misma, esto hace que converja hacia un óptimo eliminando toda posibilidad de poder encontrar mejores soluciones que permitan llegar a una solución óptima o casi óptima.

CAPITULO II

MARCO TEÓRICO

2.1 Antecedentes de la Investigación

Benavente (2011), realizó un análisis sobre el control de acceso en autopistas urbanas específicamente la autopista central de la ciudad de Santiago de Chile, donde estudió las interacciones del tráfico que se producen en torno a rampas de entrada y su impacto en el funcionamiento de las autopistas urbanas, con el propósito de develar las razones por las que se genera congestión.

El trabajo de investigación se fundamentó en un enfoque de tipo analítico y de campo cuyas técnicas de recolección de datos se basaron en videos y en la utilización de espiras magnéticas, permitiendo recolectar la información requerida directamente de la intersección a estudiar, para ello elaboró un algoritmo que reúne varias de las potencialidades del ALINEA, SWARN y ZONE (Zhang, Kim y Nie, 2001), capaz de controlar el flujo que entra por la rampa (*ramp metering*). El trabajo investigativo concluye con la importancia del uso del algoritmo creado como técnica para atacar el problema de exceso de flujo vehicular que se presentaba en la rampa de estudio, moderando la demanda en esta y evitando así la congestión en la autopista, ya que estos afectaban negativamente la capacidad de la misma haciéndola bajar en promedio un 9,8%.

Deternoz y Fernandez (2006) diseñaron un sistema de gestión y monitoreo del tránsito a través de semáforos inteligentes, el objetivo fue el diseño de un sistema de gestión y monitoreo del tránsito, cuya principal funcionalidad fue la de administrar, analizar y almacenar la información proveniente de los dispositivos de interconexión, los cuales se encontraron a su vez conectados al semáforo. Se implementó la metodología de cascada con fases solapadas, los cuales constituyeron cada una de las etapas del ciclo de vida de desarrollo del software desde la investigación preliminar hasta las pruebas del sistema ya que en las fases del modelo se superpusieron, lo que implicó que se pudiera actuar durante las etapas anteriores.

Los resultados obtenidos fueron la elaboración de un módulo que calcula, gestiona y controla las luces de los semáforos automáticamente. Este módulo además, hace que los semáforos vecinos trabajen en conjunto para descongestionar la vía, cualidad importante para la autonomía del sistema. El trabajo de investigación justifica el uso de sensores como controladores de tráfico para la contribución del descongestionamiento de las vías principales e intersecciones trabajando de manera sincronizada con los semáforos vecinos.

Gutierrez y Amati (2012) en el trabajo titulado “Viabilidad del uso de sensores de tráfico en las intersecciones semaforizadas congestionadas del municipio Maracaibo del Estado Zulia”, fue realizado en la escuela de Ingeniería Civil de la Universidad Rafael Urdaneta de Maracaibo, el objetivo de esta investigación fue la de analizar la viabilidad del uso de sensores de tráfico en las intersecciones semaforizadas que realizaban congestiónamiento en el municipio Maracaibo.

De la evaluación de diez intersecciones semaforizadas, se obtuvo como resultado viable la utilización de sensores de tráfico basándose en los valores que arrojaron los módulos de ajuste de volúmenes, flujo de saturación, análisis de capacidad y el de nivel de servicio.

La investigación se fundamentó en un diseño no experimental, transaccional y de campo cuyas técnicas de recolección de datos se basaron en la observación directa y en la utilización de planillas de conteo e inventarios de campo, permitiendo recolectar la información requerida directamente de las intersecciones a estudiar. El siguiente trabajo respalda el uso de este tipo de tecnología, demostrando que es viable su uso ya que puede regular el tráfico a través del control de los semáforos suministrando información en tiempo real del flujo vehicular presente en las intersecciones.

Uribe (2011) investigó sobre un Sistema Adaptativo de Control y Optimización del Tráfico de un Corredor Vial Semaforizado. Aplicación Ciudad de Medellín, el objetivo del trabajo fue determinar una propuesta metodológica con variables claves para desarrollar sobre un corredor vial arterial con tráfico interrumpido un Sistema Adaptativo de Control y Optimización del Tráfico, (SACOT), que permitiera ajustar el funcionamiento de los semáforos en tiempo real de acuerdo con el número de vehículos que llegan a un cruce, medidos en términos de densidad vehicular, y cuyos datos son procesados “On Line”, al realizar diferentes propuestas de temporizaciones y trasladarlas eficazmente a la vía.

Un análisis elemental permitió comprobar que la complejidad del problema aumenta si se trata de determinar el valor de la función objetivo en tiempo real, y por ende también aumenta la complejidad de la solución al operar una red de semáforos con procesos On-Line, en vez de planes con procesos Off – Line, en donde se vincula cada uno de ellos a una situación particular del tráfico con una programación preestablecida, dependiente de la hora y del día de la semana.

Para lograr los resultados se efectuaron simulaciones, con la formulación de un algoritmo propio y específico en lenguaje de programación VAP, se ejecutaron múltiples “corridos” de las cuales se eligieron las 5 de mayor significación en términos de disminuir las demoras promedio en las intersecciones, reducir los tiempos de viaje,

controlar las longitudes de cola, eliminar las deficiencias en la coordinación. Consecuentemente con esto, las variables más sensibles en cada una de las simulaciones estuvieron basadas en el grado de ocupación de los detectores sobre la vía principal y la vía secundaria.

En el trabajo de investigación se hizo necesario diseñar, modelar e implementar una herramienta tecnológica adaptable al medio con el que interactúa, con la finalidad de producir un alivio a la congestión del tráfico urbano presente durante la hora de máxima aglomeración vehicular en una ciudad como Medellín, mejorando de esta forma los indicadores de movilidad dentro del tramo vial particularmente estudiado.

2.2 Bases Teóricas

2.2.1 Optimización multiobjetivo

En la vida siempre se enfrentan problemas en donde se deben tomar decisiones, tratando de llegar a un objetivo. La optimización es una herramienta que facilita dicha tarea. Sin embargo, muchos de los problemas de la vida real envuelven una optimización simultánea de diferentes objetivos en conflicto. Este tipo de problemas es llamado optimización multiobjetivo o vectorial y fue estudiado originalmente en el área económica.

También los científicos e ingenieros se han dado cuenta de que este tipo de problemas se encuentra en todas las áreas de estudio. En el transcurso de los años, se han estado desarrollando técnicas para manejar los problemas de optimización multiobjetivo. Pero no ha sido hasta hace unos 20 años en que investigadores han utilizado el potencial de los Algoritmos Evolutivos (Evolutionary Algorithms) y otras heurísticas basadas en poblaciones (Zitzler, Thiele y Deb, 2000).

Los Algoritmos Evolutivos (EA, por sus siglas en inglés), han demostrado ser especialmente adecuados para la optimización de problemas multiobjetivo, proporcionando 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.

2.2.1.1 Definición de problemas de optimización multiobjetivo

Una gran parte de los problemas del mundo real que se presentan frecuentemente en la vida cotidiana, requieren la búsqueda de soluciones que logren la optimización de varios objetivos de manera simultánea.

En una definición más formal, un problema de optimización multiobjetivo está definido por k diferentes funciones objetivo las cuales deben ser optimizadas de forma simultánea, dado que la modificación de alguna de ellas puede ir en detrimento de otra, afectando de este modo la solución calculada en un momento en particular, lo cual se puede conseguir maximizando las k funciones, minimizándolas o combinando la maximización y la 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 (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 Ω . 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 Ω contienen todas las posibles x

que pueden ser utilizadas para satisfacer una evaluación de $F(x)$. Por lo tanto un problema de optimización multiobjetivo (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 (Sawaragi, Nakayama y Tanino, 1985).

2.2.1.2 Óptimo de Pareto

En optimización multiobjetivo, el término optimizar cambia con respecto a la optimización mono-objetivo (global), pues se trata de encontrar un compromiso entre las distintas funciones objetivo en vez de una sola solución como en optimización global. Así entonces, Pareto (1896) dio, hacia finales del siglo XIX, una definición más formal de óptimo en problemas multiobjetivo, el cual se conoce en la actualidad como óptimo de Pareto. La definición formal es la siguiente: Un vector de variables de decisión $x^* \in F$ (donde F es la zona factible) es un óptimo de Pareto si no existe otro $x \in F$ tal que: $f_i(x) \leq f_i(x^*)$ para toda $i = 1, \dots, k$ y $f_j(x) < f_j(x^*)$ para al menos una j . En otras palabras, “El óptimo de Pareto es aquel vector de variables en el cual no se pueden mejorar las soluciones del problema en una función objetivo sin empeorar cualquiera de las demás” (Abbass, 2002). Desafortunadamente, esto no proporciona una sola solución, sino que se obtiene un conjunto de soluciones llamado Conjunto de Óptimos de Pareto. El conjunto de vectores que corresponden a una solución incluida en el conjunto de óptimos de Pareto son llamados no-dominados.

2.2.1.3 Dominancia de Pareto

El término Dominancia de Pareto puede ser definido de la siguiente manera: Un vector $u = (u_1, \dots, u_k)$ domina a otro vector $v = (v_1, \dots, v_k)$ si y sólo si u es parcialmente menor a v . Es decir que para que una solución domine a otra, ésta necesita ser estrictamente mejor en al menos un objetivo, y no peor en ninguno de ellos.

2.2.1.4 Frente de Pareto

La representación de las funciones objetivo cuyos vectores son no dominados y además están en el conjunto de óptimos de Pareto es llamado el *Frente de Pareto*. La definición formal es la siguiente: Para un problema multiobjetivo dado $f(x)$ y un conjunto de óptimos de Pareto P^* , el frente de Pareto (FP^*) es: $FP^* := \{f = [f_1(x), \dots, f_k(x)] \mid x \in P^*\}$. De forma general, no existe un método eficiente para encontrar el frente de Pareto, y la mejor forma de hacerlo es probar todos y cada uno de los puntos en la zona factible (es decir, enumerar todas las soluciones posibles con una precisión dada). Obviamente, en muchas ocasiones el espacio de búsqueda es tan grande que un proceso enumerativo es incosteable (computacionalmente hablando), y de ahí la necesidad de usar heurísticas como los algoritmos evolutivos para producir aproximaciones del frente de Pareto de un problema.

2.2.1.5 Algoritmos para problema de optimización multiobjetivo

Debido a que la mayoría de los métodos de programación matemática que existen para los problemas de optimización multiobjetivo (MOP) operan sobre un solo individuo a la vez, se necesitan ejecutar en varias ocasiones para poder encontrar el conjunto de óptimos de Pareto de un MOP. En contraste, los algoritmos evolutivos tienen la ventaja de trabajar con una población (o conjunto de soluciones), lo que permite generar varias soluciones no dominadas en una sola ejecución. A continuación se menciona algunas técnicas evolutivas para los problemas de optimización multiobjetivo, dividiéndolas según la clasificación de técnicas multiobjetivo propuesta por (Cohon y Marks, 1975).

2.2.1.5.1 Técnicas a priori

2.2.1.5.1.1 Funciones de agregación lineal

Combinan los resultados de las distintas funciones objetivo en un único valor de aptitud. Este valor de aptitud se obtiene con una combinación lineal de las funciones objetivo. El problema es que se debe de introducir un factor de escalamiento para las funciones objetivo y en la práctica resulta difícil realizar una buena asignación de pesos para las funciones objetivo. Adicionalmente, este esquema no puede generar porciones cóncavas del frente de Pareto sin importar que pesos se usen.

2.2.1.5.1.2 Funciones de agregación no lineales

Es una técnica que utiliza un vector de metas para minimizar la diferencia entre los valores obtenidos. Esta técnica puede generar soluciones en las partes no convexas del frente de Pareto, pero desafortunadamente no son muy eficientes computacionalmente.

2.2.1.5.2 Técnicas progresivas

En estas técnicas, las preferencias se van dando conforme la búsqueda avanza y el tomador de decisiones indica si una solución le parece adecuada o no y el proceso actualiza las preferencias conforme el tomador de decisiones lo va indicando, guiando así el proceso de búsqueda. Existen varias técnicas en investigación de operaciones, pero aún no se han reportado versiones de ellas en computación evolutiva.

2.2.1.5.3 Técnicas a posteriori

En estas técnicas, las preferencias se expresan al final y el tomador de decisiones recibe una información completa de los resultados para así entonces tomar la decisión que mejor le convenga. Es decir, los resultados intentan mostrar todos los compromisos posibles entre las funciones objetivo tratando de generar el verdadero frente de Pareto o al menos una aproximación razonablemente buena.

2.2.1.5.3.1 Muestreo independiente

Consiste en hacer varias ejecuciones de la técnica para encontrar los puntos del frente de Pareto. En este caso, se utilizan diferentes valores de los pesos para las funciones objetivo para generar las diversas porciones del frente de Pareto. El problema de este tipo de enfoque es que se vuelve muy ineficiente para muchas funciones objetivo, pues crece exponencialmente al aumentar el número de combinaciones posibles.

2.2.1.5.3.2 Selección por criterio

Se hacen sub poblaciones y cada una de éstas optimiza una sola función objetivo. Un ejemplo claro de este tipo es el algoritmo de Schaffer llamado VEGA (*Vector Evaluated Genetic Algorithm*) Schaffer y Grefenstette (1985). Un problema de este tipo de técnica es que no se obtienen necesariamente elementos del conjunto de óptimos de Pareto.

2.2.1.5.3.3 Selección agregativa

Se utilizan funciones agregativas para resolver los problemas, con la notable diferencia de que los pesos van cambiando conforme el número de generaciones va cambiando.

2.2.1.5.3.4 Basados en Pareto

La idea fue propuesta por Abbass (2002) y la idea básica es encontrar las soluciones no dominadas y asignar una aptitud con respecto a esta dominancia de Pareto. Estas técnicas, por la importancia que tienen, se discuten en mayor detalle en la sección siguiente.

2.2.2 Métodos tradicionales de optimización

Antes de las técnicas de computación emergente o algoritmos evolutivos, los problemas de optimización multiobjetivo eran resueltos mediante otros 28 métodos apoyados en 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 Lucken, Hermosilla y Barán, 2003).

2.2.2.1 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:

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

Sujeto a:

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

se transforma en un problema de optimización de la forma:

Optimizar

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

sujeto a:

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

donde

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

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 y Haimes (1983) para obtener diferentes soluciones de Pareto Óptimas. Luego de obtener dicho conjunto se presenta al tomador de decisiones 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 tomador de decisiones 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 tomador de decisiones.

2.2.2.2 Programación de metas

Es una técnica de resolución de problemas de optimización de objetivos múltiples, que permite seleccionar las variables que ofrecen una mejor solución al problema planteado, teniendo la gran ventaja que permite trabajar con metas medidas en distintas unidades e incluso contrapuesta.

La filosofía de los problemas de programación meta es muy similar a los de programación lineal, solo que ahora además de las restricciones estructurales, se pueden tener varios objetivos simultáneos, los cuales se desean alcanzar.

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

Es importante recalcar que nunca se sacrificará una meta de mayor prioridad por una de menor prioridad, pero dentro de una misma prioridad, la desviación con mayor ponderación puede ser desplazada por la de menor ponderación, si esta última logra un valor que compense dicha ponderación.

2.2.2.3 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:

Optimizar $f_1(x)$

Sujeta a:

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

Obteniendo su solución x_1^* y $f_1^* = f_1(x_1^*)$. Posteriormente el segundo problema se formula como:

Optimizar $f_2(x)$

Sujeta a:

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

$$f_1(x) = f_1^*$$

la solución a este problema se obtiene como x_2^* y $f_2^* = f_2(x_2^*)$. Este procedimiento se repite hasta que todos los k objetivos han sido considerados, entonces el k -ésimo problema está dado por:

Optimizar $f_k(x)$

Sujeta a:

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

$$f_1(x) = f_1^*$$

·
·
·

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

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.

2.2.2.4 Restricciones ϵ

Está basado en la minimización de una función objetivo, considera las otras funciones objetivo como restricciones límite con niveles permisibles ϵ_j . Se minimiza la función objetivo más relevante, con restricciones sobre las otras funciones objetivo, y entonces los niveles ϵ_j son modificados para generar el conjunto completo de soluciones no dominadas. Los pasos son los siguientes:

1) Encontrar el mínimo de la función objetivo j -ésima, buscar x^* tal que:

$$f_j(x^*) = \min_{x \in F} f_j(x)$$

sujeto a restricciones adicionales de la forma:

$$f_i(x) \leq \epsilon_i \text{ para } i = 1, 2, \dots, k \text{ e } i \neq j$$

donde ϵ_i son valores que se asumen de las funciones objetivo, los cuales no deben ser superados.

2) Repetir el paso anterior para diferentes valores de ϵ_i . La información derivada de un conjunto bien seleccionado de ϵ_i puede ser útil en la toma de decisión. La búsqueda se detiene cuando quien toma las decisiones encuentra una solución satisfactoria. Puede ser necesario repetir el procedimiento anterior para diferentes índices de j . Para obtener valores ϵ_i adecuados se utiliza optimización con un único objetivo para cada función objetivo.

2.2.2.5 Métodos clásicos

Se les llama así a los métodos de solución de problemas de optimización multiobjetivo desarrollados inicialmente que no implementan algoritmos evolutivos. Éstos combinan de alguna forma los diferentes objetivos en uno único, lo que implica la necesidad de tener conocimiento profundo del dominio del problema que permita realizar la escalarización de forma correcta, conocimiento con el cual en general no se cuenta. Además de ello en este tipo de aplicaciones se ha identificado otras dificultades o desventajas:

- 1) Alto costo en términos computacionales para delinear el frente de pareto, debido a que el hallar un conjunto pareto optimo implica correr varias veces el algoritmo, ya que encuentran una única solución por corrida.

- 2) Alta sensibilidad de algunos algoritmos a los parámetros del algoritmo como los pesos, el orden de evaluación o el valor de las restricciones.
- 3) Sensibilidad de algunos algoritmos a la forma del frente de Pareto.
- 4) Sensibilidad ante problemas que involucran el azar o incertidumbre.
- 5) La eficiencia en la diversidad de las soluciones depende del optimizador no siempre llevando a resultados satisfactorios.
- 6) La convergencia hacia una solución óptima depende de la solución inicial elegida.
- 7) La mayoría de estos algoritmos tienden a atascarse en soluciones locales o subóptimas.
- 8) Un algoritmo que es eficiente para resolver un problema de optimización puede no serlo resolviendo uno diferente.
- 9) Estos algoritmos no son eficientes manejando problemas que tengan un espacio de búsqueda discreto.

Estas dificultades y otras pueden ser superadas con el uso de los Algoritmos Evolutivos. Los métodos clásicos para optimización multiobjetivo operan en dos fases: inicialmente generan a partir del problema multiobjetivo un problema de objetivo único, luego en la segunda fase se aplica un método de optimización tradicional para obtener soluciones.

Para la primera fase se utilizan diversos métodos, cada uno con sus propias ventajas y desventajas, en ellos, en general la técnica consiste en disponer las funciones objetivo de tal forma que se puedan unir en una sola función parametrizada cuyos parámetros variarán de modo sistemático para cada corrida, lo que concluye en múltiples corridas del algoritmo para encontrar un conjunto que aproxime al conjunto pareto óptimo. Entre estas técnicas se encuentran el método de la suma de pesos, el método de las restricciones, la programación con base a objetivos, el ordenamiento lexicográfico, entre otros. Para la segunda fase se requieren múltiples corridas del proceso de optimización para conseguir un conjunto que aproxime al conjunto pareto óptimo, entre ellos están programación lineal, búsqueda exhaustiva, optimización mediante gradiente, inteligencia artificial, entre otras.

2.2.3 Algoritmos Evolutivos Multiobjetivo

Los Algoritmos Evolutivos (EAs) se han popularizado como métodos robustos y efectivos para la resolución de problemas de optimización. Tradicionalmente, los problemas abordados consideraban la optimización de una única función objetivo, pero en la última década se han desarrollado una amplia gama de EAs para afrontar problemas con objetivos múltiples. Estos cuentan con complejidades propias que los distinguen de los problemas de un solo objetivo, y por ello los EAs para optimización multiobjetivo tienen características que los diferencian de los EAs tradicionales (Coello, Van Veldhuizen y Lamont, 2007).

De acuerdo con Coello, Van Veldhuizen y Lamont (2007), la capacidad de los EAs para resolver problemas con múltiples objetivos fue sugerida en la década de 1960 por Rosenberg, pero hasta mediados de la década de 1980 no se presentó la implementación de un algoritmo evolutivo para optimización multiobjetivo (MOEA) Schaffer y Grefenstette (1985). A partir de la década de 1990 fueron realizadas una gran cantidad

de propuestas de MOEAs, formándose una comunidad de investigadores en el área que continúan trabajando activamente en la actualidad.

Dado que trabajan en paralelo sobre un conjunto de soluciones, los EAs tienen la potencialidad de tratar problemas con objetivos múltiples, hallando en cada ejecución un conjunto de soluciones aproximadas al frente de Pareto. Esto representa una importante ventaja respecto a los algoritmos tradicionales, que solamente generan una solución por ejecución. Complementariamente, los EAs tienen otras ventajas respecto a los algoritmos tradicionales, como ser menos sensibles a la forma o a la continuidad del frente de Pareto o permitir abordar problemas con espacio de soluciones de gran dimensión (Nesmachnow, 2004).

Un MOEA debe diseñarse para lograr dos propósitos en forma simultánea: lograr buenas aproximaciones al frente de Pareto y mantener la diversidad de las soluciones, de modo de muestrear adecuadamente el espacio de soluciones y no converger a una solución única o a una sección acotada del frente. El mecanismo de los EAs permite lograr el primer propósito, mientras que para preservar la diversidad los MOEAs utilizan las técnicas de nichos, sharing, crowding o similares, utilizadas tradicionalmente por los EAs en la optimización de funciones multimodales (Nesmachnow, 2004).

2.2.4 Técnicas de algoritmos evolutivos multiobjetivo

2.2.4.1 Nondominated Sorting Genetic Algorithm (NSGA)

Srinivas y Deb (1994) propusieron el algoritmo genético de ordenación no dominada (Non-dominated Sorting in Genetic Algorithm, NSGA). Su funcionamiento se basa en el uso de varias capas de clasificación de los individuos. Antes de efectuar la selección, la población es jerarquizada en base a la no dominación, de forma que

todos los individuos no dominados se clasifican en una misma categoría con un valor arbitrario de aptitud. Para mantener la diversidad de la población se lleva a cabo una repartición de aptitud entre estos individuos clasificados usando los valores arbitrarios de aptitud previamente definidos. Posteriormente, este grupo de individuos clasificados se ignora, y se evalúa otra capa de individuos no dominados. Este proceso se repite hasta que todos los individuos de la población están clasificados.

Puesto que los individuos en el primer frente tienen el máximo valor de aptitud, siempre obtienen más copias que el resto de la población, lo que conlleva a una rápida convergencia de la población hacia las regiones no dominadas. La repartición de aptitud ayuda a distribuir los individuos sobre esta región. La eficiencia de NSGA yace en la forma en que los objetivos múltiples de un problema se reducen a una función de aptitud arbitraria usando un procedimiento de ordenamiento basado en no dominancia. La asignación de aptitud se lleva a cabo en base a los valores de los parámetros en lugar de considerar los valores de las funciones objetivo, con lo cual se persigue asegurar que se lleve a cabo una mejor distribución de los individuos, así como permitir que existan múltiples soluciones equivalentes. Sin embargo, esta técnica es más ineficiente que MOGA, tanto computacionalmente como en términos de la calidad de los frentes de Pareto que produce, y además es más sensible al factor de repartición de aptitud (Coello, 2000).

2.2.4.2 Strength Pareto Evolutionary Algorithm (SPEA)

El algoritmo evolutivo de fuerza de Pareto (Strength Pareto Evolutionary Algorithm, SPEA) se incluye dentro de los MOEAs de segunda generación, se destaca por el uso del elitismo, bien mediante una estrategia de selección o por el uso de una población secundaria Laumanns, Zitzler y Thiele (2001). Al igual que otras técnicas de optimización multiobjetivo, SPEA utiliza el concepto de dominancia de Pareto para asignar la función de aptitud a los individuos, almacena las soluciones no dominadas

externamente (elitismo), y utiliza un método de clustering para reducir el número de soluciones no dominadas almacenadas en el conjunto externo sin destruir las características del frente.

En particular, SPEA resultaba innovador por los siguientes aspectos: combina los aspectos anteriores a un único algoritmo, la aptitud de un individuo viene determinada únicamente por las soluciones almacenadas en el conjunto externo siendo irrelevante el hecho de que un miembro de la población domine a otro, todas las soluciones del conjunto externo participan en el proceso de selección, incluye un nuevo método de nichos para preservar la diversidad de la población, el cual no requiere de ningún parámetro de distancia (Zitzler, Thiele y Deb, 2000).

SPEA resultó ser un algoritmo muy influyente ya que, además de incluir los aspectos mencionados anteriormente, mejoraba el rendimiento de los utilizados hasta la época, en gran medida gracias a que era el único método que hacía uso explícito del concepto elitismo. Sin embargo también presentaba sus principales debilidades, entre las que destacaba el hecho de que al momento de realizar la asignación de aptitud, aquellos individuos que eran dominados por los mismos miembros del archivo tenían el mismo valor de aptitud. Esto significa que en el caso en el que el archivo contenga un solo individuo, todos los miembros de la población tienen el mismo ranking, independientemente de si se dominan entre ellos o no. Como consecuencia, la presión selectiva se ve rebajada sustancialmente, y en este caso particular, SPEA puede llegar a comportarse como un algoritmo de búsqueda aleatorio Baños (2006).

2.2.4.3 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 y 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: alto costo computacional del procedimiento de clasificación de las soluciones no dominadas, falta de elitismo, necesidad de especificar un parámetro de sharing (share).

Por esta razón Deb, Agrawal, Pratap y Meyarivan (2000), propusieron NSGA-II, el cual ha conseguido resultados muy buenos en numerosos estudios con diferentes problemas de test y métricas de rendimiento. En NSGA-II se propone un nuevo método de comparación crowded que resuelve los puntos débiles del NSGA. Se puede decir que NSGA-II utiliza repartición de aptitud, sin embargo cuando se agregan frentes rebasamos el tamaño máximo permitido, eliminando las soluciones peor diversificadas del ultimo frente. Por lo tanto, desde este punto de vista estamos realizando un procedimiento de aclarado, el cual está supeditado a la relación de dominancia como se extrae de la definición de nuestro operador de comparación. Finalmente, al igual que SPEA presentaba un funcionamiento híbrido entre fitness-sharing y crowding, el NSGA II utiliza como mecanismo de niching una combinación de fitness-sharing y clearing (Mahfoud, 1995).

2.2.5 Heurísticas

La idea más general del término *heurística* (proviene del griego Heurískein y puede traducirse como encontrar, hallar) Melián, Moreno y Moreno (2003), está relacionada con la tarea de resolver inteligentemente problemas reales usando el conocimiento disponible. Una definición intuitiva es la propuesta por (Zanakis y Evans, 1981):

“Las heurísticas son procedimientos simples basados en el sentido común que se supone que obtendrán una buena solución (no necesariamente la mejor) a un problema difícil de un modo sencillo y rápido”.

En Investigación de Operaciones, una heurística es una técnica de resolución de problemas de optimización (conformada por una regla o un conjunto de reglas) que busca buenas soluciones a un costo computacional razonable, sin garantizar la optimalidad. Una característica de los métodos heurísticos es que pueden encontrar una solución de alta calidad pero que no necesariamente es la óptima, e incluso, en muchos casos no se llega a establecer lo cerca que está una solución heurística de la solución óptima.

En una primera aproximación puede parecer que existen serios inconvenientes relacionados al uso de los métodos heurísticos, sin embargo, es necesario enfatizar que muchas técnicas heurísticas brindan soluciones muy buenas en la práctica. Cabe destacar el interés creciente por el estudio y aplicación de procedimientos heurísticos en Investigación de Operaciones, pasando de ser considerados pobres herramientas a instrumentos fundamentales y, en muchos casos, imprescindibles para la resolución práctica de un problema.

2.2.5.1 Cuando utilizar el procedimiento heurístico

El uso de un procedimiento heurístico en lugar de un procedimiento exacto para resolver un problema está especialmente indicado cuando:

- 1) El problema es de una naturaleza tal que no se conoce algún método exacto para su resolución.

- 2) Aunque existe un método exacto para resolver el problema, su uso es computacionalmente muy costoso.
- 3) Se prefiere abordar, por medio de heurísticas, una representación más ajustada del problema planteado que una versión menos realista de tal problema que pueda resolverse de forma exacta. Es decir, se prefiere resolver de forma aproximada un modelo ajustado a la realidad que resolver de forma exacta un modelo aproximado de la realidad. En este sentido, los métodos heurísticos son más flexibles que los métodos exactos, permitiendo la incorporación de condiciones de difícil modelización.
- 4) Se desea aumentar la eficiencia de un procedimiento exacto. En ese caso un método heurístico puede proporcionar una buena solución inicial de partida o participa en un paso intermedio del procedimiento, como por ejemplo, las reglas de selección de variables de entrada en el método simplex, o el establecimiento de cotas en el algoritmo de ramificación y acotamiento.
- 5) Se tiene que resolver repetidas veces un mismo problema, probablemente con datos distintos. En estas circunstancias se requiere un procedimiento eficiente de resolución que suministre de forma rápida una solución suficientemente buena. Estas situaciones se presentan, por ejemplo, al tener que determinar diariamente valores de las variables de interés.

2.2.5.2 Problemas de las técnicas heurísticas

A pesar de que las heurísticas son una alternativa excelente para resolver problemas de difícil solución, éstas también adolecen de ciertas propiedades deseables. Un inconveniente en la gran mayoría de los métodos heurísticos es su dependencia de la estructura del problema para el cual fue diseñado, y su falta de habilidad para adaptarse a nuevas situaciones o modificaciones del problema de partida. Así, usan propiedades de la región factible y/o de la función objetivo que hacen que los procedimientos sean válidos sólo bajo esas condiciones.

A diferencia de los métodos exactos, no existe un procedimiento conciso y preestablecido, independiente del problema. En los métodos heurísticos las técnicas e ideas aplicadas a la resolución de un problema son específicas de éste y aunque, en general, pueden ser trasladadas a otros problemas, han de ser particulares en cada caso.

Otro problema importante de las heurísticas es su incapacidad para escapar de óptimos locales. Una solución heurística para un problema podría corresponder o estar muy cercana a un óptimo local y no a un óptimo global, ya que estos algoritmos pueden quedarse atrapados en una zona del espacio de soluciones.

Para superar los inconvenientes que presentan los métodos heurísticos clásicos, durante los últimos años diferentes investigadores han estado trabajando en un tipo de procedimientos heurísticos generales que pueden usarse para resolver una amplia variedad de problemas, adaptando convenientemente los elementos que los definen y que permiten obtener mejores resultados. Tales procedimientos son las metaheurísticas.

2.2.6 Metaheurística

Las metaheurísticas son estrategias inteligentes para diseñar o mejorar procedimientos heurísticos generales con un alto rendimiento. El término fue introducido por Glover (1986) en el artículo que presenta las ideas básicas de búsqueda tabú. A partir de entonces, han surgido multitud de propuestas para diseñar buenos procedimientos para resolver ciertos problemas que, al ampliar su campo de aplicación, han adoptado la denominación de metaheurísticas.

Una definición más formal introducida por Voß (2001) se muestra a continuación: “Una metaheurística es un proceso iterativo que dirige y modifica las operaciones de otras heurísticas subordinadas para producir soluciones de alta calidad. Puede manipular una solución única (completa o incompleta) o un conjunto de ellas en cada iteración. El heurístico subordinado puede ser un procedimiento de alto (bajo) nivel, una simple búsqueda local o un método de construcción”.

De acuerdo a esta definición, las metaheurísticas se sitúan conceptualmente por encima de las heurísticas en el sentido que guían el diseño de éstas. Así, al abordar un problema de optimización, se puede escoger cualquier metaheurística para diseñar un algoritmo específico que lo resuelva aproximadamente (Martí, 2003).

Las metaheurísticas han tenido un gran desarrollo y crecimiento desde sus comienzos en los años 80. En la práctica, han tenido éxito en una gran variedad de problemas de optimización combinatoria difíciles, siendo en muchos casos la única alternativa factible para encontrar una solución de calidad en un tiempo razonable. En general, las metaheurísticas se comportan como métodos robustos y eficientes, por lo que hoy en día constituyen un área importante de investigación y aplicación.

2.2.6.1 Clasificación Metaheurísticas Híbridas

La idea de hibridar metaheurísticas no es nueva, se remonta a los orígenes mismos de las metaheurísticas. La principal motivación para hibridar algoritmos es obtener técnicas con mejor desempeño que aprovechen y unan las ventajas de estrategias purista de forma sinérgica.

Sin embargo tales híbridos no eran tan populares por el hecho que en la comunidad de investigadores defendían, diferían e incluso competían entre sí sobre cuál técnica metaheurística era su favorita y “mejor” a las demás.

Esta situación cambió radicalmente gracias al teorema del “No Free Lunch” (Wolpert y Macready, 1997), que establece que no puede haber una estrategia de optimización general que sea globalmente mejor que otra, esto es, que para cualquier algoritmo, cualquier rendimiento elevado sobre una sola clase de problemas se paga exactamente en el rendimiento en otra clase.

Torres (2010) establece en su tesis doctoral citando a Raidl (2006) que actualmente, se recomienda la hibridación de técnicas heurísticas porque está probado que producen mejores soluciones que los mecanismos puristas en la resolución de problemas NP-completos.

En cuanto a la clasificación de las técnicas han surgido varias propuestas de diferentes autores Cotta-Porras (1998); El-Abd y Kamel (2005); Talbi (2002) pero Raidl (2006) unificó dichas clasificaciones y taxonomías propuestas por los autores citados anteriormente, las cuales se pueden observar en la Figura 1.

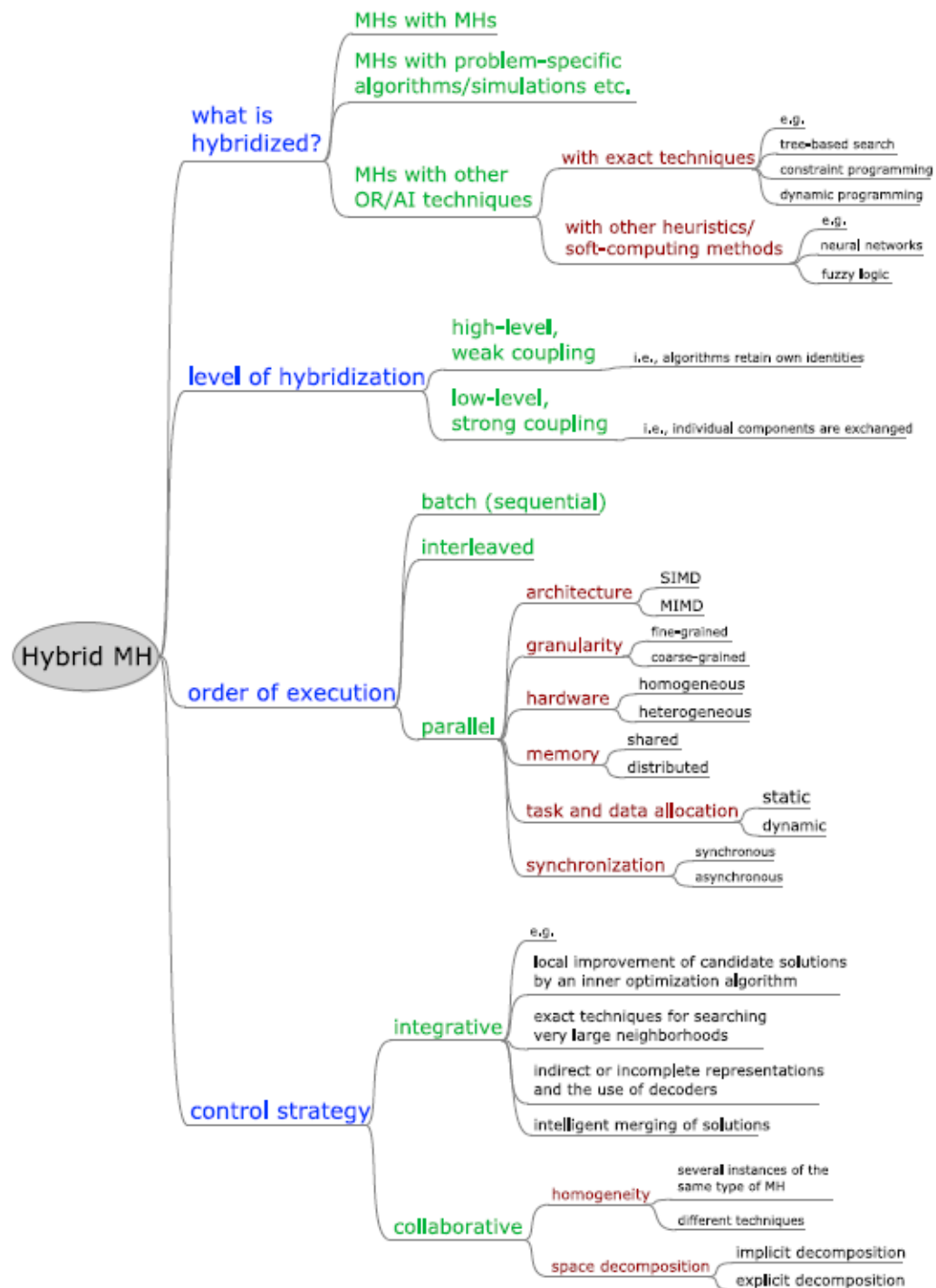


Figura 1 Clasificación resumida de las metaheurísticas híbridas

De las diferentes propiedades de los algoritmos metaheurísticos mostradas en la figura anterior, se encuentra la estrategia de control, la cual se divide en dos categorías:

- 1) Integrativa: es muy popular, consiste en que uno de los algoritmos considerado maestro incluye uno o más componentes subordinados a través de un llamado o incrustado dentro del algoritmo principal.
- 2) Colaborativa o cooperativa: consiste en el intercambio de información por parte de los algoritmos de manera paralela, pero no son parte del otro. Los algoritmos de búsqueda cooperativa consisten en la ejecución paralela de algoritmos de búsqueda que pueden ser diferentes o instancias del mismo algoritmo que trabajan en diferentes modelos o que funcionan con diferentes configuraciones de parámetros.

Por lo tanto, la estrategia de control en metaheurísticas híbridas puede ser integradora o colaborativa, y el orden de ejecutar las partes combinadas puede ser secuencial, paralelo o intercalado.

2.2.6.2 Algoritmos Genéticos

Los algoritmos genéticos (siglas en inglés GA) son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza acorde con los principios de la selección natural y la supervivencia de los más fuertes, postulados por Darwin (1859). Por imitación de este proceso, los GA son capaces de ir creando soluciones para problemas del mundo real. La evolución de dichas soluciones hacia valores óptimos del problema depende en buena medida de una adecuada codificación de las mismas.

En la naturaleza los individuos de una población compiten entre sí en la búsqueda de recursos tales como agua, comida y refugio, incluso los miembros de una misma especie compiten a menudo en la búsqueda de un compañero. Aquellos individuos que tienen más éxito en sobrevivir y en atraer compañeros tienen mayor probabilidad de generar gran número de descendientes. Por el contrario individuos poco dotados producirán un menor número de descendientes. Esto significa que los genes de los individuos mejor adaptados se propagarán en sucesivas generaciones hacia un número de individuos creciente. La combinación de buenas características provenientes de diferentes ancestros, puede a veces producir descendientes “superindividuos”, cuya adaptación es mucho mayor que la de cualquiera de sus ancestros. De esta manera, las especies evolucionan logrando características cada vez mejor adaptadas al entorno en el que viven.

Los GA usan una analogía directa con el comportamiento natural. Trabajan con una población de individuos, cada uno de los cuales representa una solución factible a un problema dado. A cada individuo se le asigna un valor o puntuación, relacionado con la bondad de dicha solución. En la naturaleza esto equivale al grado de efectividad de un organismo para competir por unos determinados recursos. Cuanto mayor sea la adaptación de un individuo al problema, mayor será la probabilidad de que el mismo sea seleccionado para reproducirse, cruzando su material genético con otro individuo seleccionado de igual forma. Este cruce producirá nuevos individuos descendientes de los anteriores, los cuales comparten algunas características de sus padres. Cuanto menor sea la adaptación de un individuo, menor será la probabilidad de que dicho individuo sea seleccionado para la reproducción, y por tanto de que su material genético se propague en sucesivas generaciones.

De esta manera se produce una nueva población de posibles soluciones, la cual reemplaza a la anterior y verifica la interesante propiedad del que contiene una mayor proporción de buenas características en comparación con la población anterior. Así a

lo largo de las generaciones las buenas características se propagan a través de la población. Favoreciendo el cruce de los individuos mejor adaptados, van siendo exploradas las áreas más prometedoras del espacio de búsqueda. Si el GA ha sido bien diseñado, la población convergerá hacia una solución óptima.

El poder de los GA proviene del hecho de que se trata de una técnica robusta, y pueden tratar con éxito una gran variedad de problemas provenientes de diferentes áreas, incluyendo aquellos en los que otros métodos encuentran dificultades. Si bien los GA no garantizan encuentre la solución óptima al problema, existen evidencias de que se encuentran soluciones a un nivel aceptable, en un tiempo competitivo en comparación con el resto de algoritmos de optimización combinatoria. En el caso que existan técnicas especializadas para resolver un determinado problema, lo más probable es que superen al GA, tanto en rapidez como en eficacia. El gran campo de aplicación de los GA se relaciona con aquellos problemas para los cuales no existen técnicas especializadas, aun en el caso de que dichas técnicas existan, y funcionen perfectamente, pueden efectuarse mejoras de las mismas hibridándolas con los GA.

2.2.6.2.1 Operaciones de un algoritmo genético

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:

- 1) Selección: se encarga de hacer una selección con los individuos más capacitados para que éstos sean los que se reproduzcan con más probabilidad de acuerdo con la teoría de Darwin en la cual los más aptos son los que deben sobrevivir y crear una nueva descendencia con más facultades.

- 2) Cruce: una vez realizada la selección de los cromosomas se procede a realizar la reproducción o cruce entre dos individuos, para crear una generación o conjunto de soluciones posibles, logrando de esta manera que el descendiente mejore la aptitud de sus padres.

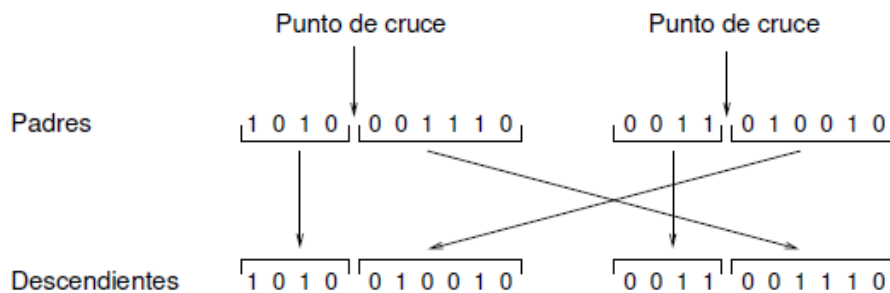


Figura 2 Operador de cruce basado en un punto

- 3) Copia: consiste en pasar a la siguiente generación un grupo de individuos sin ningún tipo de variación.
- 4) Mutación: es un operador que produce cambios aleatorios en varios cromosomas. Una forma simple de realizar la mutación deberá ser alterar uno o más genes. En un algoritmo genético, la mutación cumple el rol de reposición de genes perdidos en la población durante el proceso de selección y de provisión de aquellos genes que no están presentes en la población inicial.



Figura 3 Operador de mutación

```

BEGIN /* Algoritmo Genetico Simple */
  Generar una poblacion inicial.
  Computar la funcion de evaluacion de cada individuo.
  WHILE NOT Terminado DO
    BEGIN /* Producir nueva generacion */
      FOR Tamaño poblacion/2 DO
        BEGIN /*Ciclo Reproductivo */
          Seleccionar dos individuos de la anterior generacion,
          para el cruce (probabilidad de seleccion proporcional
          a la funcion de evaluacion del individuo).
          Cruzar con cierta probabilidad los dos
          individuos obteniendo dos descendientes.
          Mutar los dos descendientes con cierta probabilidad.
          Computar la funcion de evaluacion de los dos
          descendientes mutados.
          Insertar los dos descendientes mutados en la nueva generacion.
        END
      IF la poblacion ha convergido THEN
        Terminado := TRUE
    END
  END

```

Figura 4 Pseudocódigo del algoritmo genético simple

2.2.6.2 Recocido Simulado

2.2.6.2.1 Origen

La técnica heurística del recocido simulado (Simulated Annealing), está basada en un algoritmo propuesto por (Metropolis, Rosenbluth, Rosenbluth, Teller y Teller, 1953) en el marco de la termodinámica estadística, para simular el proceso de enfriamiento de un material (recocido). (Kirkpatrick, Gelatt Jr. y Vecchi, 1983) establecieron una analogía entre el proceso de recocido y el reto de resolver problemas de optimización combinatoria de gran escala. Desde entonces el recocido simulado ha sido utilizado para resolver en forma exitosa una amplia variedad de problemas de optimización combinatoria, convirtiéndose en una metaheurística clásica.

2.2.6.2.2 Principios del método

El proceso de recocido es una estrategia para modificar el estado de un material y alcanzar un estado óptimo mediante el control de la temperatura. El recocido comienza con el calentamiento del material a una alta temperatura, para luego enfriarlo lentamente, manteniendo en cada etapa una temperatura por cierto tiempo. Si la disminución de la temperatura es demasiado rápida, pueden originarse defectos en el material. Esta técnica de disminución controlada de la temperatura conduce a un estado sólido cristalizado, el cual es un estado estable y que corresponde a un mínimo absoluto de energía Dréo, Pétrowski, Siarry y Taillard (2006). Metropolis, Rosenbluth, Rosenbluth, Teller y Teller (1953) modelaron el proceso de recocido simulando los cambios energéticos en un sistema de partículas conforme decrece la temperatura, hasta que converge a un estado estable. De acuerdo a las leyes de termodinámica, la probabilidad de un incremento de energía ΔE a una temperatura t se puede aproximar por:

$$P(\Delta E) = e^{-(\Delta E / kt)}$$

Donde K se denomina constante de Boltzman.

En el algoritmo de Metropolis se genera una modificación en el sistema y se calculan los cambios de energía resultantes; si esta transformación origina una disminución energética, se acepta la modificación, y si ocurre un incremento de energía, el cambio será aceptado con una probabilidad dada por la formula mencionada anteriormente. A una temperatura alta la probabilidad $P(\Delta E)$ es cercana a 1, por lo que la mayoría de los cambios generados en el sistema son aceptados; a una temperatura baja $P(\Delta E)$ es cercana a 0 y la mayoría de los cambios son rechazados (Dréo, Pétrowski, Siarry y Taillard, 2006).

El método recocido simulado transpone el proceso de recocido a la solución de un problema de optimización, la función objetivo del problema, similar a la energía del material, es minimizada con la ayuda de una temperatura ficticia, la cual es un parámetro de control del algoritmo. Este parámetro debe tener el mismo efecto que la temperatura del sistema físico: conducir hacia el estado óptimo. Si la temperatura es disminuida gradualmente y de manera controlada se puede alcanzar el mínimo global, si es disminuida abruptamente se puede llegar a un mínimo local.

2.2.6.2.3 Algoritmo de recocido simulado

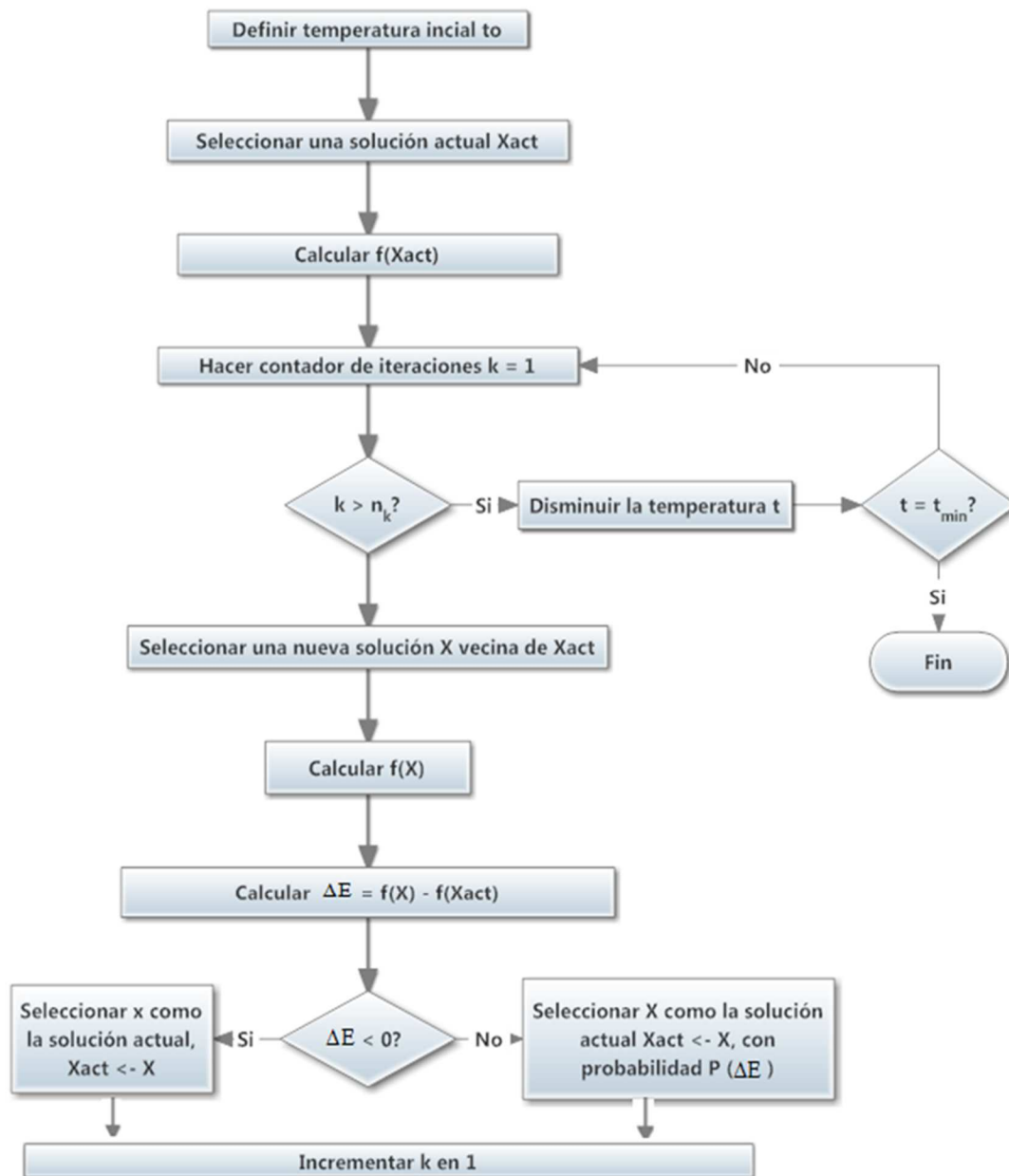


Figura 5 Algoritmo recocido simulado

El algoritmo de recocido simulado parte de una solución factible X_{act} que puede ser generada en forma aleatoria y se evalúa la función de costo (función objetivo) para esa solución, llámese a este valor $f(X_{act})$. Se comienza con una temperatura ficticia alta y se genera una nueva solución X que corresponde a una solución vecina de X_{act} , para la cual se calcula el valor asociado de la función de costo $f(X)$. Además debe calcularse la diferencia ΔE en la función de costo entre ambas soluciones:

$$\Delta E = f(X) - f(X_{act})$$

Si $\Delta E < 0$ (el costo de la nueva solución X es menor al costo de la solución actual X_{act}) la nueva solución es aceptada, es decir, que una solución de menor costo siempre se acepta. En caso contrario, la nueva solución es aceptada con una probabilidad $P(\Delta E)$:

$$P(\Delta E) = e^{-(\Delta E / t)}$$

En la práctica se selecciona un número aleatorio entre 0 y 1, y si este número es menor que $P(\Delta E)$ la nueva solución se acepta.

Aceptar una solución de menor calidad permite salir de un posible mínimo local y explorar otras áreas del espacio de soluciones. Como la simulación comienza con una temperatura alta, $P(\Delta E)$ es cercana a 1 y por lo tanto una nueva solución con un costo mayor tiene una alta probabilidad de ser aceptada. La probabilidad de aceptar una solución peor va disminuyendo a medida que la temperatura decrece.

Para cada nivel de temperatura, el sistema debe alcanzar un equilibrio, es decir, un número de nuevas soluciones debe ser ensayado antes de que la temperatura sea

reducida. Se puede mostrar que el algoritmo encontrará, bajo ciertas condiciones el mínimo global y no se estancará en un mínimo local (Moins, 2002).

El bucle interno genera una nueva solución en el entorno de la solución actual y la acepta si es mejor, en el caso de que esta nueva solución sea de menor calidad se acepta con una probabilidad $P(\Delta E)$. Este proceso se repite N veces. Cuando este ciclo iterativo se completa, la temperatura se disminuye y comienza nuevamente el bucle interno, repitiendo el proceso de creación, evaluación y posible aceptación de soluciones vecinas. Cuando la temperatura es lo suficientemente baja ($t = t_{min}$) el algoritmo finaliza y debe indicar cuál fue la mejor solución obtenida.

2.2.6.3 Hill Climbing

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 seleccionando 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 y Company, 2000).

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:

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

2.2.6.4 Tabu Search

El método Tabu Search (TS) tiene su origen en procedimientos combinatorios aplicados a problemas de curvatura no lineal desarrollados al final de la década de los 70's, aunque muchos autores le dan el crédito inicial a Fred Glover y Pierre Hansen. (Glover y Laguna, 1993) ya que son los primeros autores en desarrollar la técnica en todo su potencial hacia el segundo quinquenio de la década de los 80's, ellos utilizaron

ésta técnica para resolver problemas de optimización a gran escala. Este método se ha aplicado para resolver problemas de programación de grupos de trabajo donde se requiere de una secuencia de pasos para obtener un producto determinado, y el objetivo principal es obtener el producto en el menor tiempo posible, así que se desea "descubrir" cuál es la distribución del trabajo en los grupos de tal manera que el tiempo sea el menor, cumpliendo las normas de calidad del producto.

Este método heurístico está diseñado para escapar de la optimalidad local, basado en el manejo y uso de una colección de principios que sirven para resolver el problema de manera "inteligente", esto es, haciendo uso de memoria flexible para involucrar dos procesos, el de la adquisición y el de mejoramiento de la información; así, al tener cierta "historia" de los caminos ya recorridos y de los óptimos encontrados, se puede evitar permanecer en las mismas regiones, y recorrer regiones nuevas para encontrar otras mejores soluciones.

El método de búsqueda tabú se basa en tres puntos principales:

- 1) El uso de memorias diseñadas para permitir evaluar la información de búsqueda histórica.
- 2) Un mecanismo de memoria que restringe y libera el proceso de búsqueda.
- 3) La utilización de memorias de diferentes lapsos de tiempo: la de término corto, la de término intermedio y la de término largo, para guardar (por un tiempo) aquellas características que lograron una buena solución, y olvidando otras (ya 37 transcurrido el tiempo de memoria) para permitir al método diversificarse dentro de nuevas regiones.

CAPITULO III

MARCO METODÓLOGICO

3.1 Naturaleza de Investigación

Según Hurtado (2010), la investigación de tipo descriptiva está asociada al diagnóstico, tiene como objetivo la descripción precisa del estudio. En la investigación descriptiva se hace una enumeración detallada de las características del evento, trabajan con uno o con más eventos de estudio.

Este tipo de investigación describe la problemática actual del caso de estudio, detallando cada una de las situaciones tratadas en el mismo, permitiendo de esta manera establecer conclusiones referentes al problema.

3.2 Tipo de Investigación

La investigación de tipo tecnológica comprende con mayor énfasis la transformación, cuya finalidad es obtener un conocimiento para lograr modificar la realidad que se está estudiando. Según García (2009) en una investigación de este tipo se obtiene conocimientos que establecen con detalle acciones, características, diseño, métodos, instrumentos, y demás circunstancias, con lo que se promueve el logro de los objetivos.

El presente trabajo de investigación es de tipo tecnológico ya que tiene como finalidad diseñar e implementar un algoritmo metaheurístico híbrido que permita controlar el flujo vehicular en las rampas de acceso (ramp metering) de las autopistas, en particular la autopista I-210 W, localizada en Pasadena, California, para de esta forma optimizar el flujo vehicular dentro del sistema vial en estudio.

3.3 Diseño de Investigación

De acuerdo con Kerlinger (1983) la investigación no experimental es aquella donde se observan los fenómenos ocurridos tal y como se dan en su ambiente natural, para después analizarlos. No se construye ni se manipula ninguna situación, el investigador no puede influir ni tiene el control directo sobre las variables del objeto de estudio, porque ya sucedieron, al igual que sus efectos.

En esta investigación se utiliza el diseño no experimental, ya que el investigador del caso de estudio Meng (2009), analizó y observó las variables del objeto de estudio que se presenta tanto en la autopista I-210 W en Pasadena, California como en las rampas de acceso de la misma, sin la intención de alterar el entorno. A parte de ello la información fue procesada tal y como se recolectó sin ninguna modificación en los datos, para no provocar alguna alteración en los resultados.

3.4 Fases de la Investigación

Para la presente investigación, se realizará una serie de fases basadas en la metodología propuesta por Talbi (2009), la cual abarca desde la formulación hasta la implementación de la solución en los modelos de optimización metaheurísticos. Los pasos que se llevarán a cabo son los siguientes:

- 1) **Formulación del problema:** en esta primera fase se debe identificar el problema para posteriormente realizar un planteamiento en donde la formulación del mismo aborda los diferentes factores internos, externos y los objetivos del problema, los cuales va a servir de ayuda en la resolución del planteamiento. Para la presente investigación se tomará el problema planteado por (Meng, 2009), en donde se identifica la necesidad de optimizar el flujo vehicular en una autopista que contiene 21 rampas de acceso.
- 2) **Modelado del problema:** en este importante paso se construye un modelo matemático abstracto de acuerdo a los basamentos obtenidos en el punto anterior, para ello se puede utilizar modelos similares hasta llegar a reducir el problema a un modelo de optimización bien estudiado. En el caso del presente trabajo de investigación se toma la información de la descripción del problema para posteriormente crear el modelo de acuerdo a las características del comportamiento del sistema vial a estudiar.
- 3) **Optimización del problema:** una vez modelado el problema, el procedimiento de resolución genera una “buena” solución que puede ser óptima o subóptima. En este punto se ajusta el modelo matemático obtenido para su implementación con un algoritmo híbrido para posteriormente tomar las funciones objetivo con sus restricciones y así poder analizar el objetivo en términos de maximizar el flujo vehicular del problema modelado.
- 4) **Implementación de la solución:** una vez que se obtiene la solución se prueba e implementa en caso de considerarse aceptable el resultado del mismo. Algún conocimiento práctico puede ser introducido en la solución para ser implementado. Si la solución es inaceptable el modelo y/o algoritmo de optimización debe ser corregido y el proceso de toma de decisión debe repetirse. En el presente trabajo investigativo la optimización se realizará a

través de la implementación de un algoritmo híbrido a ejecutar en dos etapas. La primera etapa se optimiza a través del NSGA-II, en donde se obtiene un primer resultado que alimentará a un segundo algoritmo. En la segunda etapa es donde se implementa la hibridación compuesta por dos metaheurísticas para conseguir una solución del problema de optimización multiobjetivo.

CAPITULO IV

ANALISIS E INTERPRETACIÓN DE LOS RESULTADOS

En el trabajo titulado “A Pareto-optimization approach for a fair ramp metering.” (Meng, 2009), se plantea la optimización del flujo vehicular de una autopista que posee 21 rampas de acceso y 18 rampas de salida, a lo largo de un intervalo de la autopista interestatal 210W ubicada en Pasadena, California. Las rampas de acceso pueden ser controladas a través de diferentes mecanismos (carriles reversibles, carriles exclusivos, apertura del arcén, entre otros), siendo el más usado la implementación de semáforos, regulando de esta manera la incorporación de vehículos a la vía principal de la autopista, permitiendo establecer una tasa de acceso suficientemente alta sin colapsar el canal de incorporación, proporcionando a los usuarios tiempos justos en cuanto a la espera en la rampa, así como el flujo en la vía principal, consiguiendo de esta forma un desempeño aceptable del sistema vial completo. En la figura 6 se muestra la estructura de la autopista a estudiar.

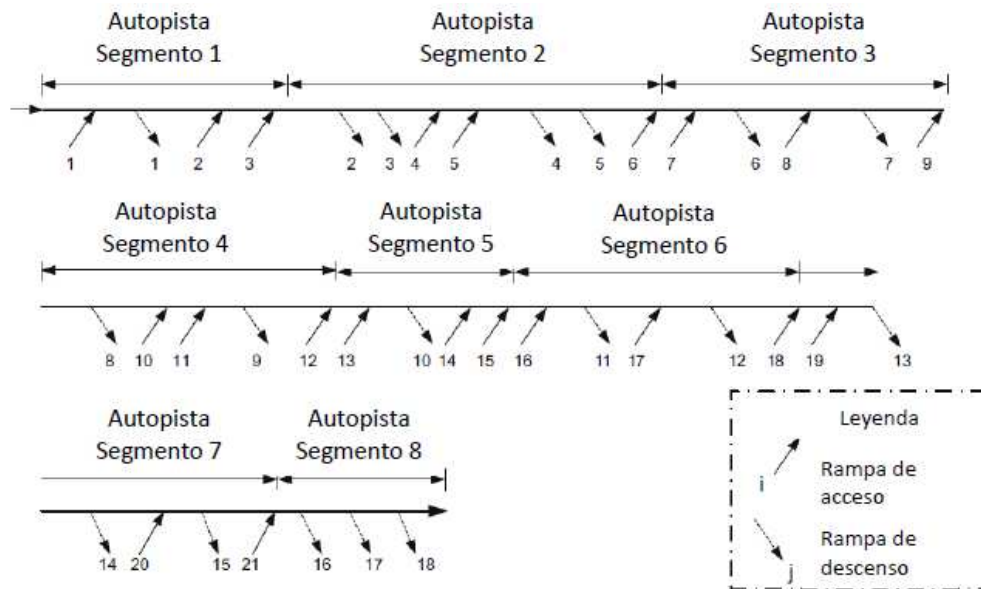


Figura 6 Estructura topológica de la autopista, separada en 8 segmentos.

(Meng, 2009)

Una vez definido el alcance del problema, el autor (Maldonado, 2012) procede a identificar los elementos que permitirán realizar una percepción del comportamiento del sistema vial completo en función de las demandas de los distintos componentes en estudio. Los siguientes elementos se identificaron:

- 1) Rampa de acceso: le permite a los usuarios integrarse al flujo principal de la autopista proveniente de otros sitios, siendo como un sistema de interconexión a otros lugares. Estas rampas poseen una demanda promedio de vehículos/hora y una capacidad máxima de vehículos/hora. El autor (Meng, 2009) agrega: “entre el 40% y 60% 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”(citado por Maldonado, 2012 p.72).

- 2) Vía principal: es la autopista en sí, permitiendo el traslado de los conductores de una localidad a otra. (Maldonado, 2012) agrega lo siguiente:

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 supera la capacidad máxima de la autopista. (p. 72).

- 3) Rampas de descenso: permiten al conductor salir de la autopista para alcanzar su lugar de destino a través de otras vías. El octavo segmento de la autopista a estudiar sólo tiene rampas de descenso, por lo tanto no se tomará en cuenta para el resto del trabajo, ya que el problema planteado solo regula entradas.

En la Figura 7 se muestra un esquema, donde se identifican los diferentes elementos que intervienen en las rampas de acceso presentes en el segmento de estudio, los cuales son: los vehículos que están en tránsito dentro de la autopista, los vehículos que están en proceso de incorporación una vez permitido el paso de la rampa a la vía principal y los vehículos que esperan la señal de control para ingresar a la autopista. Todos estos elementos deben ser considerados al momento de plantear el modelo matemático que representará este caso de estudio.

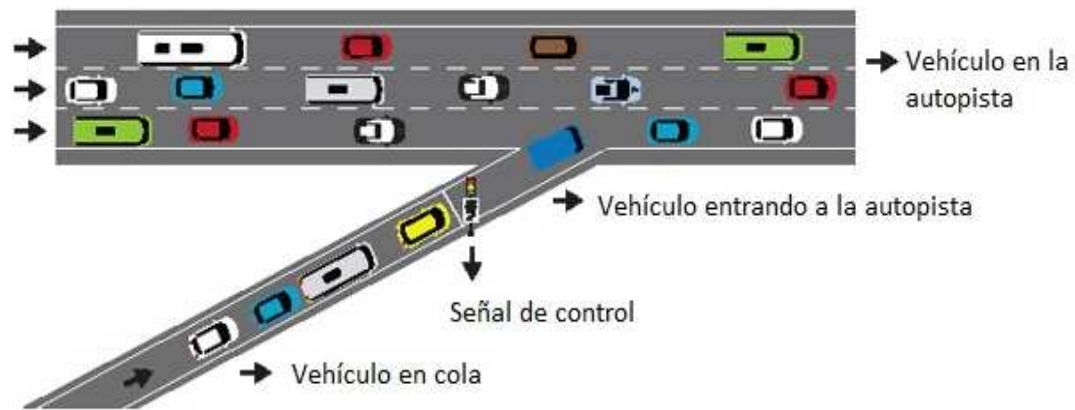


Figura 7 Esquema de una rampa de acceso

Para el desarrollo y optimización de este trabajo, se tomará parte del principio de modelado planteado por (Meng, 2009), donde define: “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.” (citado por Maldonado, 2012 p. 74).

A continuación se definen los siguientes elementos presentes en el caso de estudio:

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, el autor (Maldonado, 2012) obtiene después de varias modificaciones el siguiente modelo matemático de optimización multiobjetivo.

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:

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

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

$$d_n > 0$$

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

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

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

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

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

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

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

De acuerdo al número de variables y restricciones, se selecciona como método de optimización un algoritmo híbrido multiobjetivo, que estará compuesto por un algoritmo genético NSGA-II en su variación multiobjetivo y uno de búsqueda local Simulated Annealing para la selección de soluciones óptimas.

4.1 Implementación del NSGA II

En la implementación del algoritmo genético se utilizó una configuración idéntica a la empleada por el autor (Maldonado, 2012), esto con el objeto de comparar los resultados obtenidos por este último con las soluciones del algoritmo propuesto, la cual consistió en definir un cromosoma de 210 bits a cada uno de los individuos de la población, de los cuales 10 bits fueron asignados a cada una de las 21 rampas existentes en la autopista que conforman las variables de decisión, una probabilidad de 70% de cruce y 0.2% de mutación, el tamaño de población es de 420 individuos. 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 evolucionó a través de 100 generaciones considerando un límite de tiempo de una hora.

El algoritmo implantado en esta investigación es de tipo generacional, la población inicial (soluciones candidatas al problema) se generó aleatoriamente, posteriormente se evalúa cada solución usando la función de adaptación (relacionada con la función objetivo a optimizar), la cual le asigna un número real no negativo único a la estructura de datos que representa una de las posibles soluciones del espacio de búsqueda del problema. Por consiguiente, la función de adaptación permite determinar cuáles individuos corresponden a buenas propuestas de solución y cuáles no.

Dentro de un ciclo repetitivo se realiza la fase de reproducción según el número de generaciones establecidas anteriormente en el problema, para ello se seleccionan a

través de operadores genéticos los individuos de la población para cruzarse y producir descendientes, que constituyen, una vez mutados, la siguiente generación.

Los tres operadores básicos que de los algoritmos genéticos son: selección, mutación y cruzamiento. En el trabajo del autor (Maldonado, 2012) usó como operador de selección para los individuos de la población la versión torneo determinística, la cual consiste en seleccionar al azar un número p de ejemplares (se selecciona $p=2$). De entre los seleccionados se selecciona el más apto para pasarlo a la siguiente generación.

Una vez realizada la selección de los cromosomas se procede a realizar la reproducción o cruce entre dos de estos individuos (padres), siendo una de las operaciones fundamentales dentro del algoritmo, ya que consiste en un intercambio genético de los mismos para obtener otros dos nuevos individuos (hijos) con posible mejor aptitud que el de sus progenitores. El cruzamiento se hace reemplazando algunos genes de un padre por los genes correspondientes del otro padre.

Tras el cruce, tiene lugar la mutación que consiste en cambiar alguna(s) de las características de uno de los individuos para crear uno nuevo. Para el caso de una codificación binaria, la mutación se basa simplemente en la inversión del gen mutado que corresponderá con un bit.

Posteriormente se aplica el método elitismo, el cual es el más usado para mejorar la convergencia del algoritmo. El mecanismo elitista pretende asegurar que aquellos individuos que son los más aptos de la población actual sobrevivan y continúen participando en el proceso evolutivo, pasando a la siguiente generación de manera intacta. Para finalizar, se calcula el rank ratio (relación de rango), con la finalidad de que los individuos con un alto valor de ranking tengan más probabilidad de ser seleccionado y pueda formar parte de la nueva población.

Una vez finalizada la ejecución del algoritmo elaborado por el autor (Maldonado, 2012), se obtuvo una frontera de Pareto en cada una de las 30 iteraciones, para posteriormente calcular un promedio entre los vectores resultantes de cada una de las funciones objetivos, logrando unos resultados consistentes, sin embargo para ese trabajo no fue posible hacer una comparación para conocer qué tan eficiente era el método, ya que no se contaba con otro procedimiento que pudiera contrastar la calidad del mismo.

4.2 Implementación del Simulated Annealing

Para poder realizar la comparación de resultados y determinar qué tan eficiente es el método se procede a la hibridación del algoritmo NSGA II con el Recocido Simulado (Simulated Annealing), usando para ello el método integrativo ubicado dentro de la Clasificación de las metaheurísticas híbridas (Blum, Blesa, Roli y Sampels, 2008), este procedimiento es uno de los más populares (Hoos y Stutzle, 2005), su uso se basa en que dentro del algoritmo principal se incluye uno o más llamado(s) a otro(s) algoritmo(s) trabajando de forma paralela con la finalidad de obtener una resolución a un determinado problema de optimización.

Previamente a la ejecución del algoritmo metaheurístico se configura ciertos parámetros iniciales para el correcto funcionamiento del mismo en el archivo de entrada (llamado input.h), dentro del cual también se encuentran las variables iniciales para la operatividad exacta del NSGA II.

Las variables configuradas son: temperatura inicial, factor de enfriamiento, M (controla los pasos realizados en una simple iteración del SA, se ejecuta sobre cada individuo de la población de los padres, después que el método de selección los escoja). La temperatura inicial define el momento en el que se inicia el proceso de enfriamiento, debe ser una temperatura que permita casi (o todo) movimiento, es decir que la

probabilidad de pasar de un estado (a) al siguiente (b) (en el conjunto de soluciones S_n) sea alta, sin importar la diferencia del costo de la solución $(F(S_j) - F(S_i))$. El factor de enfriamiento es el método en donde la temperatura va tendiendo a 0. Es decir, la probabilidad de aceptación de soluciones peores tiende a 0. El proceso, por tanto, determina cómo se modifica la temperatura durante la ejecución del algoritmo. Hay dos modelos básicos que se pueden utilizar para el proceso de enfriamiento. En el primero se puede ir enfriando muy lentamente con cada iteración de la cadena de Markov de manera que en un número de pasos extremadamente alto la temperatura tienda a 0, o utilizar un proceso de enfriamiento más rápido, que permita al sistema llegar a un estado donde se estabilice la solución.

Algunos de los algoritmos de enfriamiento utilizados (Díaz, Fernández-Toribio, Allende y Manso, 2007) son los siguientes:

- Descensos constantes de temperaturas.
- Descenso exponencial: $T_{k+1} = \alpha T_k$, ($0,8 < \alpha < 0,99$).
- Criterio de Boltzmann: $T_k = T_0 / (1 + \log(k))$.
- Esquema de Cauchy: $T_k = T_0 / (1+k)$.

En el presente trabajo, se utilizó el factor descenso exponencial, ya que es el más usado y daba a entender que a valores elevados de α , con valores entre 0.8 y 0.99 (correspondientes a velocidades lentas de enfriamiento) eran los que mejores resultados proporcionaban (Díaz, Fernández-Toribio, Allende y Manso, 2007).

El ciclo Metrópolis (Hernández, 2000) es la característica esencial del recocido simulado, ya que determina como explotar aleatoriamente nuevas soluciones, rechazándolas o aceptándolas a una temperatura constante, dicho proceso de

movimiento de un estado a otro es repetido por un número de iteraciones bajo la misma temperatura.

Una vez terminada la configuración de los parámetros, el algoritmo comienza con un estado inicial, el cual se obtuvo previamente de la aplicación del método de selección (elige los padres) realizado en el algoritmo genético NSGA-II, una vez obtenido dicho estado se inicia un método iterativo a una temperatura inicial alta T_0 , la cual gradualmente se va reduciendo al multiplicarse con la variable factor de enfriamiento (α , donde $0 < \alpha < 1$, cuyos valores oscilan entre 0.8 a 0.99), a medida que el algoritmo se va ejecutando en un determinado número de ciclos M (ciclo Metrópolis).

En cada paso el algoritmo recocido simulado encuentra un vecino de la solución actual, el cual va a ser evaluado a través de la función evaluación y poder así comparar tanto el valor del vecino como el de la solución actual. Si el vecino generado es mejor que el de la solución actual, entonces el vecino es aceptado como la nueva solución, sin embargo, si las sustituciones son permitidas únicamente para las soluciones “mejores”, entonces el algoritmo quedaría atrapado por el óptimo local y no encontraría el óptimo global, por lo tanto, se debe permitir la sustitución de la solución actual por su peor vecino.

El reemplazo con un peor vecino se hace con cierta probabilidad, la cual se calcula de la siguiente manera:

$$e^{-\frac{E_{neighbor} - E_{current}}{T}}$$

Donde $E_{neighbor}$ es el valor de la función objetivo del vecino, $E_{current}$ es el valor de la función objetivo de la solución actual y T el valor de la temperatura, la cual también se usa para el control de la probabilidad en aceptar los peores vecinos.

Una vez finalizada la ejecución del algoritmo metaheurístico, retorna al algoritmo genético las nuevas soluciones para que el NSGA II pueda encontrar respuestas óptimas al problema planteado. Posteriormente se harán pruebas con el objetivo de comparar los resultados obtenidos por el autor (Maldonado, 2012) y los del algoritmo híbrido propuesto en la investigación, con la finalidad de verificar cuál de los métodos produce mejores soluciones a partir del mismo contexto del problema.

A continuación se analizan las fronteras de Pareto obtenidas por el algoritmo híbrido propuesto y las del autor (Maldonado, 2012) en la Figura 8

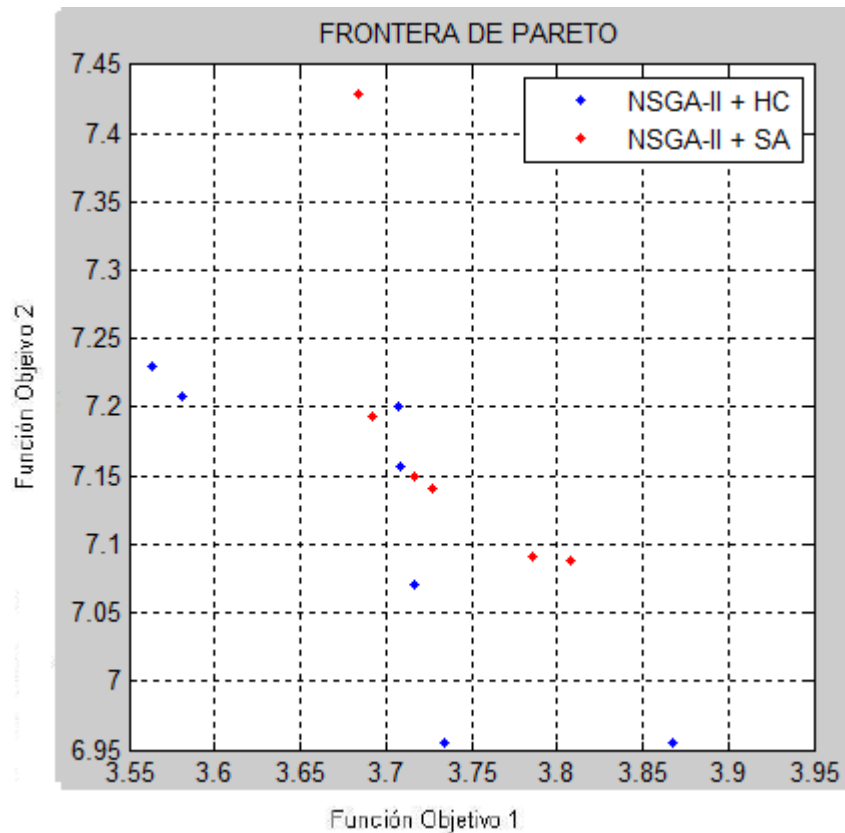


Figura 8: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 100 generaciones

En la figura 8 se puede observar de manera perceptiva que el algoritmo híbrido elaborado en este trabajo (NSGA-II + SA) pudiera ser mejor que el realizado por el autor (Maldonado, 2012) (NSGA-II + HC), esto debido a que la función objetivo 1 y la función objetivo 2 representan un indicador máximo del sistema completo, es decir, mientras más alto es el valor, mayor será el flujo vehicular entregado por la solución, por lo que se puede inferir que un valor alto en estas funciones, significa que es una mejor solución. Esta percepción empírica se puede comprobar con los indicadores Hypervolume y R2.

A continuación se analiza la solución obtenida con otros ciclos de generaciones (500 y 1000 generaciones), con la finalidad de observar las mejoras introducidas al algoritmo propuesto en el trabajo.

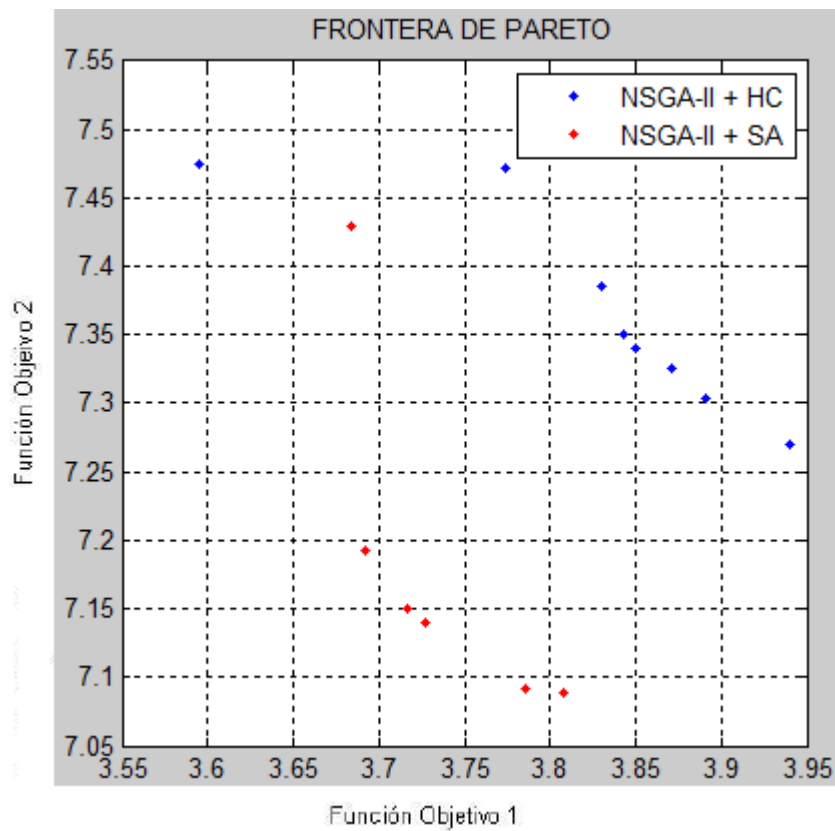


Figura 9: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 500 generaciones

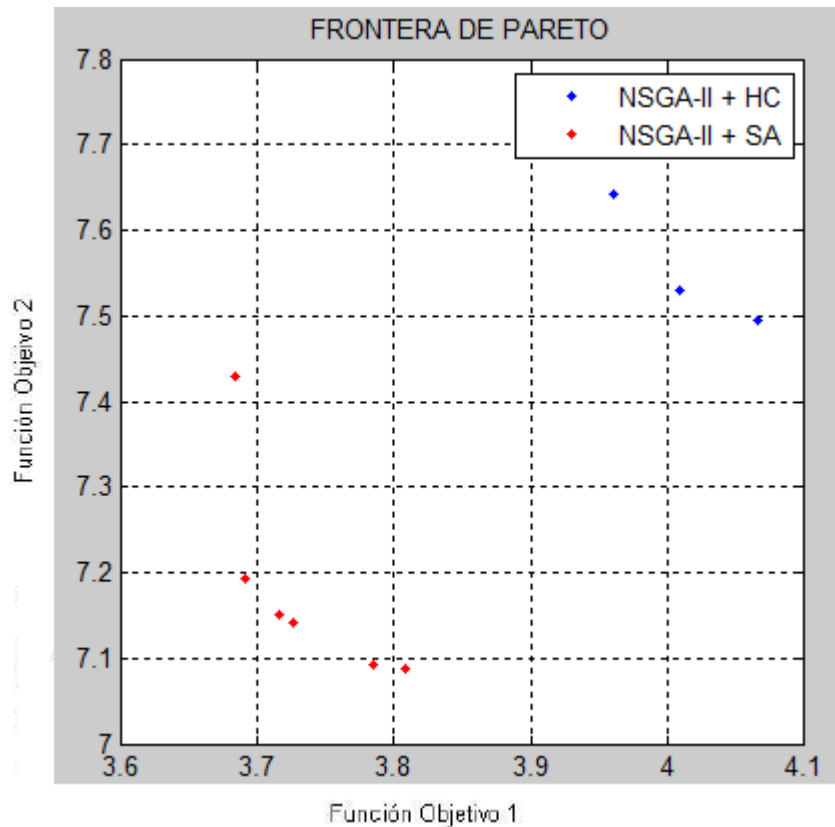


Figura 10: Frontera de Pareto generada por el algoritmo autor (Maldonado, 2012) y el algoritmo híbrido comparado en 1000 generaciones

Como se puede observar en la Figura 9 y 10, el algoritmo elaborado en este trabajo (NSGA-II + SA) pudiera ser mejor que el del autor (Maldonado, 2012) (NSGA-II + HC), ya que requiere de un esfuerzo computacional menor para tener mejores soluciones.

Con la finalidad de evidenciar los recursos computacionales utilizados por el algoritmo híbrido elaborado en el presente trabajo de investigación (NSGA-II + SA) y el del autor (Maldonado, 2012) (NSGA-II + HC), se ejecutaron ambos para cada uno de los ciclos generacionales (100, 500 y 1000), usando para ello una computadora con las siguientes especificaciones técnicas: procesador Intel Core 2 Duo 2.27 GHz,

memoria RAM de 4GB, sistema operativo Windows 7 Ultimate de 64 bits, obteniendo los siguientes resultados:

	<i>Algoritmo NSGA-II + HC</i>	<i>Algoritmo NSGA-II + SA</i>
100 Generaciones	15 minutos y 56 segundos	8 minutos y 16 segundos
500 Generaciones	49 minutos y 36 segundos	37 minutos y 26 segundos
1000 Generaciones	1 hora 37 minutos y 10 segundos	1 hora 14 minutos y 43 segundos

Tabla 1 Recursos computacionales utilizado por el algoritmo híbrido y el del autor (Maldonado, 2012) en diferentes ciclos generacionales

Con los datos obtenidos anteriormente, se corrobora que el algoritmo elaborado en el presente trabajo de investigación (NSGA-II + SA) genera soluciones de una manera eficiente y en un tiempo computacional menor con respecto a las del autor (Maldonado, 2012).

Existen diferentes indicadores para evaluar medidas de calidad, sin embargo para el presente trabajo se utilizarán el indicador Hypervolume y el R2, ya que según (Basseur, Talbi, Nebro y Alba, 2006) son los más usados para realizar la comparación de dos algoritmos evolutivos multiobjetivos (MOEA).

4.3 Indicador Hypervolume

Este indicador fue propuesto por (Zitzler y Thiele, 1998). Consiste en la medida de la región dominada por el conjunto de aproximación y acotada por un punto de referencia $\vec{r} \in \mathbb{R}^k$. Será denotado por I_H .

Si el conjunto de aproximación A está compuesto por los puntos $A = \{\vec{X}_1, \dots, \vec{X}_l\}$, entonces $I_H(A)$ es igual volumen k-dimensional (área en el caso de dos objetivos,

volumen para tres objetivos) de la región formada por la unión de los hiperrectángulos delimitados por cada $\vec{x}_j \in A$ y el punto de referencia \vec{r} (ver ejemplo en la figura 11). Una particularidad importante de este indicador es que entre todos los posibles conjuntos de aproximación para un problema particular, aquél que maximiza el hipervolumen está compuesto únicamente por puntos Pareto óptimos (Fleischer, 2003).

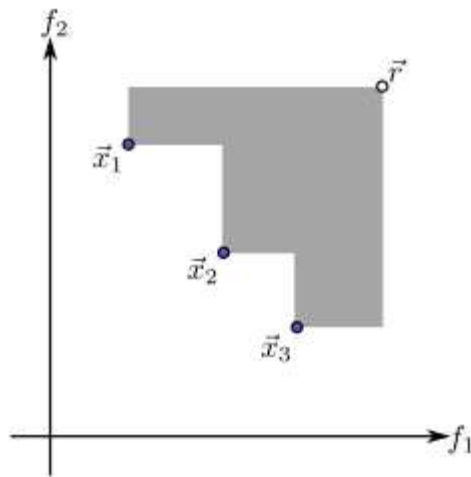


Figura 11. Ejemplo del hipervolumen para un problema de dos objetivos. El área sombreada es el valor del indicador

4.4 Indicador R2

El indicador R2 mide la convergencia al determinar la distancia mínima (en cualquier objetivo) que habría que desplazar cada solución para ser no dominada con respecto a otro frente (cuanto más pequeño mejor) (Hansen y Jaszkiewicz, 1998). Dado un conjunto de funciones U utilizadas para modelar todas las preferencias posible 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 , siendo $p(u)$ la probabilidad de que el tomador de decisiones seleccione a u como una buena solución (Pilegaard y Jaszkiewicz, 1998).

4.5 Comparación de resultados entre los algoritmos

De los indicadores seleccionados, se muestran en la figura 12 y figura 13 respectivamente las gráficas comparativas de los resultados del algoritmo híbrido propuesto en este trabajo y los del autor (Maldonado, 2012)

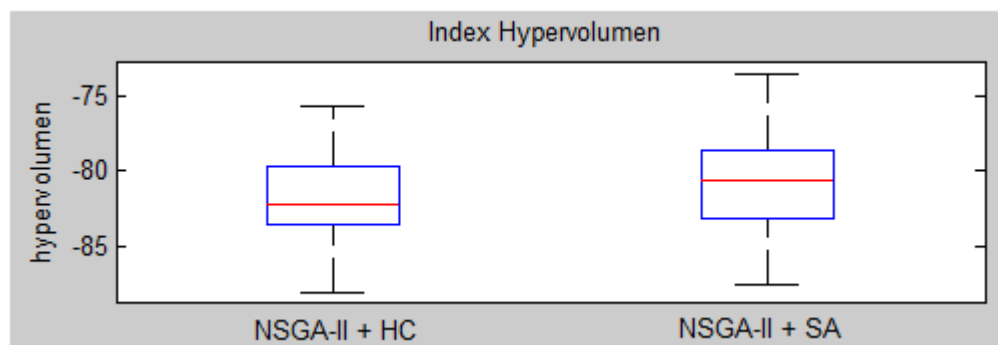


Figura 12 Índice del indicador Hypervolumen. Híbrido 100 generaciones contra algoritmo autor (Maldonado, 2012)

El resultado del presente indicador, se puede observar que las soluciones generadas por el algoritmo híbrido planteado en este trabajo es un 2,73% (resultado obtenido cuando se ejecuta el código de Matlab, el programa genera unas columnas llamadas Hfijo y Rfijo, relacionada con los indicadores Hypervolume y R2 respectivamente, posteriormente se copiaron los valores contenidos en cada columna y se calculó el promedio, de la misma manera se hizo para el resto de los cómputos de las siguientes generaciones) mejor que las del autor (Maldonado, 2012), ya que como se mencionó anteriormente en el Hypervolumen a mayor valor, mejor resultado. A continuación se calcula el índice R2, el cual se muestra en la siguiente figura 13.

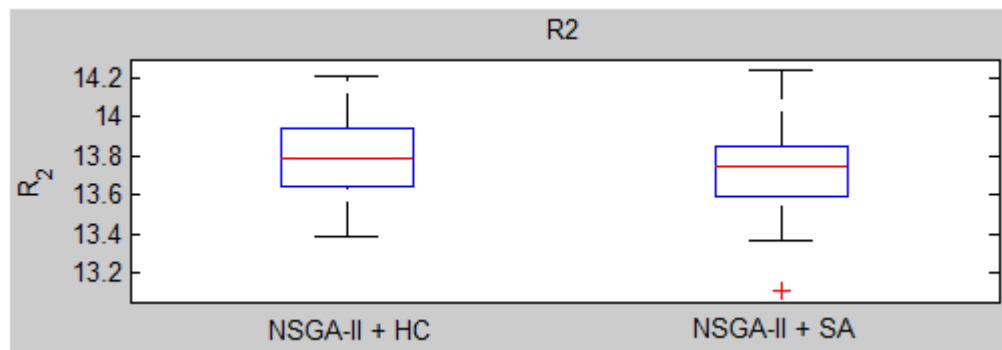


Figura 13 Índice del indicador R2. Híbrido 100 generaciones contra algoritmo autor (Maldonado, 2012)

En la Figura 13, se evidencia que el algoritmo propuesto tiene un valor del 0,87% mejor que el elaborado por el autor (Maldonado, 2012), ya que el índice R2 tiene un valor más pequeño, lo que indica que los resultados están más cerca de una solución óptima de referencia.

A continuación se comprueba la diferencia de resultados entre el algoritmo híbrido propuesto y el del autor (Maldonado, 2012), a través de las gráficas obtenidas al utilizar diferentes valores en las generaciones (las cuáles fueron de 500 y 1000).

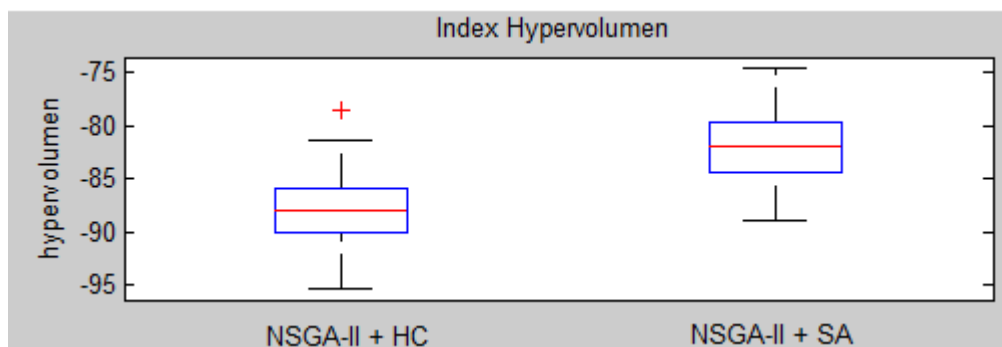


Figura 14 Índice del indicador Hypervolumen. Híbrido 500 generaciones contra algoritmo autor (Maldonado, 2012)

En el resultado de la Figura 14, se puede contemplar que para el indicador Hypervolume, las soluciones generadas por el algoritmo planteado es un 6,67% mejor que las del autor (Maldonado, 2012).

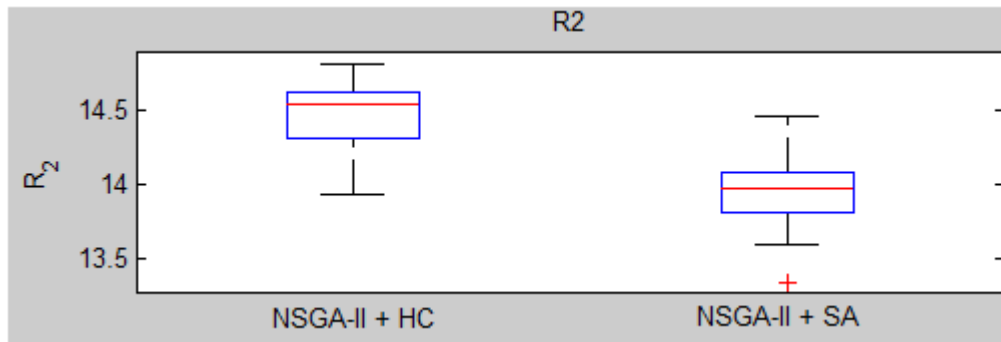


Figura 15 Índice del indicador R2. Híbrido 500 generaciones contra algoritmo autor (Maldonado, 2012)

En la Figura 15 el algoritmo propuesto para el indicador R2 es de 3,26% más pequeño que la del autor (Maldonado, 2012), lo cual nos indica que los resultados están más cerca de una solución óptima de referencia.

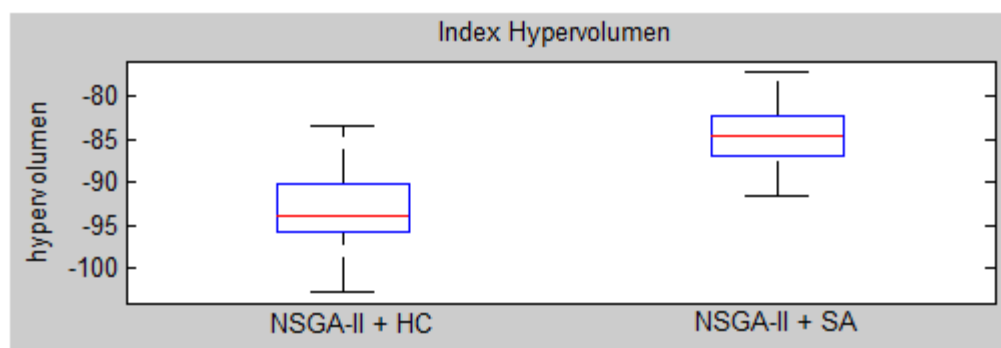


Figura 16 Índice del indicador Hypervolumen. Híbrido 1000 generaciones contra algoritmo autor (Maldonado, 2012)

Se puede observar en la Figura 16 que para el indicador Hypervolume, las soluciones generadas por el algoritmo planteado es un 10,54% mejor que las del autor (Maldonado, 2012).

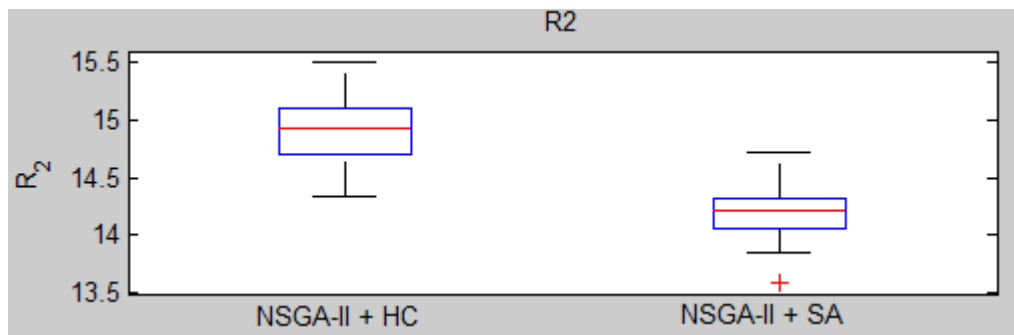


Figura 17 Índice del indicador R2. Híbrido 1000 generaciones contra algoritmo autor (Maldonado, 2012)

En la Figura 17 el indicador R2 nos muestra que las soluciones obtenidas por el algoritmo propuesto es de 5,12% más pequeño que la del autor (Maldonado, 2012), lo cual nos indica que los resultados están más cerca de una solución óptima de referencia.

Finalizado el análisis de los resultados, se procedió a modificar los valores de los parámetros iniciales (temperatura inicial=500, factor de enfriamiento=0.85, cadena de Markov=50) del algoritmo metaheurístico Recocido Simulado, con la finalidad de observar si los nuevos valores obtenidos, por el algoritmo elaborado en la presente investigación (NSGA-II + SA), mantendría la tendencia de generar mejores soluciones con respecto a los del autor (Maldonado, 2012) (NSGA-II + HC).

A continuación se analizan las fronteras de Pareto con los nuevos resultados obtenidos del algoritmo híbrido propuesto y las del autor (Maldonado, 2012)

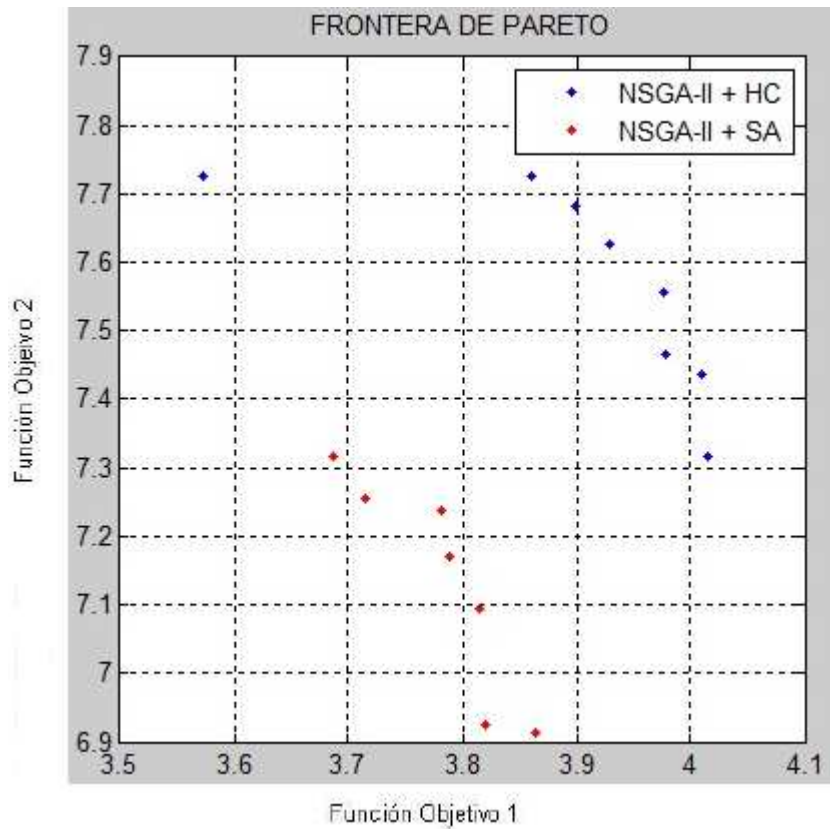


Figura 18 : Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 100 generaciones

A continuación se analiza la nueva solución obtenida con otros ciclos de generaciones (500 y 1000 generaciones), con la finalidad de observar las mejoras introducidas al algoritmo propuesto en el trabajo.

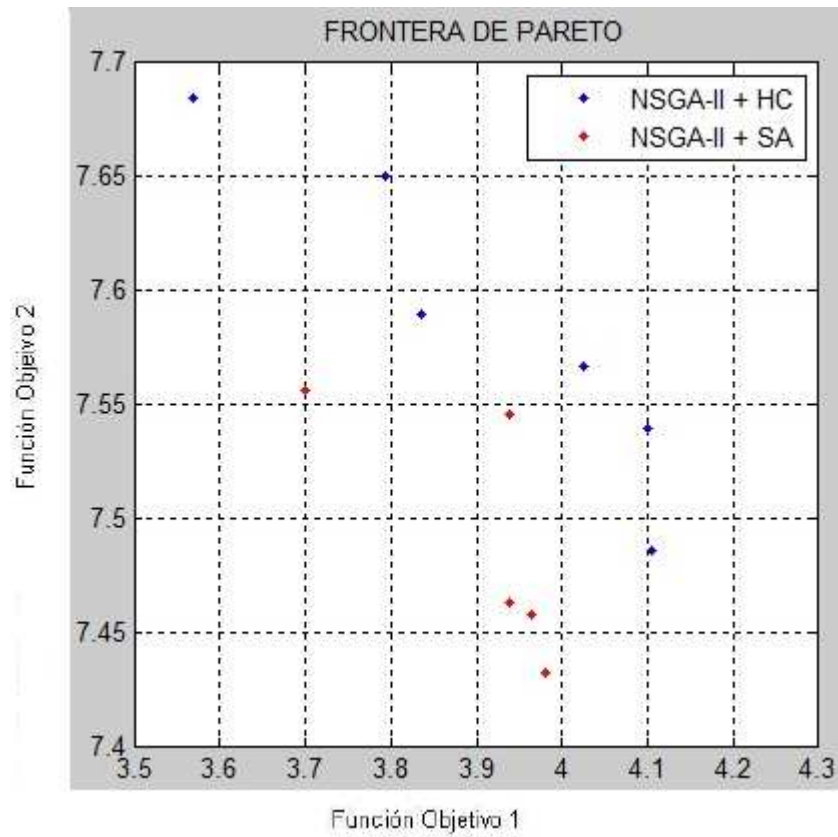


Figura 19: Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 500 generaciones

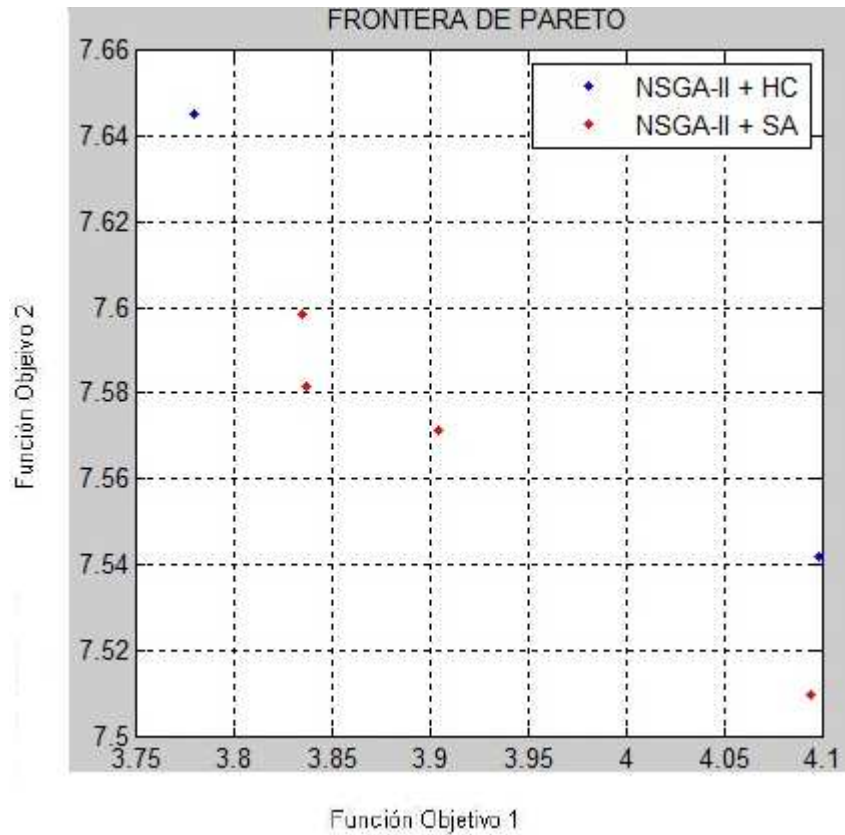


Figura 20: Frontera de Pareto generada por la modificación de los valores iniciales del algoritmo híbrido comparados con los del algoritmo autor (Maldonado, 2012) en 1000 generaciones

En las figuras 18, 19 y 20, se puede observar que el algoritmo elaborado en el presente trabajo (NSGA-II + SA) pudiera ser mejor que el del autor (Maldonado, 2012) (NSGA-II + HC), debido a que requiere un esfuerzo computacional menor para tener mejores soluciones, y también porque las funciones objetivos 1 y 2 representan un indicador máximo del sistema completo, por lo que se puede inferir que un valor alto en estas funciones, significa que es una mejor solución.

En la figura 21 se puede contemplar que para el indicador Hypervolumen, las soluciones generadas por el algoritmo planteado es un 7,86% mejor que las del autor (Maldonado, 2012).

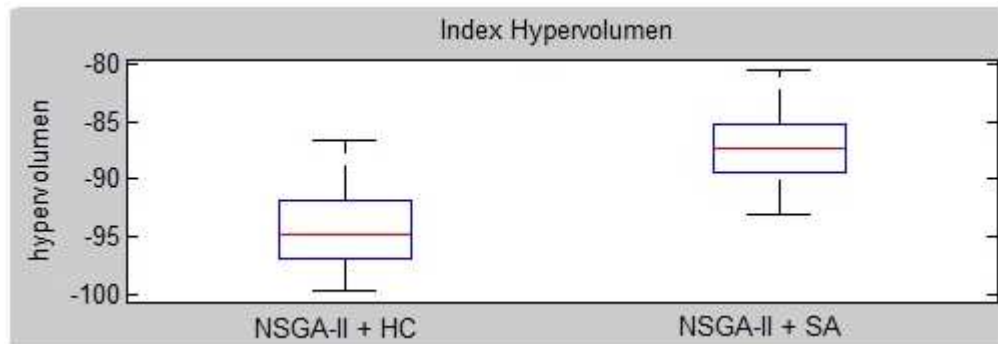


Figura 21 Índice del indicador Hypervolumen generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)100 generaciones

En la Figura 22 el algoritmo propuesto para el indicador R2 es de 9,33% más pequeño que la del autor (Maldonado, 2012), lo cual nos indica que los resultados están más cerca de una solución óptima de referencia

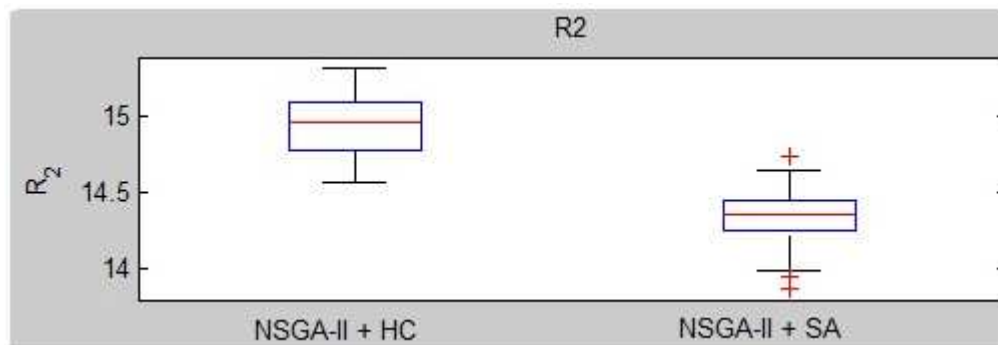


Figura 22 Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)100 generaciones

Se puede observar en la Figura 23 que para el indicador Hypervolume, las soluciones generadas por el algoritmo planteado es un 6,48% mejor que las del autor (Maldonado, 2012).

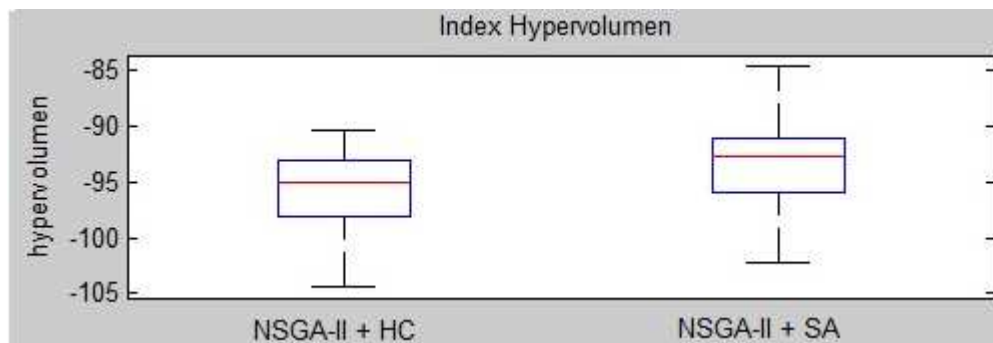


Figura 23 Índice del indicador Hypervolume generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012) 500 generaciones

En la Figura 24 el indicador R_2 nos muestra que las soluciones obtenidas por el algoritmo propuesto es de 4,62% más pequeño que la del autor (Maldonado, 2012), lo cual nos indica que los resultados están más cerca de una solución óptima de referencia.

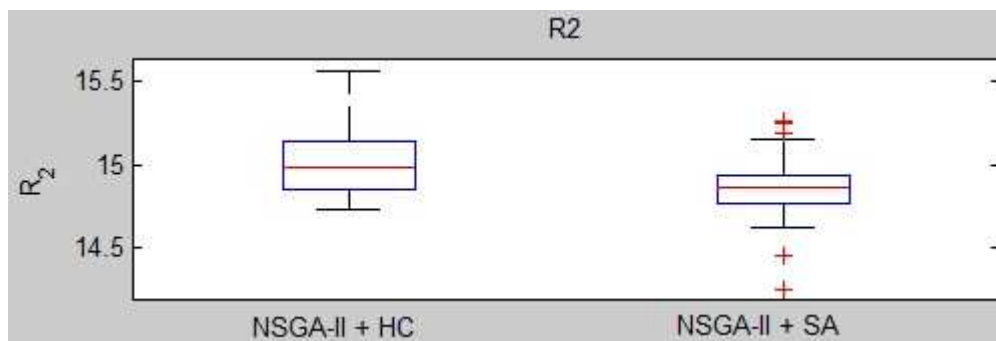


Figura 24 Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)500 generaciones

En el resultado de la Figura 25, se puede contemplar que para el indicador Hypervolumen, las soluciones generadas por el algoritmo planteado es un 3,57% mejor que las del autor (Maldonado, 2012).

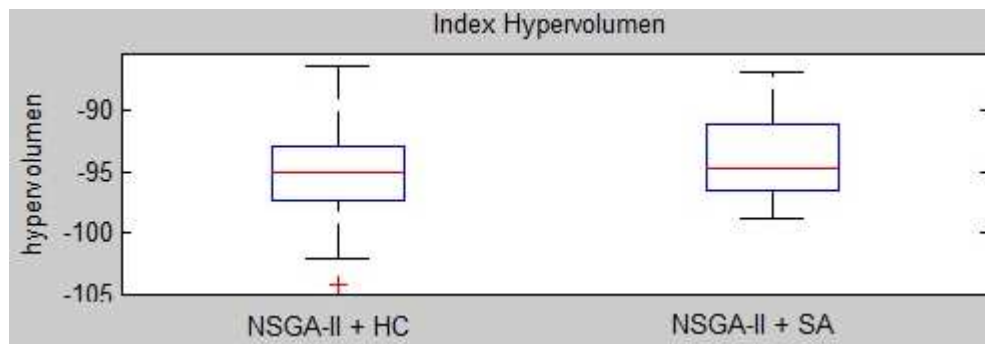


Figura 25 Índice del indicador Hypervolumen generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)1000 generaciones

En la Figura 26 el algoritmo propuesto para el indicador R2 es de 1,48% más pequeño que la del autor (Maldonado, 2012), lo cual nos indica que los resultados están más cerca de una solución óptima de referencia.

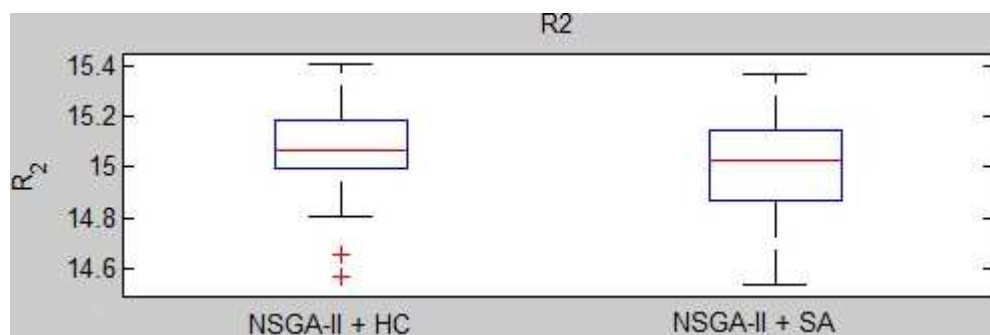


Figura 26 Índice del indicador R2 generada por la modificación de los valores iniciales del algoritmo híbrido contra el algoritmo del autor (Maldonado, 2012)1000 generaciones

Como se observa en las gráficas anteriores, se puede afirmar que el algoritmo elaborado en el presente trabajo investigativo (NSGA-II + SA) es mucho mejor en cuanto a resultado y esfuerzo computacional empleado, en cada uno de los ciclos generacionales (100, 500 y 100), que el del autor (Maldonado, 2012) (NSGA-II + HC), esto debido a la robustez del algoritmo Simulated Annealing.

Al variar los parámetros iniciales del algoritmo metaheurístico y hacer las respectivas pruebas adicionales, se observó que este tiene el mismo comportamiento de seguir generando resultados, los cuales se acercan a una solución óptima de referencia en un tiempo computacional aceptable, para cada uno de los ciclos generacionales (100, 500 y 1000), evidenciando de esta manera la eficiencia introducida por la hibridación en cuanto a la convergencia de las soluciones en problemas multiobjetivos.

CAPITULO V

CONCLUSIONES

Se revisó la documentación sobre los mecanismos de hibridación para problemas de optimización multiobjetivo, con el objeto de poder construir los lineamientos sobre el cual se desarrolló el resto de la investigación, posteriormente al planteamiento del problema, se usó un modelo matemático abstracto que fue evolucionando por el autor (Maldonado, 2012) hasta el utilizado, capaz de representar, a través de una metodología especial (Talbi, 2009) el comportamiento del caso de estudio (Meng, 2009) analizado previamente.

Se utilizó el diseño de la metaheurística híbrida, creada por el autor (Maldonado, 2012), la cual fue elaborada en función de las características del experimento numérico del caso de estudio, para posteriormente hacer la comparativa de los resultados.

Se implementó el algoritmo híbrido (utilizando para ello el algoritmo genético NSGA-II y el metaheurístico Simulated Annealing) en base a un esquema integrativo, ejecutándose para cada uno de los ciclos generacionales (100, 500 y 1000) de acuerdo a las variables de decisión que conforman el modelo planteado, logrando de esta manera unos resultados satisfactorios con el método propuesto, siendo más eficiente que el elaborado por el autor (Maldonado, 2012).

Posteriormente se cambiaron los valores de los parámetros iniciales del método metaheurístico Simulated Annealing, obteniendo de igual manera en cada uno de los ciclos generacionales (100, 500 y 1000), una convergencia eficaz en términos de calidad de los resultados y esfuerzo computacional. Lo anteriormente descrito se fundamenta en las pruebas aplicadas a los algoritmos (NSGA-II + HC y NSGA-II + SA), a través de las métricas de calidad utilizadas como lo son Hypervolume y R2, los cuales analizan la calidad de los resultados de ambos algoritmos.

El presente trabajo servirá de fundamento a investigaciones futuras asociadas al área de Inteligencia Artificial, evidenciando la potencialidad de los algoritmos híbridos en la solución de problemas multiobjetivo, particularmente, en casos donde se requiera de la optimización del tráfico automotor en vías de alto flujo vehicular concretamente en autopistas, evitando de esta manera que la cantidad de vehículos sobrepase la capacidad de estas, tratando de mantener la demanda en su punto óptimo y así lograr que los usuarios puedan tener sustancialmente un mejor desempeño total del sistema vial.

CAPITULO VI

PROPUESTA

Como continuación del trabajo investigativo, se recomienda realizar pruebas con el algoritmo genético NSGA-III (según el caso de estudio, se adapta mejor en aquellos problemas que contengan múltiples funciones objetivos, mejorando de esta manera la referencia para clasificar y seleccionar los individuos más aptos para formar parte de la siguiente población), combinándola con otras técnicas metaheurísticas, para posteriormente comparar los resultados con el propuesto en este trabajo usando la misma configuración descrita anteriormente.

Se recomienda usar dentro de la clasificación de las metaheurísticas híbridas, el control estratégico de tipo colaborativo, que consiste en el intercambio de información por parte de los algoritmos, los cuales no forman parte uno del otro, en este caso, primero se ejecuta el algoritmo genético NSGA-II, al término del mismo y con las soluciones obtenidas, comienza la ejecución del algoritmo metaheurístico Simulated Annealing.

Referencias Bibliográficas

- Abbass, H., A. (2002, Mayo). The Self-Adaptive Pareto Differential Evolution Algorithm. In Congress on Evolutionary Computation, (1), 831–836.
- Alonso, S. (2004). *La metaheurística de optimización basada en colonias de hormigas: modelos y nuevos enfoques*, Granada, Departamento de Ciencias de la Computacion e Inteligencia Artificial. Universidad de Granada.
- Ávila, B., H. (2006). Introducción a la metodología de la investigación. Chihuahua. México.
- Baños N., R. (2006). Meta-heurísticas Híbridas para Optimización mono-objetivo y multi-objetivo. Paralelización y Aplicaciones. Tesis doctoral. Universidad de Almería. Almería.
- Benavente P., J. (2011). Análisis del control de acceso en autopistas urbanas: caso de estudio en autopista central. Pontificia Universidad Católica de Chile, Santiago de Chile.
- Blum, C. & Roli, A. (2003). Metaheuristics in Combinational Optimization: Overview and Conceptual Comparision. 35. New York: ACM.
- Blum, C., Blesa, M. J., Roli, A., & Sampels, M. (2008). *Hybrid Metaheuristics. An Emerging Approach to Optimization*. Berlin: Springer.

- Booker, L. (1982). Intelligent behavior as an adaptation to the task environment. Ann Arbor, Michigan, USA: University of Michigan.
- Brassard, G., & Bratley, P. (2000). *Fundamentos de algoritmia*. Prentice Hall.
- Cassidy, M. & Rudjanakanoknad, J. (2005). Increasing the capacity of an isolated merge by metering its on-ramp. *Transportation Research Part B: Methodological*, 39(10), 896-913.
- Chankong, V. & Haimes, Y. Y. (1983). *Multiobjective Decision Making: Theory and Methodology*. California: Dover Publications.
- Coello, C. A. (2000). An Updated Survey of GA-Based Multiobjective Optimization Techniques. *ACM Computing Surveys*, 32(2), 109-143.
- Coello, C., Van Veldhuizen D., Lamont G. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. New Jersey: Springer-Verlag.
- Cohon, J., L. & Marks, D., H. (1975). A review and evaluation of multiobjective programming techniques. *Water Resources Research*, 11(2). 208–220.
- Concepción, M. A., Soria, J. A., & Ruben, R. G. (2002). *Investigación Operativa: Modelos y Técnicas de Optimización*. España: UNIVERSIDAD POLITECNICA DE VALENCIA. SERVICIO DE PUBLICACION.
- Conesa, J., Contero, M., & Company, P. (2000). Comportamiento de los algoritmos de optimización en la reconstrucción geométrica de sólidos. *Anales de ingeniería mecánica*, 425-430.

- Cotta-Porras, C. (1998). A study of hybridisation techniques and their application to the design of evolutionary algorithms. *AI Communications*, 11(3,4), 223-224.
- Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. London: John Murray.
- Deb, K., Agrawal, S., Pratap, A., Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multiobjective Optimization: NSGA II. *Lecture Notes in Computer Science*, 1917, 849-858.
- Deternoz, D., & Fernandez, F. (2006). *Sistema de Gestión y Monitoreo del Tránsito a través de Semáforos Inteligentes*. Caracas.
- Díaz, P., Fernández-Toribio, G., Allende, J., & Manso, A. (2007). Metaheurísticas de optimización combinatoria: Uso de Simulated Annealing para un problema de calendarización. *Tecnología y Desarrollo, Volumen V*, 1-25.
- Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for hard Optimization*. Berlin: Springer-Verlag.
- El-Abd, M., y Kamel, M. (2005, Agosto). A taxonomy of cooperative search algorithms. Ponencia presentada en el Segundo Taller Internacional sobre Metaheurísticas Híbridas. Barcelona-España.
- Fleischer, M. (2003). The measure of pareto optima applications to multi-objective metaheuristics.74-74. Berlin: Springer.

- García, C., F. (2009). La investigación tecnológica. Investigar, idear e innovar en Ingenierías y Ciencias Sociales. México: Editorial Limusa.
- Glover, F. (1986). Tabu Search: A Tutorial. University of Colorado, Center of Applied Artificial Intelligence.
- Glover, F., & Laguna, M. (1993). Tabu Search. Modern Heuristic Techniques for Combinatorial Problems. *Blackwell Scientific Publishing*, 71-140.
- Goldberg, D. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Alabama: Addison-Wesley.
- Gutierrez, A., & Amati, V. (2012). Viabilidad del uso de sensores de tráfico en las intersecciones semaforizadas congestionadas del municipio Maracaibo del Estado Zulia. Maracaibo, Zulia, Venezuela.
- Hansen, M. & Jaszkiwicz, A. (1998). Evaluating the quality of approximations to the nondominated set. Institute of Mathematical Modelling Technical University of Denmark.
- Hernández, A., J. (2000). Desarrollo, implementación y prueba de un algoritmo de reconstrucción de objetos a partir de una representación axonométrica, utilizando técnicas de optimización. Universidad Politécnica de Valencia, Valencia.
- Hoos, H. & Stutzle, T. (2005). Stochastic local search: foundations and applications. *1a ed. Springer Verlag*, 61-74.

- Hurtado, J. (2010). Metodología de la investigación Holística Guía para la comprensión holística de la ciencia. Cuarta edición. Quirón Ediciones S.A. Cooperativa Editorial Magisterio (Bogota, Colombia) Ciea-Sypal (Caracas, Venezuela).
- Kirkpatrick, S., Gelatt Jr., C., & Vecchi, M. (1983). Optimization by Simulated Annealing. *Science* 220 (4598), 671-679.
- Laarhoveen, P. J. M. & Aarts, E. H. L. (1989), "Simulated Annealing: Theory and Applications". Kluwer Academic Publishers.
- Laumanns M., Zitzler E., Thiele L. (2001). On the Effects of Archiving, Elitism and Density Based Selection in Evolutionary Multi-Objective Optimization. Springer-Verlag Lecture Notes in Computer Science. 181-196.
- Liu, D. (2007). A Multiobjective Memetic Algorithm Based on Particle Swarm Optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , 37(1), 42-50.
- Mahfoud, S. W. (1995). Niching Methods for Genetic Algorithms. Tesis Doctoral. University of Illinois at Urbana-Champaign.
- Maldonado C., J. (2012). 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. Universidad de los Andes, Mérida.
- Martí. R. (2003). Procedimientos Metaheurísticos en Optimización Combinatoria. *Matemàtiques*, 1(1). 3-62.

- McDermott, J.M., Kolenko, S.J., & Wojcik, R.J. (1979). Chicago Area Expressway Surveillance and Control: Final Report. Illinois: Illinois Department of Transportation.
- Melián B., Moreno. J.A., Moreno. J.M. (2003). Metaheurísticas: Una visión global. Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial 19(2): 7-28.
- Meng, Q. (2009). A Pareto-optimization approach for a fair ramp metering. Transportation Research Part C: Emerging Technologies, 18(4), 489–506.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculation by fast computing machines. *Journal of chemistry* 21, 1087-1091.
- Moins S. 2002. Implementation of a simulated annealing algorithm for Matlab. Training performed in Electronic Systems Linköping Institute of Technology. Linköping University Electronic Press. Linköping , Suecia. 34 p.
- Nesmachnow, S. (2004). Una Versión Paralela del Algoritmo Evolutivo para Optimización Multiobjetivo NSGA-II y su Aplicación al Diseño de Redes de Comunicaciones Confiables. Universidad de la República, Uruguay.
- Pareto, V. (1896). Cours D'Economie Politique, volume I and II. Lausanne: F. Rouge.
- Pilegaard, M. (1998). Metaheuristics for multiple objective combinatorial optimization. Institute of Mathematical Modelling.

- Raidl, G. (2006). A Unified View on Hybrid Metaheuristics. *Springer-Verlag*, 1-12.
- Reeves, C. (1993). *Modern Heuristic Techniques for Combinatorial Problems*. New York: John Wiley & Sons Inc.
- Richetta, A. (2002). EURAMP Projects Sites. Recuperado el 24 de Agosto de 2014, de European Ramp Metering Project: <http://www2.napier.ac.uk/euramp>.
- Sawaragi, Y.; Nakayama, H. & Tanino, T. (1985). *Theory of Multiobjective Optimization* (vol. 176 of Mathematics in Science and Engineering). Orlando: Academic Press Inc.
- Schaffer J. D. & Grefenstette J. J. (1985). Multiobjective Learning via Genetic Algorithms. In *Proceedings of the 9th International Joint Conference on Artificial Intelligence* (1), 593–595, San Francisco: Morgan Kaufmann.
- Srinivas, N., & Deb, K. (1994). Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3), 221-248.
- Talbi, E. G. (2002). A taxonomy of hybrid metaheuristics. *Journal of heuristics*, 8(5), 541-564.
- Talbi, E. (2009). *Metaheuristics: from design to implementation*. New Jersey: John Wiley & Son.
- Torres M., D. (2010). *Metaheurísticas híbridas en selección de subconjuntos de características para aprendizaje no supervisado*. Tesis Doctoral. Universidad Autónoma de Aguascalientes, México.

- Uribe, M. (2011). Sistema Adaptativo de Control y Optimización del Tráfico de un Corredor Vial Semaforizado. Aplicación Ciudad de Medellín. Medellín, Colombia.
- Von Lüken, C., Hermosilla A., Barán B. (2003). **Algoritmos Evolutivos para la Optimización Multiobjetivo: Un estudio comparativo en un ambiente paralelo asíncrono**. Tesis de Maestría, Universidad Nacional de Asunción, San Lorenzo.
- Voß S. 2001. Meta-heuristics: The State of the Art. Lecture Notes in Computer Science, 2148, 1-23.
- Wolpert, D. H., y Macready, W. G. (1997). No free lunch theorems for optimization. *Evolutionary Computation, IEEE Transactions on*, 1(1), 67-82.
- Zanakis S. H. & Evans. J. R. (1981). **Heuristic Optimization: why, when and how to use it**. *Interfaces* 11(5). 84–90.
- Zhang, L., & Levinson, D. (2005). "Balancing Efficiency and Equity of Ramp Meters ASCE". *Journal of Transportation Engineering* 131(6), 477-481.
- Zhang, M., Kim, T., Nie, X., (2001). Evaluation of on-ramp control algorithms. *Institute of Transportation Studies*, University of California.
- Zitzler, E., Deb, K., & Thiele, L. (2000), Comparison of Multiobjective Evolutionary Algorithms: Empirical Results, *Evolutionary Computation*, 8(2), 173–95.
- Zitzler, E., and Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms a comparative case study. 292- 301. Berlin: Springer.