

INSTITUTO DE COMPUTACIÓN, FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA
MONTEVIDEO, URUGUAY

PROYECTO DE GRADO
INGENIERÍA EN COMPUTACIÓN

**Tráfico y Polución Urbana:
Algoritmos Evolutivos Multiobjetivo para el
Tratamiento del Tráfico y la Contaminación
en Áreas Cosmopolitas**

Matias F. Péres F. y Germán Ruiz R.
matias.peres@fing.edu.uy, german.ruiz@fing.edu.uy

Abril de 2016

Supervisores:

Sergio Nesmachnow, Universidad de la República.
Ana Carolina Olivera, CONICET - Universidad Nacional de la Patagonia Austral.

Tráfico y Polución Urbana: Algoritmos Evolutivos Multiobjetivo para el Tratamiento del Tráfico Urbano y la Contaminación en Áreas Cosmopolitas

Matias F. Péres F. – Germán Ruiz R.

Proyecto de Grado

Instituto de Computación - Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, Abril de 2016

URBAN TRAFFIC AND POLLUTION:
MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS FOR
TREATMENT OF
URBAN TRAFFIC AND CONTAMINATION IN COSMOPOLITAN
AREAS

ABSTRACT

Over the last decades, road transport has become by far the major source of environmental pollution and traffic congestion in urban areas. In the present work, several state-of-the-art parallel multiobjective evolutionary algorithms are used to address a signal control problem considering minimize both, travel times and vehicle emissions. Aptitude functions are calculated using a microscopic traffic simulation model. A methodology for estimating origin-destination matrices using traffic counts and GPS traces is also described. The numerical analysis applied to three different networks within the Uruguayan city of Montevideo, reveals that the parallel evolutionary algorithms are able to outperform the current configuration as well as typical planning techniques.

TRÁFICO Y POLUCIÓN URBANA:
ALGORITMOS EVOLUTIVOS MULTIOBJETIVO PARA EL
TRATAMIENTO DEL TRÁFICO Y LA CONTAMINACIÓN
EN ÁREAS COSMOPOLITAS

RESUMEN

En las últimas décadas, el tráfico vehicular se ha convertido en la principal fuente de congestión y de contaminación ambiental en zonas urbanas. Este trabajo estudia el problema de minimizar las emisiones vehiculares de contaminantes y el tiempo de viaje de los vehículos mediante la utilización de distintos algoritmos evolutivos multiobjetivo paralelos del estado del arte. Para el cálculo de las funciones de aptitud se utiliza un modelo microscópico de simulación. También se describe una metodología para la estimación de matrices origen-destino mediante conteos de tráfico y trazas de GPS. El análisis experimental realizado sobre tres zonas de la ciudad de Montevideo (Uruguay) en distintas franjas horarias demuestra que los algoritmos evolutivos paralelos son capaces de alcanzar resultados de alta eficacia numérica en comparación con la situación actual y las técnicas de planificación habituales.

Índice general

1. Introducción	1
2. Computación Evolutiva	3
2.1. Introducción	3
2.2. Conceptos básicos de la Computación Evolutiva	4
2.2.1. Estructura general de un AE	7
2.3. Principales paradigmas de la Computación Evolutiva	8
2.3.1. Algoritmos genéticos	8
2.3.2. Estrategias evolutivas	10
2.3.3. Programación evolutiva	11
2.3.4. Evolución diferencial	12
2.4. AE para optimización multiobjetivo	13
2.4.1. Optimización multiobjetivo	13
2.4.2. Algoritmos evolutivos multiobjetivo	15
2.4.3. NSGA-II	16
2.4.4. GDE3	17
2.4.5. MOEA/D	19
2.4.6. ϵ -NSGA-II	21
2.5. AE paralelos	21
2.5.1. Modelos de paralelización	22
2.6. Evaluación experimental de AEMO	24
2.6.1. Métricas de evaluación	25
2.6.2. Análisis estadístico	27
2.6.3. Análisis no paramétrico (análisis de varianza)	28
3. Planificación de ciclos de semáforos y las emisiones vehiculares	31
3.1. Introducción	31
3.2. Planificación de ciclos de semáforos	32
3.2.1. Red de tráfico	32
3.2.2. Flujo vehicular	34
3.2.3. Matriz OD	35
3.2.4. Simulación de tráfico	37
3.2.5. Formulación general de la planificación de ciclos de semáforos	38
3.2.6. Formulación matemática del problema	38
3.3. Estimación de una matriz OD utilizando conteos de tráfico	39

3.3.1.	Localización de conteos de tráfico	40
3.3.2.	Ordenamiento de arcos basado en flujo previo	41
3.4.	Contaminación atmosférica	43
3.4.1.	Contaminantes atmosféricos	44
3.4.2.	Contaminación vehicular	46
3.5.	Planificación de ciclos de semáforos considerando emisiones vehiculares	50
3.5.1.	Formulación matemática del problema considerando emisio- nes vehiculares	50
3.6.	Trabajos relacionados	51
4.	Estrategia de resolución	55
4.1.	Introducción	55
4.2.	Procedimiento general de evaluación	55
4.3.	SUMO: Simulation of Urban Mobility	57
4.4.	Modelado de la red de tráfico	58
4.4.1.	Modelado físico y reglamentario	58
4.4.2.	Modelado del flujo vehicular	61
4.5.	Generación de tráfico SUMO mediante integración GPS/GIS	63
4.5.1.	PowerGen	64
4.5.2.	PriorWeight	67
4.5.3.	TrafficGenerator	67
4.6.	Diseño e implementación de los AEMO propuestos	68
4.6.1.	Entorno de programación	68
4.6.2.	Operadores evolutivos	70
4.6.3.	Técnicas de paralelización	70
4.6.4.	Representación de soluciones	71
4.6.5.	Funciones objetivo	71
5.	Análisis experimental	73
5.1.	Introducción	73
5.2.	Plataforma de desarrollo y ejecución	73
5.3.	Instancias de evaluación	74
5.3.1.	Estudio del tráfico	75
5.3.2.	Validación del valor Ranking	75
5.4.	Calibración paramétrica	78
5.5.	Experimentos sobre la zona de Aguada	79
5.5.1.	Comparación entre AEMO	80
5.5.2.	Comparación entre AEMO y configuración real	80
5.5.3.	Comparación con técnicas alternativas	81
5.5.4.	Análisis de una solución en el frente de Pareto	84
5.6.	Experimentos sobre la zona de Jacinto Vera	86
5.6.1.	Comparación entre AEMO y configuración real	87
5.6.2.	Análisis de una solución en el frente de Pareto	90
5.7.	Experimentos sobre la zona Centro	90
5.7.1.	Comparación entre AEMO y configuración real	91

5.7.2. Comparación con técnicas alternativas	93
5.7.3. Análisis de una solución en el frente de Pareto	96
6. Conclusiones y trabajo futuro	99
6.1. Conclusiones	99
6.2. Trabajo futuro	100
Apéndices	
A. Generación de las instancias de evaluación	103
A.1. Generación de la red SUMO	103
A.2. Uso de las herramientas para el análisis de trazas de GPS	105
A.2.1. PowerGen	105
A.2.2. PriorWeight	105
A.2.3. TrafficGenerator	106
B. Ejecución de los AEMO	107
B.1. MOEA Framework	107
B.2. Ejecución en cluster FING	107
C. Resultados de la calibración paramétrica	109
C.1. Etapa 1	109
C.2. Etapa 2	111
C.3. Etapa 3	114
D. Resultados de comparación entre los AEMO ejecutados	115
D.1. Escenario Aguada	115
D.2. Escenario Jacinto Vera	117
D.3. Escenario Centro	118
Bibliografía	118

Índice de figuras

2.1.	Cruzamiento en un AG: de 1-punto, n-puntos y cruzamiento uniforme.	10
2.2.	Ejemplo de frente de Pareto PF^{true} para $\Omega \subset \mathbb{R}^2$.	14
2.3.	Proceso de determinación de nueva población en el algoritmo NSGA-II.	17
2.4.	Modelo de paralelización <i>maestro-esclavo</i> .	22
2.5.	Vista esquemática del modelo de islas (a), modelo celular (b) y de híbrido del modelo de islas y del modelo celular (c).	23
2.6.	Ilustración de búsqueda para captura de proximidad y diversidad (a) y de posibles soluciones (b).	25
2.7.	Hipervolumen para un caso de dos objetivos.	26
3.1.	Captura de una intersección en una red de tráfico.	32
3.2.	Diagrama de una intersección controlada por un programa de ciclos.	33
3.3.	Diagramas fundamentales: ejemplos de la relación entre tasa de flujo, densidad y velocidad.	36
3.4.	Mecanismos de la contaminación atmosférica.	46
3.5.	Capa de smog sobre la ciudad de São Paulo (Brasil)	49
4.1.	Procedimiento general utilizado por los algoritmos evolutivos para evaluar los individuos de la población.	56
4.2.	Captura de una simulación en SUMO-GUI.	57
4.3.	Proceso de abstracción de la red de tráfico.	59
4.4.	Corrección de conexiones y cruces en una intersección de una red SUMO.	60
4.5.	Conexiones y estados del programa de ciclos en una intersección de una red SUMO.	61
4.6.	Imagen aérea de una intersección compleja estudiada y la correspondiente red SUMO durante una simulación.	62
4.7.	Esquema de la estrategia desarrollada para la generación de tráfico SUMO a partir de trazas de GPS	64
4.8.	Diagrama de componentes para el programa PowerGen.	65
4.9.	Ejemplo de mapa OSM y su correspondiente ruteable.	66
4.10.	Ejemplo de ajuste de traza GPS utilizando MatchGPX2OSM	66
4.11.	Diagrama de componentes para PriorWeight.	67
4.12.	Diagrama de componentes para TrafficGenerator.	68

4.13. Diagrama de los componentes principales en la resolución del problema de planificación de ciclos de semáforos.	69
4.14. Ejemplo de un programa de ciclos, representación y simulación SUMO.	71
5.1. Zonas de la ciudad de Montevideo elegidas para la evaluación de los algoritmos.	74
5.2. Trazas de GPS de la compañía CUTCSA dispuestas sobras las zonas de evaluación.	76
5.3. Trazas de GPS Flotilla dispuestas sobras las zonas de evaluación.	76
5.4. Instancias utilizadas para la calibración de los algoritmos.	78
5.5. Red SUMO para el escenario Aguada.	80
5.6. Frentes de Pareto para la instancia Aguada/mañana.	82
5.7. Frentes de Pareto para la instancia Aguada/tarde.	82
5.8. Frentes de Pareto para la instancia Aguada/noche.	82
5.9. Frentes de Pareto alternativos para la instancia Aguada/mañana.	85
5.10. Frentes de Pareto alternativos para la instancia Aguada/tarde.	85
5.11. Frentes de Pareto alternativos para la instancia Aguada/noche.	85
5.12. Capturas de simulaciones particulares para la instancia Aguada/tarde.	86
5.13. Red SUMO para el escenario Jacinto Vera.	87
5.14. Frentes de Pareto para la instancia Jacinto Vera/mañana.	89
5.15. Frentes de Pareto para la instancia Jacinto Vera/tarde.	89
5.16. Frentes de Pareto para la instancia Jacinto Vera/noche.	89
5.17. Capturas de simulaciones particulares para la instancia Jacinto Vera/tarde.	90
5.18. Red SUMO para el escenario Centro.	91
5.19. Frentes de Pareto para la instancia Centro/mañana.	94
5.20. Frentes de Pareto para la instancia Centro/tarde.	94
5.21. Frentes de Pareto para la instancia Centro/noche.	94
5.22. Frentes de Pareto alternativos para la instancia Centro/mañana.	95
5.23. Frentes de Pareto alternativos para la instancia Centro/tarde.	95
5.24. Frentes de Pareto alternativos para la instancia Centro/noche.	95
5.25. Capturas de simulaciones particulares para la instancia Centro/tarde.	97
A.1. Interfaz gráfica de PowerGen.	105

Índice de cuadros

3.1. Resumen de trabajos relacionados al problema de planificación de ciclos de semáforos	52
5.1. Resultados de validación para el estimador de volumen vehicular ranking.	77
5.2. Resumen de la calibración paramétrica aplicada a los algoritmos. . .	79
5.3. Porcentaje de mejora de AEMO y configuración real para la instancia Aguada.	83
5.4. Mejora de la técnica onda verde para la instancia Aguada.	84
5.5. Porcentaje de mejora de la configuración elegida sobre la real para la instancia Aguada/tarde.	86
5.6. Porcentaje de mejora de AEMO sobre la configuración real para la instancia Jacinto Vera.	88
5.7. Porcentaje de mejora de la configuración elegida sobre la real para la instancia Jacinto Vera/tarde.	90
5.8. Porcentaje de mejora de AEMO sobre la configuración real para la instancia Centro.	92
5.9. Mejora de la técnica onda verde para la instancia Centro.	93
5.10. Mejora de la técnica onda verde + DCI para la instancia Centro. . .	96
5.11. Porcentaje de mejora de la configuración elegida sobre la real para la instancia Centro/mañana.	96
C.1. Resultados de la calibración paramétrica etapa 1, NSGA-II	110
C.2. Resultados de la calibración paramétrica etapa 1, GDE3	110
C.3. Resultados de la calibración paramétrica etapa 1, MOEA/D	110
C.4. Resultados de la calibración paramétrica etapa 2, NSGA-II	111
C.5. Resultados de la calibración paramétrica etapa 2, GDE3	112
C.6. Resultados de la calibración paramétrica etapa 2, MOEA/D	113
C.7. Resultados de la calibración paramétrica etapa 3, ϵ -NSGA-II	114
D.1. Comparación entre los AEMO para la instancia Aguada/ <i>mañana</i> . .	115
D.2. Comparación entre los AEMO para la instancia Aguada/ <i>tarde</i> . . .	116
D.3. Comparación entre los AEMO para la instancia Aguada/ <i>noche</i> . . .	116
D.4. Comparación entre los AEMO para la instancia Jacinto Vera/ <i>mañana</i>	117
D.5. Comparación entre los AEMO para la instancia Jacinto Vera/ <i>tarde</i>	117

D.6. Comparación entre los AEMO para la instancia Jacinto Vera/ <i>noche</i>	117
D.7. Comparación entre los AEMO para la instancia Centro/ <i>mañana</i> . .	118
D.8. Comparación entre los AEMO para la instancia Centro/ <i>tarde</i> . . .	118
D.9. Comparación entre los AEMO para la instancia Centro/ <i>noche</i> . . .	118

Índice de algoritmos

1.	Pseudocódigo de un algoritmo evolutivo general	7
2.	Pseudocódigo de un algoritmo genético simple	8
3.	Pseudocódigo del algoritmo NSGA-II	16
4.	Pseudocódigo del algoritmo GDE3	18
5.	Pseudocódigo del algoritmo MOEA/D-DE	20

Capítulo 1

Introducción

El tráfico vehicular se ha convertido en la principal fuente de contaminación ambiental y congestión en zonas urbanas. Enfoques tradicionales han estudiado el problema de planificación de ciclos de semáforos simulando el tráfico con herramientas macroscópicas o analíticas, sin considerar el comportamiento individual del conductor (Robertson et al., 1980). Recientemente, se han desarrollado aplicaciones para el modelado del tráfico y las emisiones en forma microscópica. Estos modelos han sido acoplados para estimar las emisiones instantáneas de monóxido de carbono (CO), dióxido de carbono (CO₂), óxido de nitrógeno (NO_x) y smog, basadas en las actividades segundo a segundo de vehículos con comportamiento individual (Stevanovic et al., 2009).

El continuo crecimiento del tráfico provoca preocupación sobre el impacto de las emisiones vehiculares en la salud humana y en la calidad del aire, impulsando la demanda de un sistema normativo coherente para la gestión del tráfico, establecimiento de parámetros para calidad del aire y emisiones a nivel urbano a escalas regionales y nacionales (Costabile and Allegrini, 2008). La gran cantidad de semáforos operativos y la multiplicidad de combinaciones posibles de los mismos hace que la planificación conjunta sea un problema cada vez más complejo.

Por otra parte, los algoritmos evolutivos han demostrado su capacidad en la resolución de problemas complejos de diversa índole (Coello Coello et al., 2007; Nesmachnow, 2014), y en particular, para la optimización del tráfico urbano (Nishuichi and Yoshii, 2005; Mussone et al., 2010; Olivera et al., 2015). La complejidad asociada al problema del tráfico y su tratamiento realista motiva su estudio con técnicas que permitan disminuir el tiempo total de cómputo de los algoritmos evolutivos utilizados.

El proyecto que se describe en este informe surge de una iniciativa colaborativa e interdisciplinaria entre el Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET, Argentina), ANII (Agencia Nacional de Investigación e Innovación, Uruguay) y el Centro de Cálculo de la Universidad de la República (Uruguay). En el mismo se considera necesaria la generación de escenarios realistas de tráfico y la obtención de programas de ciclos de semáforos que permitan no solo minimizar el tiempo perdido de los vehículos en sus viajes, sino también

reducir las emisiones vehiculares en pos de mejorar la calidad del aire de zonas específicas de la ciudad de Montevideo (Uruguay).

Los principales aportes de este proyecto se mencionan a continuación:

- la integración de algoritmos evolutivos con un modelo microscópico de tráfico, con el objetivo de minimizar las emisiones vehiculares simultáneamente a la optimización del flujo de tráfico;
- el desarrollo de una estrategia confiable para obtener información actualizada de los patrones de viaje en una región, utilizando trazas de GPS; y
- la incorporación de técnicas de paralelismo para reducir el costo computacional de los algoritmos desarrollados al trabajar sobre instancias realistas.

En el contexto del proyecto se desarrollaron algoritmos evolutivos multiobjetivo paralelos que consideran la planificación de ciclos de semáforos como forma de mejorar el flujo vehicular y reducir las emisiones contaminantes de los automóviles. Para el análisis realista del tráfico se desarrolló una metodología de estimación de matrices origen-destino utilizando trazas de GPS. Estas matrices son utilizadas para conocer el flujo vehicular dentro de la red de tráfico. Los resultados obtenidos muestran como influye la planificación de ciclos de semáforos dentro de la red de tráfico a tal punto de posibilitar el mejoramiento de las condiciones del tráfico y reducir las emisiones vehiculares.

El resto del informe se organiza de la siguiente manera. El Capítulo 2 introduce los conceptos básicos de la computación evolutiva. Se proporciona una visión general de la optimización evolutiva multiobjetivo, analizando en detalle algoritmos evolutivos multiobjetivo del estado del arte, métricas y pruebas para evaluar el desempeño de los mismos y las técnicas de paralelización más importantes. La planificación de ciclos de semáforos y su relación con el flujo vehicular y las emisiones vehiculares se presenta en el Capítulo 3. El Capítulo 4 describe la estrategia de resolución utilizada. El análisis experimental y los resultados se reportan en el Capítulo 5. Por último, las conclusiones y el trabajo futuro son expuestos en el Capítulo 6.

Capítulo 2

Computación Evolutiva

El principal propósito de este capítulo es proporcionar una visión general del campo conocido como *optimización evolutiva multiobjetivo* que refiere al uso de algoritmos evolutivos para resolver problemas de optimización multiobjetivo.

2.1. Introducción

En muchas aplicaciones se debe hacer frente a problemas de optimización complejos con restricciones complicadas. Estos problemas pueden ser difíciles de resolver debido a su complejidad, no linealidad y su, potencialmente, alta dimensionalidad. Estos problemas pueden ser incluso NP-difíciles y requerir métodos de resolución alternativos a los métodos convencionales, que por lo general, no pueden hacer frente a este grado de complejidad.

Durante las décadas de 1960 y 1970 Fogel et al. (1966), Rechenberg (1973), Holland (1975) y Schwefel (1975) unificaron los principios fundamentales de la evolución biológica —herencia de genes, variación de genes en una población, traducción del genotipo en fenotipo y selección de los más aptos— en un paradigma de algoritmos que hoy se conoce como *computación evolutiva*. La evolución como concepto biológico denota la frecuencia del cambio de los genes en una población, de una generación a la siguiente (Curtis and Barnes, 1989). Como son los genes son los que definen la estructura y las características de un ser vivo, se puede decir que la evolución es el cambio de estas características en una población, provocado por mecanismos tales como la mutación y recombinación sexual. Se puede además utilizar el concepto de la evolución para explicar cómo las poblaciones se adaptan ante cambios en su entorno. A medida que los individuos cambian, aquellos que son más aptos se reproducirán más, y sus características se extenderán por toda la población. La ‘*supervivencia del más apto*’, en el sentido del principio Darwiniano (Darwin, 1859), explica cómo la evolución es utilizada por la naturaleza para encontrar “soluciones” a ciertos ambientes.

La computación evolutiva se ha aplicado ampliamente para encontrar soluciones a problemas de optimización, generación automática de programas y para llevar a cabo otras tareas en el campo de la inteligencia artificial, tales como aprendizaje

y formulación automática de hipótesis. Hoy en día, la computación evolutiva se compone de muchas variantes de algoritmos que son estructuralmente similares, pero se especializan en determinadas características del dominio de búsqueda. A estos algoritmos se les denomina colectivamente como *algoritmos evolutivos* (AE).

A continuación, en la Sección 2.2 se presentan los principios básicos de la computación evolutiva seguido por una breve descripción de los diferentes tipos de AE (algoritmos genéticos, estrategias evolutivas, programación evolutiva y evolución diferencial) en la Sección 2.3. En la sección 2.4 se introducen conceptos básicos en el campo de la optimización evolutiva multiobjetivo y se describen con detalle cuatro AE multiobjetivo populares y del estado del arte. La Sección 2.6 se centra en el estudio de rendimiento para algoritmos multiobjetivo. Por último, en la Sección 2.5 se estudian técnicas de paralelización con el objetivo de mejorar la eficiencia computacional de los AE multiobjetivo.

2.2. Conceptos básicos de la Computación Evolutiva

La idea esencial de la computación evolutiva es la exploración aleatoria del espacio de búsqueda con una población de soluciones candidatas al problema que se desea resolver, las cuales evolucionan en cada iteración del algoritmo. En los AE, las soluciones candidatas son llamadas *individuos* y las iteraciones se conocen como *generaciones*. A cada individuo se le asocia una aptitud (o *fitness*) mediante la cual se evalúa su propia supervivencia en cada generación. Al conjunto de individuos de una generación se le conoce como *población* del algoritmo. La transición de una generación a la siguiente se descompone generalmente en varias etapas. La primera consiste en seleccionar un conjunto de individuos candidatos de la población (a esta etapa se la denomina *selección*); luego, los individuos seleccionados atraviesan un proceso que consiste en la sucesiva aplicación de reglas de modificación (*operadores evolutivos*), terminando en la creación de uno o más individuos (*descendientes* o hijos). En un primer paso, las soluciones existentes son diversificadas con variación aleatoria. Luego, se seleccionan las mejores soluciones y estas proporcionan una dirección para la búsqueda. Los hijos reemplazan a sus padres total o parcialmente de acuerdo a diferentes esquemas de reemplazo, constituyendo así la población de la siguiente generación. Por último, el algoritmo evoluciona hasta que alcanza una condición de finalización.

Los datos manejados por un AE se estructuran en dos capas, *fenotipo* y *genotipo*. Haciendo una analogía con los cromosomas en una célula, el genotipo se corresponde con el “código genético” y es análogo a los genes, mientras que el fenotipo expresa los comportamientos globales y las estructuras que acompañan a la manifestación del genotipo en el medio ambiente. El genotipo es un conjunto de bajo nivel y objeto de acción de los operadores genéticos. El valor de fitness queda determinado únicamente por el fenotipo.

A continuación se definen algunos conceptos básicos de la computación evolutiva:

- **Exploración y explotación:** la mayoría de los métodos de computación evolutiva comparten los principios de *exploración* y *explotación*. Exploran el dominio de búsqueda de modo de encontrar el óptimo mediante los operadores de *mutación* y *cruzamiento*. La mutación diversifica las soluciones candidatas existentes con medios de variación aleatoria, mientras que el cruzamiento o recombinación, produce nuevos candidatos a solución combinando las características de al menos dos soluciones. El cruzamiento también es parte esencial del principio de explotación, explotando el conocimiento descubierto sobre el desempeño de los individuos. Los operadores de selección logran este objetivo eligiendo los individuos más aptos para la reproducción (selección de padres) o permitiendo que estos sobrevivan (selección de supervivencia).
- **Variables:** el genotipo, o bien la representación de una solución candidata, consiste de una cadena de valores de largo fijo sobre la que trabaja la heurística de optimización. Los algoritmos genéticos canónicos trabajan con cadenas de valores binarios, por ej. $\mathbf{b} = (b_1, \dots, b_n) \in \{0, 1\}^N$ donde b_i denota un *gen* y N es el tamaño del genotipo o cromosoma. Una población P consiste de un conjunto de genotipos. Para un gen $b_i \in \{0, 1\}$, el índice i denota el *lugar* o *locus* de b_i , mientras que los valores potenciales 0 y 1 son los *alelos* de b_i . Otras representaciones numéricas o continuas (utilizadas con menor frecuencia) trabajan con cadenas de valores reales $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbb{R}^N$.
- **Operadores evolutivos:** se trata de aquellos procedimientos que manipulan la información genética de la población para producir nuevos individuos a partir de la población original. Su propósito es introducir diversidad genética o mejorar las soluciones generando lo que se conocen como descendientes. Existen diversos tipos de operadores evolutivos que están diseñados para distintos tipos de representaciones, así como características del espacio de búsqueda. Los operadores más importantes son mutación y cruzamiento. Cada operador tiene un parámetro asociado que controla la probabilidad de su aplicación en cada generación.
 - **Mutación:** la mutación es la fuente de los cambios evolutivos y se realiza perturbando uno o más alelos del cromosoma de un individuo generando un individuo ligeramente distinto pero relacionado con el progenitor. Permite alcanzar una región del espacio de búsqueda diferente a la del individuo original, lo que ayuda a que la búsqueda no se estanque en un óptimo local. El papel de este operador depende del tipo de algoritmo en el que se incluya. Por ejemplo, en los algoritmos genéticos es el encargado de contribuir a la diversidad de la población mientras que en la programación evolutiva es el único operador encargado de toda la búsqueda. Otra característica importante es que el operador de mutación es el responsable de garantizar que cualquier solución del espacio sea alcanzable

(Eiben and Smith, 2007). En representaciones binarias, la mutación se corresponde con la inversión de un bit con determinada probabilidad (probabilidad o tasa de mutación). Para representaciones distintas se deben diseñar otros operadores de mutación. Por ejemplo, para espacios de búsqueda numéricos, la mutación puede ser implementada como la adición de un número aleatorio. En programación genética, la mutación es el intercambio de un subárbol por otro subárbol aleatorio. Tradicionalmente, la mutación se considera como un operador secundario en los algoritmos genéticos.

- **Cruzamiento:** el cruzamiento tiene como objetivo la herencia de material genético a partir de al menos dos soluciones parentales. Los genes de los padres se combinan para formar las cadenas cromosómicas de sus descendientes. El principio detrás del cruzamiento es combinar dos o más individuos con características deseadas para producir uno o mas individuos que combinen tales cualidades, buscando preservar los buenos genes y obtener mejores soluciones al problema. Su función depende del tipo de algoritmo en el que se incluya. Varios operadores de recombinación han sido propuestos en la literatura. Los más conocidos son los de un punto, dos puntos, múltiples puntos y uniforme.
 - **Selección:** existen dos tipos de selección. La *selección parental* es un proceso que da dirección a la búsqueda evolutiva determinando que individuos seleccionados son adecuados para engendrar la siguiente generación de individuos. Mientras más alto sea el valor de la función de desempeño de un individuo, más oportunidades tendrá de reproducirse. Esta selección debe ser balanceada con los operadores de variación (mutación y recombinación). Si la selección es muy fuerte, individuos de calidad subóptima pueden dominar la población y por lo tanto reducir la diversidad genética requerida para el cambio y el avance; por otro lado, una selección muy débil puede resultar en un proceso evolutivo muy lento. Varios esquemas de selección han sido propuestos en la literatura: *selección proporcional*, *por escalamiento*, *por torneo*, *basada en rango*, etc. No existe consenso respecto a cual esquema es preferible, aunque algunos son más sencillos conceptualmente que otros, y por tanto más fáciles de implementar.
- El segundo tipo de selección es la *selección de supervivencia*. Un vez creada la nueva población temporal consistente de la población actual junto con los nuevos individuos creados, se seleccionan aquellos individuos que formaran parte de la población siguiente.
- **Función de fitness:** la adaptabilidad de un individuo se evalúa mediante una *función de fitness* (también conocida como función de evaluación, función objetivo o función de aptitud). El propósito de esta función es medir la capacidad del individuo para resolver el problema bajo estudio.

- **Condición de parada:** la terminación de un AE depende de los requerimientos definidos por el usuario. Las condiciones de parada típicas son: convergencia del fitness (no se observan mejoras luego de un número predefinido de generaciones), máximo número de generaciones, máximo número de llamadas a la función de fitness y límite del tiempo de ejecución.

2.2.1. Estructura general de un AE

En resumen, un AE se puede describir de la siguiente manera:

1. La motivación del algoritmo es buscar el máximo valor de una función dada $f : \Omega \rightarrow \mathbb{R}$ (función de fitness). Cada elemento del conjunto Ω es un individuo, y el valor de $f(\cdot)$ determina el éxito o adaptabilidad de ese individuo en su medio ambiente.
2. El algoritmo es iterativo y en cada generación t produce un subconjunto de Ω , determinando la población en la generación t . Se necesita por tanto de una regla que establezca como generar la población inicial (en la generación 0) y de una condición de parada que especifique en qué momento deben terminar las iteraciones.
3. La composición de la población en la generación t se determina a partir de la población de la generación $t - 1$ de acuerdo a un proceso estocástico que traduce en operaciones matemáticas los procesos biológicos de reproducción, mutación y recombinación. Los miembros de la generación t se consideran los padres y los de la generación $t + 1$ los hijos. El número de individuos en una población puede aumentar o disminuir en cada iteración dependiendo del algoritmo. El pseudocódigo de un AE general se muestra en el Algoritmo 1.

Algoritmo 1: Pseudocódigo de un algoritmo evolutivo general

```

 $t \leftarrow 0$ 
Inicializar_poblacion( $P_t$ )
Evaluar( $P_t$ )
mientras no CriterioParada hacer
     $P'_t \leftarrow \text{Selección}(P_t)$ 
     $P''_t \leftarrow \text{Variación}(P'_t)$ 
    Evaluar( $P''_t$ )
     $P_{t+1} \leftarrow \text{Reemplazo}(P''_t, P_t)$ 
     $t \leftarrow t + 1$ 

```

Existen varias clases de AE, en parte con diferentes antecedentes históricos. Se diferencian en el tipo de representación, operadores genéticos y dominios de problemas. En la siguiente sección se presentan los principales paradigmas.

2.3. Principales paradigmas de la Computación Evolutiva

Clasificar los algoritmos evolutivos no es una tarea simple y se han propuesto varios criterios distintos. Sin embargo, por razones sobre todo históricas, suele hablarse de tres paradigmas principales que originaron la familia de los AE: *algoritmos genéticos*, *estrategias evolutivas* y *programación evolutiva*.

En los algoritmos genéticos los operadores evolutivos son utilizados para cambiar y mejorar una población de cromosomas abstractos que representan soluciones a un problema. Relacionados a los algoritmos genéticos se encuentran las estrategias evolutivas, las cuales están más enfocadas en la mutación. En la programación evolutiva el foco está en la interacción entre ‘especies’ más que entre individuos como en los otros paradigmas. Por otro lado, en años recientes se han propuesto y explorado nuevos paradigmas dentro de la computación evolutiva. Entre estos paradigmas se destaca la *evolución diferencial*, enfocada en la mutación y utilizada para resolver problemas de optimización en espacios continuos. Las siguientes secciones dan una breve reseña de estos paradigmas.

2.3.1. Algoritmos genéticos

Los principios básicos de los algoritmos genéticos (AG) fueron establecidos por John Holland a principios de los años 1960 (Holland, 1962b,a), motivado por su interés en resolver problemas de aprendizaje de máquina, y se encuentran descritos en varios textos (Goldberg, 1989; Davis, 1991; Michalewicz, 1992; Reeves, 1993).

En el AG propuesto por Holland, la población está formada por cromosomas cuya estructura es una cadena de *bits* de tamaño fijo. Hoy en día el alfabeto utilizado para representar los individuos es natural al problema y no está necesariamente restringido a cadenas binarias ($\{0, 1\}^*$), aunque buena parte de la teoría en la que se fundamentan los AG utiliza dicho alfabeto. El Algoritmo 2, muestra el pseudocódigo del llamado *algoritmo genético simple* o *canónico*.

Algoritmo 2: Pseudocódigo de un algoritmo genético simple

```

 $t \leftarrow 0$ 
Inicializar_poblacion( $P_t$ )
Evaluar( $P_t$ )
mientras no CriterioParada hacer
    Padres  $\leftarrow$  Selección( $P_t$ )
    Hijos  $\leftarrow$  Cruzamiento(Padres)
     $P'_t \leftarrow$  Mutación(Hijos)
    Evaluar( $P'_t$ )
     $P_{t+1} \leftarrow$  Reemplazo( $P_t$ ,  $P'_t$ )
     $t \leftarrow t + 1$ 

```

Los AG comienzan con una población inicial de individuos usualmente elegidos aleatoriamente dentro del espacio de búsqueda. La transición de una generación a la siguiente consiste en un proceso de reproducción, de naturaleza estocástica, articulado en tres fases: selección (probabilística), cruzamiento (operador principal) y mutación (operador secundario). La mutación puede llegar a ser un operador opcional o estar ausente en algunas variantes de AG que utilizan otros operadores para introducir diversidad.

El paso de selección resulta en la elección de los individuos de la población actual que se van a reproducir. La selección se realiza de forma tal que aquellos individuos mejor adaptados se reproduzcan con más frecuencia. Para ello, se utiliza un procedimiento de forma tal que, la probabilidad de que un individuo sea elegido para reproducción, sea mayor al ser mayor su valor de fitness. La forma más sencilla de conseguir esto es mediante un procedimiento conocido como *selección proporcional*. Este consiste en crear una ruleta con todos los individuos. Cada individuo de la población obtiene una superficie en la ruleta proporcional a su valor de fitness (ruleta ponderada). La selección se realiza mediante la rotación de la rueda y la elección del individuo ubicado en el lugar donde se detuvo la rueda. Otros procedimientos populares son: muestreo estocástico universal y selección por torneo.

Una vez que se eligieron los candidatos para reproducción, la fase de cruzamiento considera pares de individuos elegidos al azar. Estos se aparean y producen un par de hijos que comparten genes de ambos padres. El cruzamiento se aplica con probabilidad fija (normalmente comprendida entre 0.5 y 1.0) a cada par. Se utilizan varios tipos de cruzamiento, siendo el más simple, el cruzamiento por un punto, extensible a n puntos. En la recombinación de un punto, un único punto de corte es seleccionado aleatoriamente en los progenitores; luego los segmentos antes y después del punto de corte son intercambiados. La recombinación de n puntos es una generalización de esta idea que introduce varios puntos de corte e intercambia los segmentos entre dichos puntos. En la recombinación uniforme, los segmentos intercambiados se reducen a bits únicos. En lugar de puntos de corte, se sortea aleatoriamente de cual progenitor proviene cada bit (ver Figura 2.1). En el caso de que el operador no se aplique, la descendencia se obtiene simplemente duplicando los padres.

Una vez generados los hijos, son sometidos al operador de mutación. Al igual que sucede en la naturaleza, la mutación es considerada un evento extraño, por lo que se utiliza una probabilidad pequeña. Cada gen de cada cromosoma, es sometido a mutación con dicha probabilidad, independientemente de los demás genes. Si se trata de un gen binario, la mutación simplemente cambia un 0 por un 1 o viceversa, mientras que si el alfabeto es más complejo, se deben definir las reglas de mutación. Se suele realizar una elección uniforme entre todos los posibles símbolos del alfabeto, en caso de que este sea finito, o una perturbación basada en una variable gaussiana en caso de tratarse de números reales.

Para la selección de supervivencia o estrategia de reemplazo se asume una población con número de individuos constante igual a μ . Una forma de proceder es generando μ hijos y reemplazando la totalidad de la población en cada generación

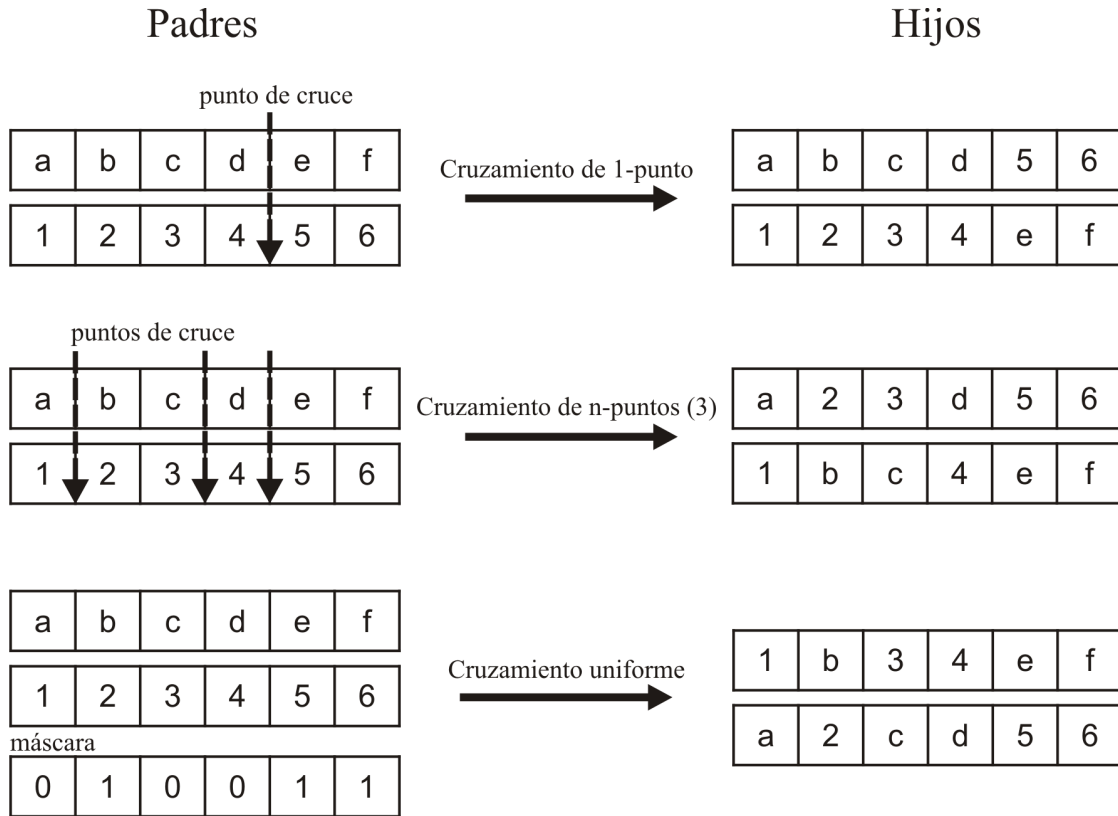


Figura 2.1: Cruzamiento en un AG: de 1-punto, n-puntos y cruzamiento uniforme.

(modelo generacional). Sin embargo, esto puede llevar a la eliminación de aquellos individuos más aptos por lo que se suelen aplicar técnicas de elitismo (preservación del individuo más apto). Otra opción es reemplazar solo una fracción de la población actual, o incluso, uno solo de los individuos en cada generación (modelo de estado estacionario (Whitley, 1989)).

Si el AG ha sido correctamente implementado, la población evolucionará a lo largo de las generaciones sucesivas de tal manera que la adaptación media extendida a todos los individuos de la población, así como la adaptación del mejor individuo se irán incrementando hacia el óptimo global (De Jong, 1995).

2.3.2. Estrategias evolutivas

Las estrategias evolutivas fueron inventadas por Rechenberg y Schwefel con el objetivo de resolver problemas hidrodinámicos de alto grado de complejidad (Rechenberg, 1973; Schwefel, 1977).

En su forma original, una estrategia evolutiva es un algoritmo de optimización de una función real de M variables $f : \mathbb{R}^M \rightarrow \mathbb{R}$. Una población de vectores en $\mathbb{R}^M$ son evolucionados teniendo como función de fitness a f , en caso de maximización o $-f$ de lo contrario. El algoritmo es iterativo y comienza con una población de individuos elegidos aleatoriamente en el espacio de soluciones. Como se indica en la Ecuación 2.1, en cada generación se generan descendientes seleccionando un

individuo padre y perturbándolo mediante la adición de un vector de números gaussianos independientes con una media cero y desviación estándar σ ($N(0, \sigma)$),

$$x^{t+1} = x^t + N(0, \sigma) \quad (2.1)$$

donde t se refiere a la generación (o iteración) en la que se encuentra el algoritmo.

Una vez que se crearon todos los hijos, la población de la siguiente generación es formada seleccionando aquellos individuos más aptos de acuerdo a las estrategias $(\mu + \lambda)$ -ES ó (μ, λ) -ES, donde μ es el número (constante) de individuos en la población y λ es el número de hijos en cada generación:

- $(\mu + \lambda)$ -ES: μ padres crean λ hijos, luego todos los individuos $(\mu + \lambda)$ son ranqueados y los mejores μ son elegidos para constituir la próxima población de individuos.
- (μ, λ) -ES: λ hijos son generados por μ padres, $\lambda > \mu$, y solo los mejores μ mas aptos de los λ hijos constituyen la próxima población por lo que todos los individuos desaparecen en cada iteración.

Si se considera $1 \leq \mu \leq \lambda$ la estrategia $(\mu + \lambda)$ -ES es marcadamente elitista y puede perjudicar el proceso de adaptación a cambios en la topología del espacio viable. La estrategia (μ, λ) -ES se indica como más robusta principalmente cuando se utilizan poblaciones reducidas.

En años recientes se han introducido un número importante de extensiones y generalizaciones, por lo que las actuales estrategias evolutivas empleadas están mucho más elaboradas. La mayoría implementa técnicas de *auto-adaptación* y *recombinación* (Baragona et al., 2011).

2.3.3. Programación evolutiva

La programación evolutiva fue propuesta originalmente por Fogel, Owens y Walsh (Fogel et al., 1966), con el objetivo de generar un enfoque alternativo a la inteligencia artificial. En lugar de emular o simular seres humanos, ya sea en su estructura neurofisiológica o sus comportamientos particulares, se modelo la evolución natural como un proceso que genera organismos cuyo intelecto mejora con el tiempo. Se define inteligencia como la habilidad de un organismo para lograr objetivos en diversos entornos y se considera al comportamiento inteligente como la habilidad de predecir ocurrencias en un ambiente dinámico, mejorando la auto-adaptación. Para formalizar dicho proceso se elaboraron una serie de experimentos en los que se evolucionaban autómatas de estado finito utilizados para la predicción de secuencias de símbolos.

La programación evolutiva es una abstracción de la evolución al nivel de las especies, por lo que no se requiere el uso de un operador de recombinación. Inicialmente se genera una población de autómatas generados en forma aleatoria que son expuestos por un determinado tiempo al ambiente (reciben secuencias de símbolos de entrada). Luego, se produce la generación siguiente mediante mutación de cada

autómata presente. Los autómatas hijos son evaluados de acuerdo una función de fitness (de recompensa) que determina la cantidad de símbolos predichos correctamente. Se seleccionan para la próxima etapa la mitad de los mejores genitores y la mitad de los mejores descendientes, lo que da origen a una estrategia del tipo (μ, μ) en terminología de estrategias evolutivas.

Este procedimiento es bastante general y puede ser adaptado a muchos problemas diferentes dado que la representación de cada individuo como autómata de estado finito es bastante flexible. Un algoritmo de programación evolutiva típico para representaciones con valores reales trabaja con vectores y perturbaciones Gaussianas como medio de mutación. Usualmente, la selección de padres se realiza en forma determinista mientras que la selección de supervivientes se realiza mediante reglas estocásticas.

La programación evolutiva ha sido propuesta como base de la evolución de programas dando surgimiento a una nueva disciplina conocida como *programación genética*. La idea es evolucionar programas por medio de los operadores genéticos tales como mutación, cruzamiento y selección, de los modelos más exitosos cualitativamente. La representación más utilizada son los árboles, pero los programas también se pueden representar en lenguaje de máquina como lenguaje ensamblador (Koza (1992)).

2.3.4. Evolución diferencial

La evolución diferencial, desarrollada por Rainer Storn y Kenneth Price a mediados de la década de 1990 (Storn and Price, 1995; Storn, 1996), es una heurística relativamente reciente, utilizada principalmente para resolver problemas de optimización en espacios continuos. La evolución diferencial comparte muchas similitudes con los AE tradicionales, como ser el uso de una población inicial aleatoria, que es luego mejorada mediante las fases de selección, mutación y cruzamiento. Sin embargo, a diferencia de los AG que suelen utilizar representación binaria, cada individuo es codificado mediante un vector real $x_i \in \mathbb{R}^M$ al que se le aplica mutación basada en la distribución de las soluciones en la población actual.

La transición de una generación a la siguiente se consigue tratando a cada individuo en forma separada y produciendo un hijo que lo reemplace en la siguiente generación. En cada generación, cada individuo x_i es sometido al *proceso de evolución diferencial*: primero se eligen aleatoriamente tres individuos v_0, v_1 y v_2 , distintos entre sí y distintos de x_i y se crea un nuevo individuo mutante u (individuo de prueba) como se indica en la Ecuación 2.2.

$$u = v_0 + F(v_1 - v_2). \quad (2.2)$$

El peso diferencial F tiene un valor fijo elegido por el implementador (normalmente en el rango $(0, 1+]$). El individuo u es luego recombinado con x_i mediante cruzamiento uniforme con probabilidad $CR \in [0, 1]$. El cruzamiento puede ser realizado de forma binomial o exponencial. Finalmente, el individuo original y el hijo son comparados y aquel con mayor fitness es retenido e ingresado en la población

de la siguiente generación. El proceso de evolución diferencial se repite para cada individuo en la población.

Existe un esquema de nomenclatura desarrollado para referenciar a las distintas variantes de la evolución diferencial. La más popular se conoce como “DE/rand/1/bin” donde “DE” significa evolución diferencial por sus siglas en inglés, “rand” indica que los individuos elegidos para calcular la mutación son elegidos en forma aleatoria, “1” es el número de pares de soluciones escaladas (v_1, v_2) y finalmente “bin” significa que se utiliza cruzamiento binomial. Existen más de 10 variantes distintas de la evolución diferencial que pueden ser encontradas por el lector en Price et al. (2006).

2.4. AE para optimización multiobjetivo

A continuación se introducen los AE para optimización multiobjetivo que serán utilizados luego en la resolución de la problemática planteada en el capítulo 3.

2.4.1. Optimización multiobjetivo

En muchas disciplinas del mundo real, a menudo se desea optimizar problemas con dos o más objetivos mutuamente contrapuestos. Por ejemplo, en finanzas, el rendimiento y riesgo son generalmente objetivos conflictivos. En problemas de diseño mecánico se pueden tener objetivos de rendimiento y costos de manufactura, con restricciones de recursos. Un *problema de optimización multiobjetivo* consiste en encontrar un vector de variables de decisión que satisfagan un conjunto de restricciones y optimice un conjunto de funciones objetivo, generalmente de naturaleza conflictiva. Se puede enunciar formalmente, sin pérdida de generalidad, como se indica en la Ecuación 2.3 (Rao, 1991).

$$\begin{aligned}
 &\text{Minimizar } f_i(\mathbf{x}), i = 1, 2, \dots, M \\
 &\text{Sujeto a:} \\
 &\quad g_j(\mathbf{x}) \leq 0, j = 1, 2, \dots, J \\
 &\quad h_k(\mathbf{x}) = 0, k = 1, 2, \dots, K
 \end{aligned} \tag{2.3}$$

Se busca un vector $\mathbf{x}^* \in \mathbb{R}^N$ que minimice las funciones objetivo f_i sin violar ninguna de las restricciones. Los problemas de maximización pueden ser convertidos en problemas de minimización y todas las restricciones se pueden expresar de la forma $g_j(\mathbf{x}) \leq 0$.

Típicamente un problema de optimización multiobjetivo es convertido a un problema de optimización monoobjetivo predefiniendo pesos que expresan la importancia relativa de cada objetivo. La optimización simultánea de los objetivos sin articular la importancia relativa no siempre resulta en una única solución debido a la naturaleza conflictiva de los objetivos. Por lo tanto, la noción de óptimo

cambia, y no se busca obtener una solución única sino un conjunto de soluciones que representan los mejores compromisos posibles entre los diferentes objetivos. A estas soluciones se les llama *óptimos de Pareto* o *Pareto eficientes*. Sea $\Omega = \{\mathbf{x} \in \mathbb{R}^N \mid h_k(\mathbf{x}) = 0, g_j(\mathbf{x}) \leq 0\}$ se consideran las siguientes definiciones:

Definición 1. Eficiencia. Una solución factible $\mathbf{x} \in \Omega$ es *óptimo de Pareto* o *Pareto eficiente* si y solo si $\nexists \mathbf{y} \in \Omega$ tal que $\forall i = 1, 2, \dots, M, f_i(\mathbf{y}) \leq f_i(\mathbf{x})$ con al menos una desigualdad estricta.

Definición 2. Dominancia de Pareto. Dadas dos soluciones $\mathbf{x}, \mathbf{y} \in \Omega$ tales que $\forall i = 1, 2, \dots, M, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, se dice que $\mathbf{x}$ domina débilmente a $\mathbf{y}$ y se denota $\mathbf{x} \preceq \mathbf{y}$. Si además $\exists r \in \{1, 2, \dots, M\} \mid f_r(\mathbf{x}) < f_r(\mathbf{y})$, se dice que $\mathbf{x}$ domina a $\mathbf{y}$ y se denota $\mathbf{x} \prec \mathbf{y}$.

Definición 3. Conjunto de Óptimos de Pareto. Dado un problema de optimización multiobjetivo $F = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$, se define el conjunto de óptimos de Pareto, P^{true} como:

$$P^{\text{true}} = \{\mathbf{x} \in \Omega \mid \nexists \mathbf{x}' \in \Omega, \mathbf{x}' \prec \mathbf{x}\} \quad (2.4)$$

Definición 4. Frente de Pareto. Dado un problema de optimización multiobjetivo $F = (f_1(\mathbf{x}), \dots, f_M(\mathbf{x}))$ y su conjunto de óptimos de Pareto, P^{true} , se define el frente de Pareto, PF^{true} , como:

$$PF^{\text{true}} = \{F(\mathbf{x}) \mid \mathbf{x} \in P^{\text{true}}\} \quad (2.5)$$

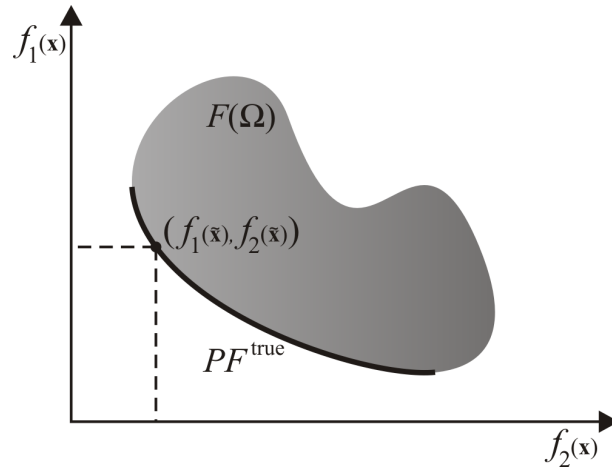


Figura 2.2: Ejemplo de frente de Pareto PF^{true} para $\Omega \subset \mathbb{R}^2$. $\tilde{\mathbf{x}}$ se corresponde con un óptimo de Pareto y $F(\Omega) = \{F(\mathbf{x}) \mid \mathbf{x} \in \Omega\}$.

En la Figura 2.2 se puede ver un ejemplo de frente de Pareto para un caso de dos dimensiones. En un problema de optimización multiobjetivo, se pretende encontrar la mayor cantidad de óptimos de Pareto posible (optimización de Pareto). Encontrar el frente de Pareto exacto para un problema arbitrario puede resultar

muy difícil por lo que generalmente se aceptan aproximaciones razonables para un determinado tiempo de computación. Una gran parte de las técnicas de optimización multiobjetivo están limitadas a frentes de Pareto con ciertas características (por ej., frentes convexos) y suelen requerir un punto de búsqueda inicial o generar una sola solución por ejecución.

2.4.2. Algoritmos evolutivos multiobjetivo

Los *algoritmos evolutivos multiobjetivo* (AEMO) son AE diseñados para resolver problemas de optimización multiobjetivo. Su popularización en años recientes es debida principalmente a su propiedad de búsqueda paralela, procesando en cada iteración una población de soluciones potenciales en lugar de una única solución al problema a resolver. Además, los AEMO no requieren que el problema cumpla condiciones de continuidad y/o derivabilidad en sus funciones objetivo, ni que éstas estén dadas en forma algebraica. Estas características les da a los algoritmos evolutivos una gran ventaja respecto a las técnicas clásicas de optimización.

La diferencia más importantes entre un AE y un AEMO radica en el esquema para determinar la adaptabilidad de los individuos. Mientras que en un AE se calcula una sola función de fitness, en un AEMO se calculan varias funciones de fitness (generalmente idénticas a las funciones objetivo). Sin embargo, el operador de selección espera un solo valor de aptitud. Por consiguiente, en un AEMO se necesita un proceso adicional para convertir este vector de valores de fitness en un escalar. En general se distinguen tres esquemas para llevar a cabo este proceso: basados en criterios, basados en agregación y basados en el concepto de dominancia de Pareto (ver Zitzler and Thiele (1999)).

Uno de los primeros AEMO fue el *Vector Evaluated Genetic Algorithm* (VEGA) desarrollado por Schaffer en su tesis doctoral (Schaffer, 1984). El VEGA no se basa en el concepto de dominancia de Pareto sino que divide la población en tantas subpoblaciones como objetivos haya y aplica la selección de acuerdo a un solo objetivo en cada subpoblación.

En la década de 1990 se realizaron una gran cantidad de propuestas de MOEAs basados en el concepto de no-dominancia (jerarquización de la población) y el uso de algún tipo de mecanismo para preservar la diversidad de las soluciones. Se destacan el *Multiple Objective Genetic Algorithm* (Fonseca and Fleming, 1993), *Non-Dominated Sorting Genetic Algorithm* (NSGA) (Srinivas and Deb, 1995), *Niched-Pareto Genetic Algorithm* (Horn et al., 1994) y el *Strength Pareto Evolutionary Algorithm* (SPEA) (Zitzler and Thiele, 1998, 1999). Todos estos, excepto el SPEA, son no-elitistas por lo que no mantienen, en forma obligatoria, la mejor solución encontrada durante el proceso evolutivo. A los AEMO no-elitistas también se les conoce como AEMO de primera generación. A estos le siguieron los AEMO elitistas, como el SPEA ya mencionado y su versión mejorada SPEA2 (Zitzler et al., 2002), NSGA versión II (NSGA-II) (Deb et al., 2002) y el *Pareto-Archived Evolution Strategy* (Knowles and Corne, 2000).

Por último, entre los AEMO más recientes destacan: *ϵ -dominance Multiobjective Evolutionary Algorithm* (Laumanns et al., 2002), *ϵ -dominance Non-Dominated*

Sorting Genetic Algorithm II (ϵ -NSGA-II) (Kollat and Reed, 2006), *Indicator-Based Evolutionary Algorithm* (Zitzler and Künzli, 2004), *Generalized Differential Evolution 3* (GDE3) (Kukkonen and Lampinen, 2005) y *Multiobjective Evolutionary Algorithm Based on Decomposition* (MOEA/D) (Zhang and Li, 2007). En lo que sigue discutiremos en profundidad los algoritmos NSGA-II, GDE3, ϵ -NSGA-II y MOEA/D. Los primeros dos fueron elegidos por ser algoritmos populares del estado del arte provenientes de paradigmas distintos que se han empleado con éxito por bastante tiempo. Por otro lado, se eligieron los algoritmos ϵ -NSGA-II y MOEA/D por representar una generación más reciente de AEMO.

2.4.3. NSGA-II

NSGA-II (Deb et al., 2002) es un AEMO elitista, del estado del arte y basado en ordenamiento no dominado, que ha sido aplicado con éxito en la resolución de problemas de optimización en múltiples áreas. NSGA-II es una evolución de NSGA siendo más eficiente en términos computacionales, además de incorporar un operador de *crowding* (aglutinamiento) para mantener la diversidad y utilizar un mecanismo elitista consistente en seleccionar las mejores soluciones generadas durante la búsqueda. El pseudocódigo de NSGA-II se muestra en el Algoritmo 3.

Algoritmo 3: Pseudocódigo del algoritmo NSGA-II

```

 $t \leftarrow 0$ 
Inicializar_poblacion( $P_0$ )
Evaluar( $P_0$ )
 $Q_0 \leftarrow$  Generar Hijos ( $P_0$ ) // selección por torneo y reproducción
mientras no CriterioParada hacer
     $F \leftarrow$  Sorting No Dominado( $P_t \cup Q_t$ ) // generar conjuntos no dominados
     $P_{t+1} \leftarrow \emptyset, i \leftarrow 0$ 
    mientras  $|P_{t+1}| + |F_i| \leq \#P$  hacer
        Asignar Distancia de Crowding ( $F_i$ )
         $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
     $i \leftarrow i + 1$ 
    Sorting por Distancia( $F_i$ )
     $P_{t+1} \leftarrow P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$ 
     $Q_{t+1} \leftarrow$  Generar Hijos ( $P_{t+1}$ ) // selección por crowding y reproducción
     $t \leftarrow t + 1$ 

```

Inicialmente se crea una población inicial de individuos (P_0) en forma aleatoria. Los individuos son evaluados y sobre ellos se aplican los operadores evolutivos de selección (torneo binario) y reproducción (cruzamiento y mutación) para crear un grupo de descendientes. En la generación t , se emplea una jerarquización de Pareto la cual consiste en clasificar los individuos en varias categorías. Dicha clasificación se lleva a cabo de la siguiente forma: en cada generación, la población de padres P_t y la población de hijos Q_t son combinadas en una única población $P_t \cup Q_t$. Si el

tamaño total de la población es de $\#P$, entonces la población combinada tiene un tamaño de $2 \times \#P$. La población combinada es luego ordenada mediante *sorting no-dominado*: el conjunto de individuos no dominados F_0 conforma el primer nivel en la jerarquización de Pareto. Posteriormente, se elimina el conjunto F_0 de la población combinada y se forma un nuevo conjunto F_1 con los individuos no dominados respecto a la población restante. Este conjunto constituye el segundo nivel en la jerarquía. El proceso continúa hasta que el número de individuos seleccionados sea igual a $\#P$.

Si al seleccionar el último conjunto no-dominado se sobrepasa el tamaño de la población, entonces este conjunto es reducido en base a una métrica de distancia entre individuos conocida como *distancia de crowding*. Esta distancia también se utiliza en el operador de selección (excepto para generar Q_0), lo que ayuda a mantener la diversidad en la población y mejora la exploración del espacio de búsqueda. Ver Figura 2.3.

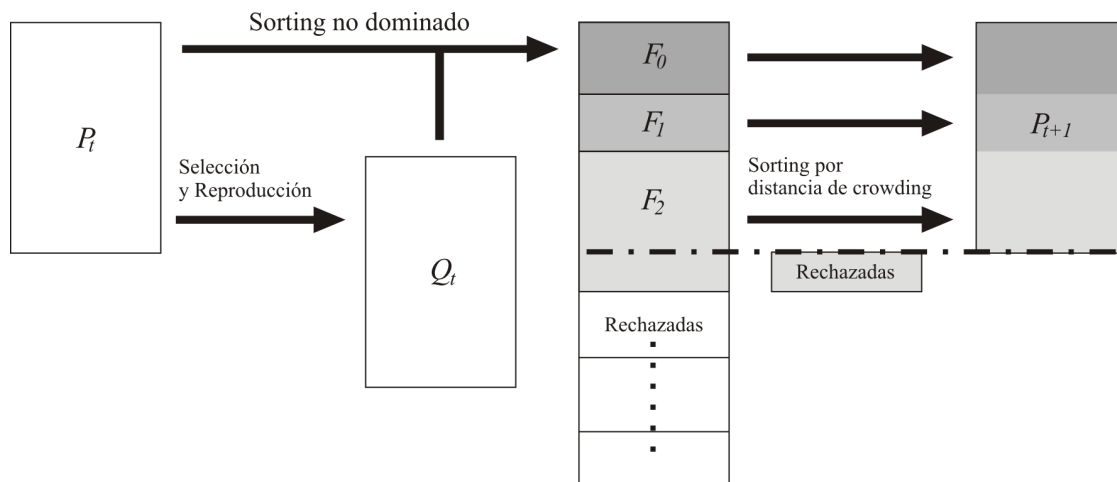


Figura 2.3: Proceso de determinación de nueva población en el algoritmo NSGA-II.

2.4.4. GDE3

Se han propuesto muchas extensiones de la técnica de evolución diferencial (ver sección 2.3.4) para optimización multiobjetivo con restricciones (Zhang et al., 2009; Li and Zhang, 2009). El algoritmo *Generalized Differential Evolution* (GDE) (Lampinen, 2001) extiende el algoritmo de evolución diferencial básico para un problema de optimización multiobjetivo, pudiendo ser aplicado a un número arbitrario de objetivos y restricciones. La regla de selección es modificada y consiste en seleccionar al individuo de prueba para la próxima generación si este domina débilmente al individuo padre, ya sea en el espacio de funciones objetivo o en el espacio de restricciones. No se incluye clasificación no dominada ni técnicas de preservación de diversidad.

GDE en su tercer versión, GDE3 (Kukkonen and Lampinen, 2005; Kukkonen and Deb, 2006), extiende el método “DE/rand/1/bin” a problemas con M objetivos

y K restricciones y ha mostrado ser superior a otros AEMO para un conjunto de problemas numéricos (Kukkonen and Lampinen, 2007; Tan, 2008). El pseudocódigo de GDE3 se muestra en el Algoritmo 4.

Algoritmo 4: Pseudocódigo del algoritmo GDE3

```

 $t \leftarrow 0$ 
Inicializar_población( $P_0$ )
Evaluar( $P_0$ )
mientras no CriterioParada hacer
     $P'_t \leftarrow P_t$ 
    para  $x_i$  en  $P'_t$  hacer
         $u_i \leftarrow \text{Mutación\_y\_cruzamiento}(x_i)$ 
        Evaluar( $u_i$ )
        si  $u_i \preceq x_i$  entonces
             $x_i \leftarrow u_i$ 
            si  $\forall j : g_j(u_i) \leq 0 \wedge x_i \not\prec u_i$  entonces
                 $P'_t \leftarrow P'_t \cup \{u_i\}$ 
     $P_{t+1} \leftarrow \text{Sorting\_No\_Dominado\_y\_Reducción}(P'_t)$ 
     $t \leftarrow t + 1$ 

```

Inicialmente se genera una población inicial de individuos (P_0) en forma aleatoria. Los individuos son evaluados para luego pasar por el proceso de evolución diferencial (mutación y cruzamiento) siguiendo lo visto en la sección 2.3.4. En la generación t , al igual que en el algoritmo NSGA-II (ver Sección 2.4.3), se realiza sorting no dominado utilizando el concepto de distancia de crowding. La diferencia radica en el tamaño de la población temporal. En NSGA-II se obtiene una población combinada de tamaño $2 \times \#P$, a la cuál se le aplica luego jerarquización de Pareto y se eligen $\#P$ individuos. En el caso de GDE3, el tamaño de la población temporal es de $\#P + m$ con $m \in [0, \#P]$ debido a que el individuo de prueba es incluido solo cuando éste y el individuo padre son soluciones factibles y no se dominan entre sí. El sorting no dominado se modifica para tener en consideración restricciones y la selección provee una mejor distribución del conjunto de individuos.

La selección en GDE3 se basa en las siguientes reglas:

- En el caso de vectores no factibles, se selecciona el vector de prueba si domina débilmente al vector padre en el espacio de restricciones, de lo contrario se selecciona el vector padre.
- En el caso de vectores factibles y no factibles, se selecciona el vector factible.
- Si ambos vectores son factibles, entonces se selecciona el vector de prueba si débilmente domina al vector padre en la espacio de la función objetivo. Si el vector padre domina al vector de prueba, entonces se selecciona el vector padre. Si no se dominan entre sí en el espacio de la función objetivo, entonces ambos vectores son seleccionados para la próxima generación.

Se pueden obtener variaciones de GDE3 simplemente utilizando un proceso de reproducción (mutación y cruzamiento) distinto al visto en la sección 2.3.4.

2.4.5. MOEA/D

La descomposición de un problema de optimización multiobjetivo en un conjunto de problemas de optimización monoobjetivo es una estrategia básica en la técnicas tradicionales de optimización multiobjetivo. La mayoría de los AEMO actuales, sin embargo, se basan en la dominancia de Pareto entre soluciones. MOEA/D propuesto por Zhang and Li (2007), es un AEMO reciente cuyo mecanismo principal es la descomposición explícita del problema en un conjunto de subproblemas de optimización escalar, los cuales se optimizan simultáneamente mediante la evolución de una población de soluciones. El objetivo en cada uno de estos problemas es una agregación de todos los objetivos, definiéndose relaciones de vecindario entre los subproblemas. La implementación propuesta por Zhang and Li (2007) utiliza el operador de cruzamiento *simulated binary crossover* (SBX). A continuación se analizará una implementación propuesta por Li and Zhang (2009) basada en la técnica de evolución diferencial (MOEA/D-DE). Esta implementación utiliza el enfoque de descomposición “de Tchebycheff” requerido para descomponer el problema original en una serie de problemas de optimización escalares. En el Algoritmo 5 puede verse el pseudocódigo del MOEA/D-DE.

Con el enfoque de Tchebycheff, los problemas escalares de optimización son de la forma indicada en la Ecuación 2.6.

$$\begin{aligned} \text{Minimizar } g(x|\lambda, z^*) &= \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\} \\ \text{sujeto a } x &\in \Omega \end{aligned} \quad (2.6)$$

donde $\lambda = (\lambda_1, \dots, \lambda_m)$ es un vector de pesos ($\lambda_i \geq 0$ para cada $i = 1, \dots, m$, $\sum_{i=1}^m \lambda_i = 1$) y $z^* = (z_1^*, \dots, z_m^*)$ es un punto de referencia. Ahora bien, si se considera un conjunto de vectores peso $\lambda^1, \dots, \lambda^N$ se obtienen N problemas de optimización multiobjetivo donde $\lambda = \lambda^i$ para el problema i -ésimo. Si N es suficientemente grande y los vectores peso son elegidos apropiadamente, las soluciones óptimas a estos subproblemas darán una buena aproximación del frente de Pareto. En MOEA/D, el vecindario del subproblema i -ésimo consiste de T subproblemas cuyos vectores peso están cerca de λ^i .

En cada generación t , se mantiene:

- una población de N puntos $x^1, \dots, x^N \in \Omega$, donde x^i es la solución actual al subproblema i -ésimo.
- $FV^1, \dots, FV^N$, donde $FV^i = (f_1(x^i), \dots, f_m(x^i)) \forall i = 1, \dots, m$.
- $z = (z_1, \dots, z_m)^T$, donde z_i es el mejor valor encontrado hasta el momento para el objetivo f_i .

Algoritmo 5: Pseudocódigo del algoritmo MOEA/D-DE

```

 $B(i) \leftarrow \text{Calcular\_distancias\_euclidianas\_y\_vecindario}(\lambda^i, \dots, \lambda^N)$ 
Inicializar_población( $x^1, \dots, x^N$ )
 $FV^i \leftarrow F(x^i)$ 
Inicializar_referencia( $z$ )
mientras no CriterioParada hacer
    para  $i \leftarrow 1, \dots, N$  hacer
        si  $\text{rand}([0, 1]) < \delta$  entonces
             $P \leftarrow B(i)$  en otro caso  $P \leftarrow \{1, \dots, N\}$ 
         $y \leftarrow \text{Reproducción}(i)$ 
         $\text{Reparar}(y)$ 
         $\text{Actualizar\_referencia}(z)$ 
         $c \leftarrow 0$ 
        mientras  $c < n_r \wedge P \neq \emptyset$  hacer
             $j \leftarrow \text{rand}(P)$ 
            si  $g(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$  entonces
                 $x^j \leftarrow y, FV^j \leftarrow F(y)$ 
                 $c \leftarrow c + 1, P \leftarrow P - \{j\}$ 

```

Inicialmente, se calculan las distancias euclidianas entre todo par de vectores peso y se elijen los T más cercanos a cada uno: para cada $i = 1, \dots, N$, se establece $B(i) = \{i_1, \dots, i_T\}$ donde $\lambda^{i_1}, \dots, \lambda^{i_T}$ son los T vectores peso más cercanos a λ^i . Luego se genera una población inicial $x^1, \dots, x^N$ en forma aleatoria. El punto de referencia se inicializa asignando $z_j = \min_{1 \leq i \leq N} f_j(x^i)$ para $j = 1, \dots, m$. Luego, mientras no se cumpla el criterio de parada, se ejecutan los siguientes pasos para $i = 1, \dots, N$:

- Rango de actualización: se decide si los padres serán elegidos del vecindario con probabilidad δ .
- Reproducción: se elijen dos índices, r_1 y r_2 de P , y se genera un vector $\bar{y}$ aplicando un operador de evolución diferencial a x^i , x^{r_1} y x^{r_2} . La nueva solución y se obtiene aplicando mutación sobre $\bar{y}$ con probabilidad p_m .
- Reparación: se ajustan aleatoriamente los elementos de y que están fuera de los límites de Ω .
- Actualización de z : para $j = 1, \dots, m$, si $z_j > f_j(y)$, asignar $z_j = f_j(y)$.
- Actualización de soluciones: se sustituyen hasta un máximo de n_r soluciones con índices en P por el vector y mientras este tenga mejor fitness.

MOEA/D-DE ha sido comparado con el algoritmo NSGA-II con los mismos operadores de reproducción mostrando un desempeño superior en varias instancias de prueba con conjuntos de Pareto complicados (Li and Zhang, 2009).

2.4.6. ϵ -NSGA-II

ϵ -NSGA-II (Kollat and Reed, 2006) es una versión modificada de NSGA-II, que incorpora *ϵ -dominance archiving* (Laumanns et al., 2002), adaptación dinámica del tamaño de la población y terminación automática para evitar la calibración de parámetros exhaustiva. El concepto de ϵ -dominance permite al usuario especificar la precisión con qué se quiere cuantificar cada objetivo en un problema de optimización multiobjetivo. Es importante señalar que la adición de *ϵ -dominance archiving* no agrega parámetros adicionales al algoritmo. En cambio, permite al usuario definir los requisitos de precisión que tienen sentido para su aplicación particular.

ϵ -NSGA-II utiliza una serie de “ejecuciones conectadas” donde pequeñas poblaciones son inicialmente explotadas para precondicionar la búsqueda y automáticamente adaptar el tamaño de la población acorde con la dificultad del problema. A medida que la búsqueda progresa, el tamaño de la población se adapta automáticamente basado en el número de soluciones ϵ -no-dominadas encontradas por el algoritmo.

Las soluciones ϵ -no-dominadas encontradas después de cada generación se almacenan en un archivo y se utiliza posteriormente para dirigir la búsqueda usando un esquema de inyección de 25 %. En el esquema de inyección, 25 % de la población subsiguiente estará compuesta de las soluciones del archivo y el otro 75 % será generado aleatoriamente. Esto ayuda a la búsqueda de dos maneras: (i) dirigiendo la búsqueda usando soluciones previamente evolucionadas y (ii) mediante la adición de nuevas soluciones para fomentar la exploración de regiones adicionales del espacio de búsqueda. Este esquema de inyección limita el tamaño de la población a cuatro veces el número de soluciones que existen en la ϵ -resolución especificada por el usuario.

La búsqueda se termina en todas las ejecuciones (es decir, a través de todas las poblaciones utilizadas) si el número y la calidad de soluciones no ha aumentado por encima de un valor determinado luego de dos ejecuciones seguidas.

2.5. AE paralelos

Con la aplicación cada vez más extendida de AE, existe una continua necesidad de resolver problemas complejos del mundo real, con un alto grado de efectividad y eficiencia (Coello Coello et al., 2007). La rápida evolución de la tecnología en componentes que sirven para incrementar la eficiencia computacional, como ser, procesadores (CPUs multi-núcleo (David and Wen-mei, 2010)), redes (Infiniband), arquitecturas (GRIDs, clusters), memoria (compartida y distribuida (Chapman et al., 2008)), capacidad de almacenamiento, etc., ha popularizado el uso de la paralelización en el diseño e implementación de algoritmos de optimización multiobjetivo (Talbi et al., 2008; Talbi and Meunier, 2006; Streichert et al., 2005; Branke et al., 2004). La integración de la computación paralela y la computación evolutiva ha dado nacimiento a los *algoritmos evolutivos paralelos*. Un AE paralelo busca

encontrar soluciones tan buenas o mejores que su contraparte serial AE en menor tiempo, a la vez que explora un territorio más amplio del espacio de soluciones (Van Veldhuizen et al., 2003).

Un AE paralelo puede ser útil cuando aparecen problemas en los cuales es costoso, desde un punto de vista computacional, calcular las funciones objetivo, como en Martikainen and Ovaska (2008). En este caso, se podrían utilizar técnicas para descomponer paralelamente esta función. Otra posibilidad es distribuir la población entre varios procesadores. En la siguiente sección se presentan las estrategias de paralelización más conocidas en la literatura.

2.5.1. Modelos de paralelización

En general, se distinguen tres modelos principales de paralelización:

- *Maestro-esclavo* o de *paralelización global*: este modelo es de los más simples de implementar. Se tiene una única población de individuos, gestionada por un proceso maestro que controla todo el proceso evolutivo y distribuye la evaluación de las funciones objetivo entre varios subprocesos esclavos (ver Figura 2.4). Aquellas operaciones que requieren operar sobre la población completa (asignación de fitness, selección, etc) son ejecutadas por el maestro, mientras que el resto (cruzamiento, mutación) son candidatas a ejecutarse por los esclavos al igual que la evaluación de las funciones objetivo. La exploración del espacio de búsqueda es conceptualmente idéntica a la conseguida en la versión serial, por lo que se obtienen las mismas soluciones. Siempre que el tiempo de evaluación de las funciones objetivo sea suficientemente alto en relación al tiempo de comunicación entre el maestro y los esclavos, se reduce el tiempo total de ejecución. Algunos ejemplos se pueden encontrar en Xiong and Li (2003); Xu et al. (2005); Talbi and Meunier (2006).

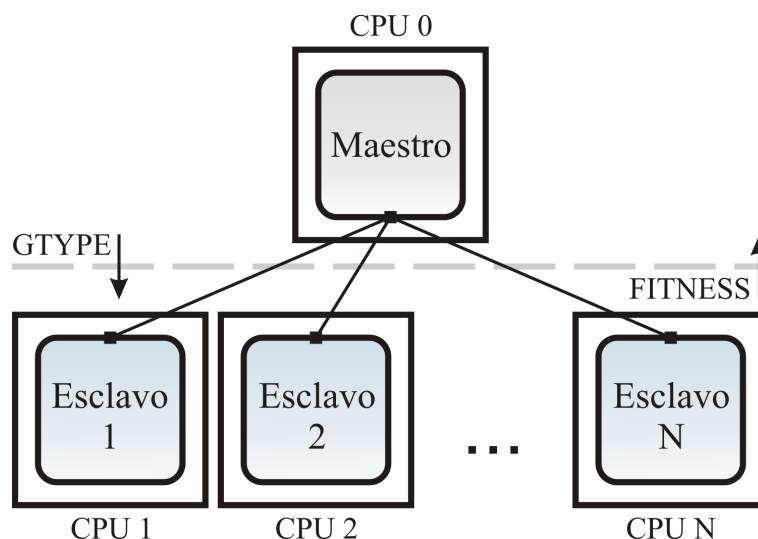


Figura 2.4: Modelo de paralelización *maestro-esclavo*.

- *Modelo de islas*: este modelo está inspirado en el fenómeno natural donde poblaciones separadas geográficamente evolucionan en forma aislada. La población total del algoritmo es dividida en varias subpoblaciones o *demes*, las cuales sufren procesos de evolución independientes. Cada proceso evolutivo podría utilizar distintos parámetros así como un AEMO diferente. Se tiene además un mecanismo de comunicación entre las islas denominado *migración*, donde individuos migran de una isla a otra siguiendo una política establecida. Continuando con la idea de copiar el funcionamiento de la evolución de las especies, resulta natural imitar la evolución en grupos separados, dando lugar a especies o nichos y no se debería permitir que cualquier individuo interactúe con cualquier otro de la población total (*panmixia*). Los algoritmos así paralelizados demuestran comportarse de forma distinta a la secuencial y, en general, mejoran la búsqueda (Alba, 1999). En la Figura 2.5 (a) se puede ver una representación de este modelo. Algunos ejemplos se pueden encontrar en Banos et al. (2006); Essabri et al. (2006); Talbi and Meunier (2006).
- *Modelo celular*: diseñado para ser implementado usando computadoras masivamente paralelas, el modelo celular cuenta con una población dividida espacialmente entre los distintos procesadores. El cruce y la selección de individuos se realiza entre individuos que pertenecen a un mismo *vecindario*, formado por un conjunto de individuos adyacentes. Los vecindarios se pueden solapar (como se observa en la Figura 2.5 (b)), o pueden ser dinámicos, de manera que las buenas soluciones se difundan lentamente a través de la población (Van Veldhuizen et al., 2003).

Otra técnica consiste en aplicar la idea de hibridación incorporando los modelos anteriores a un sistema jerárquico. Por ejemplo, el modelo de islas puede emplear el modelo celular en cada una de las islas (ver Figura 2.5 (c)). Para más información ver Van Veldhuizen et al. (2003).

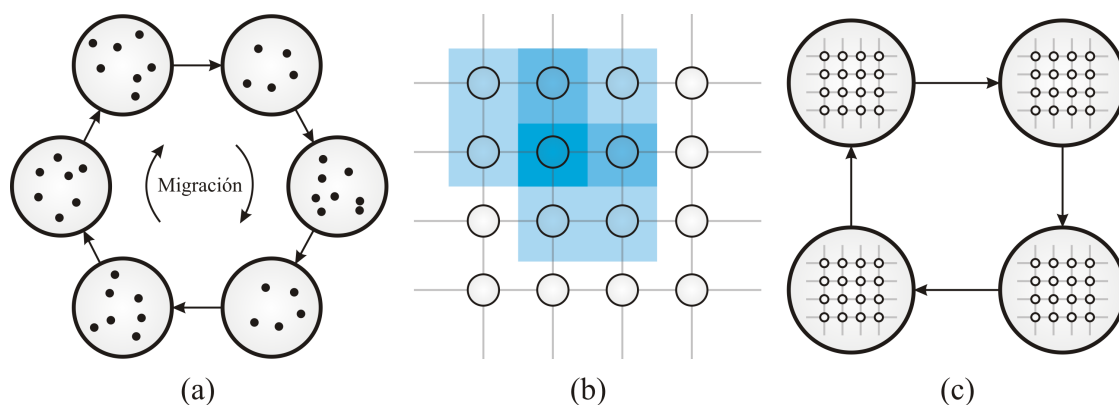


Figura 2.5: Vista esquemática del modelo de islas con topología de anillo (a), modelo celular (b) y de híbrido entre el modelo de islas (en el nivel superior) y el modelo celular (en el nivel inferior) (c).

2.6. Evaluación experimental de AEMO

En la mayoría de los problemas de optimización que involucran múltiples objetivos o criterios, no hay una solución única que sea óptima simultáneamente con respecto a todos los objetivos, sino que existen varias que son incomparables. El número de soluciones posibles, que constituye el llamado espacio de búsqueda, es con frecuencia muy grande de tal manera que los algoritmos computacionales son el método de elección. La pregunta fundamental que responden los estudios comparativos es si un algoritmo dado A , es mejor que un algoritmo dado B . Generalmente, al seleccionar un problema, se intenta resolverlo utilizando varios algoritmos y se comparan sus resultados para determinar cuál produce el mejor resultado. Sin embargo, existen varios factores en juego que afectan la conclusión de tales estudios.

Medir la calidad de un resultado para un problema de optimización multiobjetivo es significativamente más difícil que para un problema de optimización monoobjetivo. En el caso monoobjetivo, se puede medir la diferencia entre las soluciones obtenidas sin ambigüedad y comparar con el óptimo global (si es conocido). Esta medida se puede usar como métrica de rendimiento. En la optimización multiobjetivo, no existe una única medida de rendimiento. Adicionalmente, la definición de buen rendimiento puede variar de persona a persona. En Zitzler et al. (2000), se consideran los siguientes tres objetivos principales en la optimización multiobjetivo:

- *Proximidad*: Minimizar la distancia entre el conjunto de soluciones obtenidas por el algoritmo y el frente óptimo de Pareto.
- *Distribución*: Se desea una buena distribución de las soluciones encontradas (en la mayoría de los casos uniforme). La evaluación de este criterio debe basarse en una métrica de distancia determinada.
- *Consistencia*: Maximizar la extensión de las soluciones del conjunto no-dominado obtenido. Esto es, para cada objetivo, se deben descubrir una amplia gama de valores.

Un algoritmo ideal es aquel que produce resultados que cumplen los tres objetivos ya que estos están cerca del resultado óptimo (proximidad) y capturan la *diversidad* entre los objetivos (distribución, consistencia). Esto requiere de una búsqueda hacia el conjunto óptimo de Pareto y a lo largo del conjunto como se muestra en la Figura 2.6. Para capturar analíticamente estas metas, se utiliza una serie de métricas de rendimiento frecuentemente usadas en la literatura. Estas incluyen *hipervolumen*, *distancia generacional*, *distancia generacional invertida*, *indicador épsilon aditivo*, *máximo error al frente de Pareto* y *espaciamiento*. Las mismas se describen en la próxima sección.

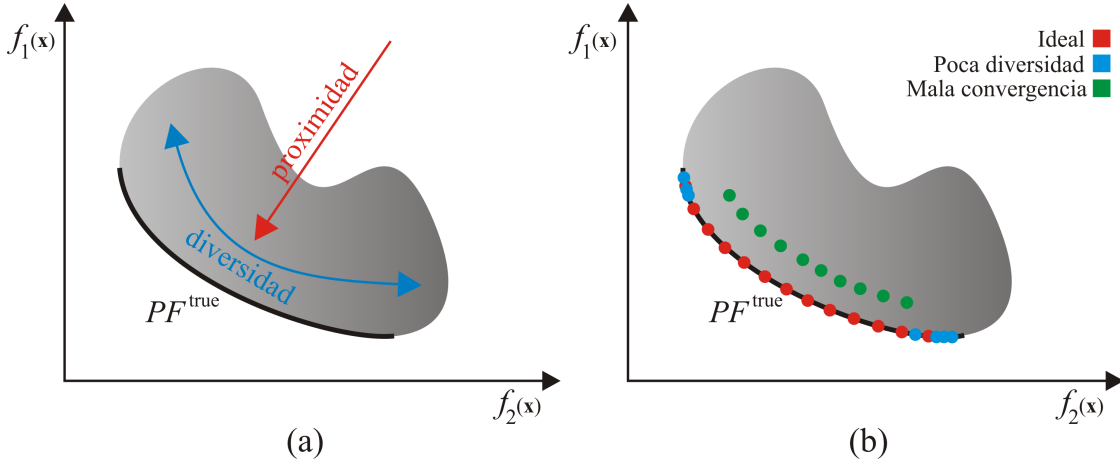


Figura 2.6: En la gráfica (a) se muestra la búsqueda para capturar proximidad y diversidad de las soluciones, en la (b) se ilustran tres conjuntos de soluciones no dominadas: ideal, con poca diversidad y con mala convergencia.

2.6.1. Métricas de evaluación

Sea Q el conjunto de soluciones no dominadas resultante de la ejecución de un algoritmo de optimización y QF su imagen en el espacio de objetivos, se consideran las siguientes métricas:

- *Hipervolumen* (HV): este indicador de calidad calcula el volumen cubierto por el hipercubo formado por los vectores de QF en problemas donde se minimicen todos los objetivos (Zitzler and Thiele, 1999). Matemáticamente, para cada elemento $y \in QF$ se construye un hipercubo hc_y con un punto de referencia p_{ref} que forma la diagonal del hipercubo. El punto de referencia se puede encontrar construyendo un vector con los peores valores de las funciones objetivo, de forma tal que sea al menos débilmente dominado por todos los miembros de QF . Luego se calcula el HV mediante una unión de todos los hipercubos como se expresa en la Ecuación 2.7.

$$HV = volumen \left(\bigcup_{y \in QF} hc_y \right) \quad (2.7)$$

La Figura 2.7 muestra el HV para un ejemplo de dos objetivos. Los valores altos de HV son convenientes.

- *Distancia generacional* (DG): indica cuán lejos se encuentra QF del verdadero frente PF^{true} (Van Veldhuizen, 1999). Formalmente se define como se indica en la Ecuación 2.8,

$$DG = \frac{\left(\sum_{x \in Q} d(x)^p \right)^{1/p}}{|Q|} \quad (2.8)$$

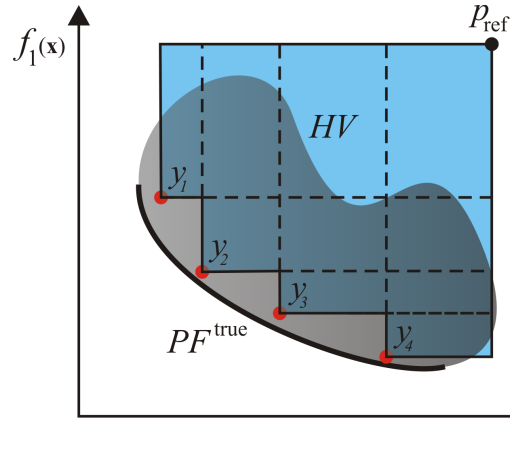


Figura 2.7: Hipervolumen para un caso de dos objetivos.

Para $p = 2$, $d(x)$ es la distancia euclídea, en el espacio objetivo, entre el miembro x de Q y el miembro de P^{true} más cercano. Ver Ecuación 2.9.

$$d(x) = \min_{t \in P^{\text{true}}} \sqrt{\sum_{i=1}^M (f_i(x) - f_i(t))^2} \quad (2.9)$$

Un valor de DG igual a cero indica que todos los elementos generados se encuentran en el frente de Pareto. Para un conjunto QF con gran fluctuación de distancias puede que no se logre revelar la verdadera distancia por lo que se recomienda que QF tenga un gran número de soluciones.

- *Distancia generacional invertida* (DGI): en forma inversa a la distancia generacional, este indicador de calidad se utiliza para medir qué tan lejos se encuentran los elementos que están en el frente de Pareto del conjunto de vectores no-dominados encontrados. La DGI reduce el sesgo que puede surgir cuando QF no cubre completamente el frente de Pareto. Si se utiliza un conjunto de puntos distribuidos uniformemente sobre todo el frente de Pareto, un valor pequeño de DGI indica buena convergencia y buena distribución de las soluciones (Ishibuchi et al., 2015).
- *Indicador- ϵ aditivo* ($I_{\epsilon+}$): como se observa en la Definición 5, para dos conjuntos A y B , el indicador- ϵ aditivo puede pensarse como una medida de la mínima distancia que debemos desplazar B para que sea dominado por A (Zitzler et al., 2003).

Definición 5. Indicador- ϵ aditivo binario. *Dados dos conjuntos $A, B \subseteq \Omega$ se define el $I_{\epsilon+}$ como:*

$$I_{\epsilon+}(A, B) = \min_{\epsilon \in \mathbb{R}} \{ \forall z^2 \in B \exists z^1 \in A : z^1 \succeq_{\epsilon+} z^2 \} \quad (2.10)$$

donde $z^1 \preceq_{\epsilon+} z^2$, si y solo si, $\forall i = 1, 2, \dots, N : z_i^1 \leq \epsilon + z_i^2$ para un $\epsilon > 0$ dado.

- *Máximo error al frente de Pareto* (MEF): representa la máxima distancia de las soluciones en QF a la solución más cercana del frente de Pareto, de acuerdo a la expresión de la Ecuación 2.11 (Knowles and Corne, 2002).

$$\text{MEF} = \max_{x \in Q} \left(\min_{t \in P^{\text{true}}} \sum_{i=1}^M |f_i(x) - f_i(t)|^2 \right)^{1/2} \quad (2.11)$$

Se desea un valor de MEF pequeño, aunque se debe tener en cuenta que es una medida de convergencia conservativa y puede reflejar información incorrecta sobre la distribución de las soluciones.

- *Espaciamiento* (ESP): el indicador de espaciamiento evalúa la distribución de los elementos del conjunto QF . En Schott (1995) se propone la definición indicada en la Ecuación 2.12.

$$\text{ESP} = \sqrt{\frac{1}{|Q|-1} \sum_{x \in Q} (d(x) - \bar{d})^2} \quad (2.12)$$

donde, $d(x) = \min_{k \in Q \wedge k \neq x} \sum_{i=1}^M |f_i(x) - f_i(k)|$ y $\bar{d}$ es el promedio de todos los $d(x)$, $\bar{d} = \sum_{x \in Q} \frac{d(x)}{|Q|}$. Notar que la medida de distancia es distinta de la distancia euclidiana mínima entre dos soluciones. Esta métrica toma un valor de cero para una distribución ideal, lo que indica que todas las soluciones encontradas están igualmente espaciadas, por lo que un valor pequeño es deseable.

Muchos de los indicadores de rendimiento requieren de P_{true} . Para muchos de los problemas del mundo real, sin embargo, no es posible determinar el conjunto de óptimos de Pareto y es una práctica común el uso de la mejor aproximación como conjunto de referencia. A este conjunto generalmente se le denota P^* y consiste de todas las soluciones no dominadas producidas por todos los algoritmos de optimización para una instancia del problema en particular.

2.6.2. Análisis estadístico

Dado un determinado problema de optimización, el objetivo consiste en encontrar soluciones de alta calidad en un tiempo razonable. Ahora bien, debido a que los AEMO son estrategias aproximadas no deterministas, ejecutar un algoritmo varias veces sobre un mismo problema no garantiza encontrar la misma solución y su rendimiento (efectividad y eficiencia) es difícil de predecir. Dado que se desconoce la solución óptima para el problema de optimización, es necesario realizar comparaciones empíricas con las que se pueda evaluar la calidad de las soluciones

aportadas por los algoritmos. En este sentido, es importante abordar el número de ejecuciones independientes necesarias para adquirir datos estadísticos suficientes. Usualmente, se necesitan 30 o más, de acuerdo al *teorema del límite central* (Coello Coello et al., 2007), aunque puede variar de 10 a más de 30 dependiendo del dominio del problema. De este modo, un algoritmo no produce un único resultado sino una distribución de resultados (para cada métrica). Al comparar algoritmos es necesario entonces utilizar pruebas estadísticas que permitan determinar si sus distribuciones son similares o diferentes.

Las pruebas estadísticas pueden ser *paramétricas* o *no paramétricas*. Los métodos paramétricos se basan en el muestreo de una población con parámetros específicos (como la media poblacional, desviación estándar, etc) y exigen que se cumplan determinados supuestos sobre las poblaciones originales de las que se extraen los datos (generalmente que la población se distribuya normalmente). Además se emplean con datos en una escala ordinal, de intervalo o de razón. En el caso de que uno de estos supuestos no se cumpla, es probable que las pruebas paramétricas generen resultados erróneos, y por ende, las conclusiones de sus hipótesis serán inválidas. Los métodos estadísticos no paramétricos en cambio, ofrecen menor rigidez y no requieren que se cumplan ciertas condiciones como la distribución normal de los datos. Los métodos no paramétricos se aplican a una gran variedad de situaciones aunque la potencia de las pruebas es menor a las pruebas paramétricas. Cuando se trata de una distribución normal, otros métodos pueden ser más precisos, aunque no existe error si la prueba no paramétrica concluye que no existe diferencia entre los resultados.

2.6.3. Análisis no paramétrico (análisis de varianza)

Existen muchos métodos estadísticos no paramétricos que pueden ser utilizados para saber si algunos de los indicadores de calidad estudiados son diferentes (mejor o peor) para dos o más algoritmos. Uno de los métodos más utilizados es la prueba de *Mann-Whitney* (también llamada *suma de rangos Wilcoxon*) para dos muestras independientes. Se trata de la versión no paramétrica de la conocida prueba *t de Student*. La prueba de *Kruskal-Wallis* es la forma general de la prueba de Mann-Whitney para tres o más muestras. Esta prueba no asume normalidad en los datos, en oposición al tradicional método de análisis de varianza *ANOVA* (Higgins, 2003).

Dentro de la inferencia estadística, un contraste de hipótesis es un procedimiento para juzgar si una propiedad que se supone en una población estadística es compatible con lo observado en una muestra de dicha población. Mediante esta teoría se aborda el problema estadístico considerando una hipótesis nula H_0 y una hipótesis alternativa H_1 , y se intenta dirimir cuál de las dos es la hipótesis verdadera. Se denomina hipótesis nula, a la hipótesis que se desea contrastar. El p -valor (p_v) es una medida directa de lo verosímil que resulta obtener una muestra como la actual si es cierta H_0 y permite decidir qué hipótesis se acepta de acuerdo a un nivel de riesgo α determinado (por ejemplo, 0,05 ó 0,01). Si $p_v < \alpha$, se acepta H_1 , si $p_v \geq \alpha$ se acepta H_0 .

En la prueba de Kruskal-Wallis (utilizada para el análisis de los algoritmos desarrollados en este proyecto), la hipótesis nula afirma que las muestras son aleatorias, extraídas de poblaciones con la misma distribución. La hipótesis alternativa afirma que al menos una de las poblaciones tiene mediana distinta a las otras. Consideremos h muestras aleatorias e independientes de tamaños $n_1, n_2, \dots, n_h$ extraídas de la misma población o de h poblaciones idénticas. Sea $N = \sum_{i=1}^h n_i$ la suma total de observaciones. Se asignan rangos de 1 hasta N al conjunto de N observaciones como si se tratara de una sola muestra. Si existe empate se asigna el promedio de los rangos empatados. Sea r_{ij} el rango asignado a la observación j de la muestra i , y r_i la suma de los rangos asignados a las n_i observaciones de la muestra i , entonces:

$$r_i = \sum_{j=1}^{n_i} r_{ij} \text{ y } \bar{r}_i = \frac{r_i}{n_i} \quad (2.13)$$

Tenemos que el estadístico de la prueba Kruskal-Wallis esta dado por:

$$K = \frac{12}{N(N+1)} \sum_{i=1}^h \frac{r_i^2}{n_i} - 3(N+1) \quad (2.14)$$

Finalmente, el p-valor es aproximado por $Pr(\chi_{h-1}^2 \geq K)$. Si algún n_i es pequeño (< 5) la distribución de K puede ser distinta de la *chi-cuadrado*.

Capítulo 3

Planificación de ciclos de semáforos y las emisiones vehiculares

Este capítulo tiene como objetivo introducir el problema de optimización de ciclos de semáforos considerando emisiones vehiculares. Se presenta una metodología para el estudio del flujo vehicular mediante conteos de tráfico.

3.1. Introducción

Actualmente, el volumen de tráfico vehicular existente en las grandes ciudades provoca severos problemas relacionados con la contaminación atmosférica (CO , NO_x , PM_x , etc.) y sonora, la seguridad, así como la disponibilidad de lugares para estacionamiento. Debido a que las mejoras en infraestructura física suelen traer aparejado un alto costo de inversión, y no siempre son abordables, los investigadores en este área están de acuerdo en que una planificación correcta y eficiente de los semáforos puede ayudar a reducir estos problemas mediante la mejora de la circulación de vehículos en las ciudades (McCrea and Moutari, 2010; Sánchez et al., 2008; Olivera et al., 2015). Por otra parte, el número de semáforos operativos está creciendo y su planificación conjunta es cada vez más compleja debido a la gran cantidad de combinaciones de periodos de colores que se deben manejar. Por tanto, la utilización de sistemas automáticos inteligentes para la planificación óptima de ciclos de semáforos es una tarea indispensable en la gestión del tráfico (García-Nieto et al., 2013).

La siguiente sección realiza una aproximación general al problema definiendo aquellos elementos de la realidad que deben tenerse en cuenta al momento de la formalización. Entre estos elementos destacan el flujo vehicular, el cuál se puede describir mediante el concepto de *matriz origen-destino* (matriz OD). En la Sección 3.3, se analiza un método manual de estudio del flujo vehicular para un área determinada. Luego, en la Sección 3.4 se presenta una breve introducción al problema de contaminación atmosférica profundizando en las principales emisiones vehicula-

sección están gobernados por un programa común. Asimismo, la combinación de colores de los semáforos, durante un periodo de ciclo determinado, debe ser siempre válida, siguiendo las reglas de tráfico específicas de dichas intersecciones con el fin de evitar colisiones y accidentes automovilísticos.

En la Figura 3.1 se presenta una intersección semaforizada de una red de tráfico, donde se destacan aquellos elementos que son tenidos en cuenta al momento de considerar el problema de optimización de ciclos (semáforos, restricciones de giro, carriles, etc). Ahora bien, la identificación de estos elementos es un paso fundamental en el proceso de modelado de la realidad, dado que los elementos no considerados (como por ej. las personas), no serán tenidos en cuenta al momento de obtener una solución al problema. Una vez determinadas las características básicas de la intersección, se procede a identificar el programa de ciclos que controla los semáforos de la intersección. Para ello, se identifican los distintos estados del programa (asociados a su respectiva duración), como se puede ver en la Figura 3.2. Un estado particular de un programa de ciclos se puede representar como una cadena de caracteres, donde se tiene un carácter por cada semáforo del programa y su valor indica el color del semáforo. Esto es, si el valor del primer carácter de la cadena es “V” entonces el semáforo 1 se encuentra en verde, si el valor es “A”, se encuentra en amarillo y si es “R” se encuentra en rojo. Cada programa de ciclos cuenta con una secuencia finita de estados que se repite indefinidamente. Más adelante se hará referencia a la configuración real del programa, que refiere a la configuración cuyos estados tienen una duración igual a la utilizada en la realidad.

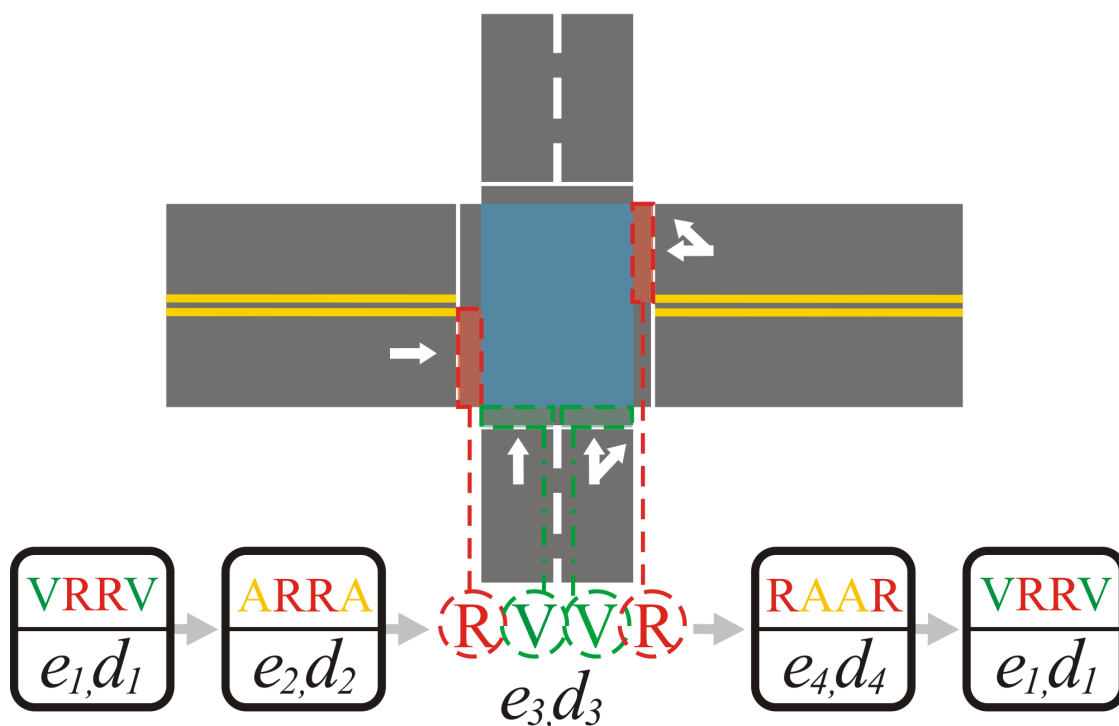


Figura 3.2: Diagrama de una intersección controlada por un programa de ciclos. Se describe la transición entre los estados (e_i) y sus respectivas duraciones (d_i).

Otro de los elementos a considerar es el tránsito vehicular o automovilístico (también llamado tráfico vehicular, o simplemente tráfico), fenómeno causado por el *flujo de vehículos* en la red. Este concepto se define a continuación.

3.2.2. Flujo vehicular

El *flujo vehicular* define las características y el comportamiento del tráfico, y su comprensión es uno de los requisitos básicos en el planteamiento, proyecto y desarrollo de una red óptima de calles con movimiento eficiente y problemas mínimos de congestión de tráfico dentro del sistema de transporte. Por dicho motivo, durante los últimos cincuenta años se han desarrollado una amplia gama de teorías y modelos de flujo de tráfico que intentan modelar el flujo vehicular en términos de algunas de sus características.

Los modelos se pueden clasificar en base al nivel de detalle con el que se describe el sistema vehicular. Los más destacados son los llamados modelos *macroscópicos* y *microscópicos*. Los modelos macroscópicos (Shvetsov and Helbing, 1999; Velasco and Saavedra, 2008) se enfocan en captar las relaciones globales del flujo y no toman en cuenta las características específicas de los vehículos. Generalmente, debido a su comportamiento estadístico, se utilizan promedios para variables tales como tamaño de vehículos y distancia de separación entre vehículos. Por otro lado, los modelos microscópicos (Knospe et al., 2000), se enfocan en la descripción del comportamiento del flujo vehicular detallando las características propias de cada vehículo y especificando las reglas de movimiento individual. A pesar de que con estos modelos se obtiene una visión más completa sobre la dinámica del tráfico, su generación requiere una gran cantidad de datos los cuales no siempre son accesibles fácilmente.

En contraste con las variables de flujo de tráfico microscópicas, las cuales se centran en cada vehículo en forma individual, las variables macroscópicas reflejan el estado promedio de la fluidez del tráfico en su conjunto. Las más importantes son la *tasa de flujo* o simplemente *flujo*, *velocidad*, y *densidad* (o *concentración*).

La tasa de flujo q se define tradicionalmente como “el número promedio de vehículos” n que atraviesa un punto o una sección transversal durante una unidad de tiempo (T). De acuerdo a esta definición, la tasa de flujo es una variable local (dado que se define en una sección transversal) definida por la Ecuación 3.1.

$$q = \frac{n}{T}. \quad (3.1)$$

De forma similar, la densidad k se define como “el número de vehículos por unidad de distancia” y es una variable instantánea dado que se computa en un instante de tiempo. Esta variable da una idea de que tan ocupado está un segmento de calle, ignorando la heterogeneidad del tráfico y el largo de los vehículos. Dado un segmento de calle de largo X y un número m de vehículos que ocupan el segmento en un instante dado, se define k como se indica en la Ecuación 3.2.

$$k = \frac{m}{X}. \quad (3.2)$$

La velocidad se define como “la distancia cubierta por unidad de tiempo”. En la práctica, la velocidad instantánea de cada vehículo es casi imposible de seguir por lo que se realiza un promedio basado en muestras de tiempo o espacio. Puede computarse de dos formas: en una sección trasversal (velocidad media local o velocidad media temporal u_L), o en un instante de tiempo (velocidad media instantánea o velocidad media espacial u_M).

Si se considera que todos los vehículos sobre el segmento se mueven a la misma velocidad u , podemos definir el período $T = \frac{X}{u}$. Ahora bien, el número de vehículos que se encuentra en tiempo $t = 0$ sobre el segmento (igual a la densidad k multiplicado por el largo del segmento X) se corresponde con el número de vehículos que atraviesa el punto final del segmento $x = X$ durante el período $[0, T]$ (igual al flujo q multiplicado por la duración del periodo). Se tiene entonces:

$$kX = qT \Leftrightarrow q = k \frac{X}{T} = ku. \quad (3.3)$$

La relación $q = ku$ se conoce con el nombre de *ecuación de continuidad* y se mantiene cuando la velocidad es constante, o bien, si se utiliza la velocidad media espacial $u = u_M$.

Una relación estadística fundamental entre las variables definidas es el *diagrama fundamental de flujo vehicular*. Esta relación se representa comúnmente mediante la relación flujo-densidad $q = Q(k)$, aunque también se puede expresar como $u = U(k)$ y $u = U(q)$ de acuerdo a la ecuación de continuidad. La premisa subyacente al diagrama fundamental es que bajo condiciones de tráfico similares, los conductores se comportarán de manera similar. En la Figura 3.3, se muestran ejemplos de la relación entre tasa de flujo, densidad y velocidad (Hoogendoorn and Knoop, 2013). La capacidad de la calle q_c , la densidad crítica k_c y la velocidad crítica u_c son los puntos más importantes. k_c y u_c ocurren cuando ocurre q_c . Es posible ver además la diferencia entre la *libre circulación de tráfico* o *free flow* ($k < k_c$) y el estado de congestión.

3.2.3. Matriz OD

Una matriz OD o *tabla de viajes OD* es la principal fuente de entrada de muchos estudios de transporte, tales como, pronóstico de patrones de demanda futuros, ingeniería de planificación urbana, migración humana y gestión y control de transporte. Las matrices OD representan la distribución espacial de viajes entre las zonas de análisis en una red de transporte y son el resultado de la suma de los movimientos individuales que se producen en un intervalo de tiempo.

Las matrices OD se pueden categorizar en estáticas o dinámicas según su aplicación. Una matriz estática no considera flujos de tráfico dependientes del tiempo y se asume que representa una situación estable durante un período de tiempo determinado, mientras que una matriz dinámica se aplica en estrategias de corto plazo como orientación, control de tráfico en autopistas, intersecciones, etc (Bera and Rao, 2011). El enfoque aplicado en este trabajo se basa en una estrategia de estimación con aproximación estática.

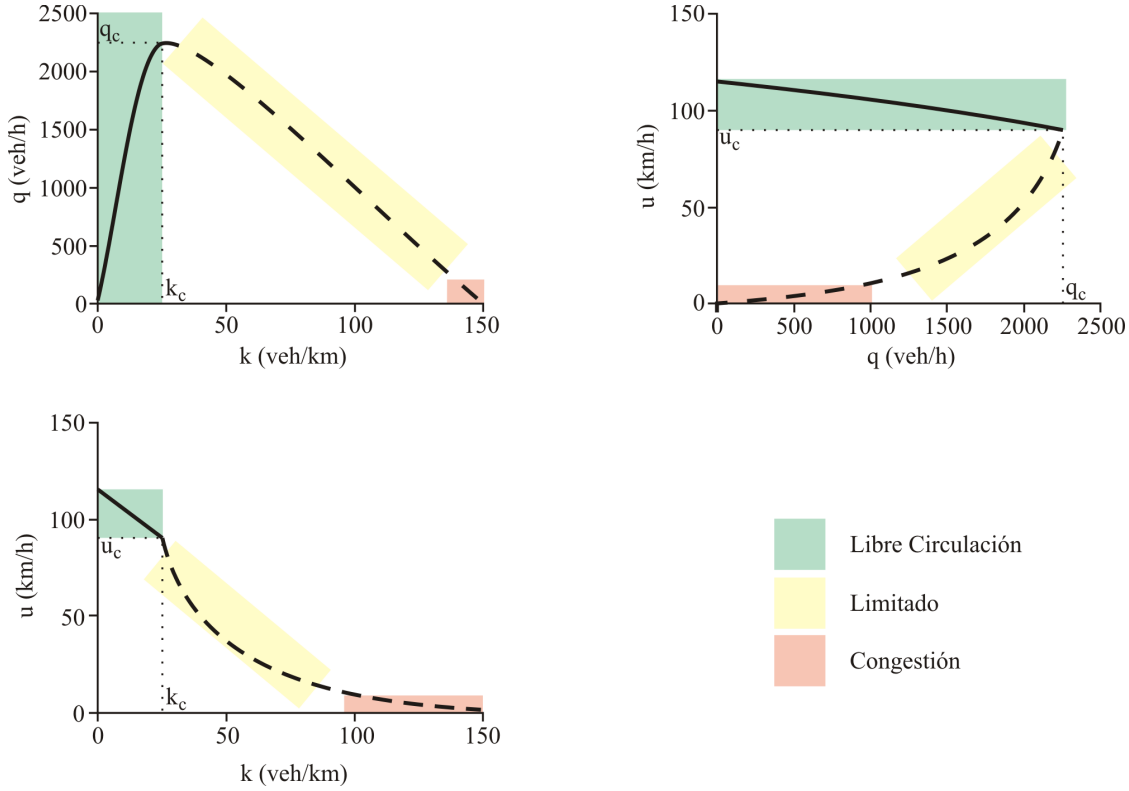


Figura 3.3: Diagramas fundamentales: ejemplos de la relación entre tasa de flujo, densidad y velocidad.

Más específicamente, dada una red dirigida $G(N, A)$ donde N es el conjunto de nodos y A es el conjunto de arcos dirigidos en la red, una matriz OD M describe el flujo de tráfico entre el conjunto de nodos de entrada $N_e \subset N$ y el conjunto de nodos de salida $N_s \subset N$. Se consideran los siguientes elementos:

- W es el conjunto de todos los pares OD $w = (i, j)$.
- $M \in \mathbb{N}^{|N_e| \times |N_s|}$; cada elemento $M[i, j]$ representa el volumen de tráfico del par OD $w \in W$ que fluye del nodo de entrada $n_i \in N_e$ al nodo de salida $n_j \in N_s$ (flujo unidireccional) durante un período de tiempo T_M .
- $volumen_M(l) = \sum_{w \in W} \sum_{r \in R_w} \delta_{r,l}^w p_r M[w]$, $\forall l \in A$, donde R_w es el conjunto de caminos entre el par OD w , p_r es la proporción de elección de la ruta r y $\delta_{r,l}^w = 1$ si el camino r contiene al arco l , 0 de lo contrario.

Como se observa en el último punto, es posible determinar la cantidad de vehículos que atraviesan el arco l durante el tiempo T_M , siempre y cuando se cuente con los conjuntos de caminos y su proporción de elección entre cada par OD. El $volumen_M(l)$ se corresponde con la variable macroscópica tasa de flujo q del arco l .

3.2.4. Simulación de tráfico

El término simulación hace referencia a una amplia colección de métodos y aplicaciones que imitan las operaciones o el funcionamiento de un proceso, normalmente mediante un ordenador y con el software adecuado. A este proceso se le denomina normalmente como *sistema* y para estudiarlo científicamente se debe hacer un conjunto de supuestos acerca de como se comporta. Estos supuestos, constituyen un *modelo* del sistema (Kelton and Law, 2000).

La simulación es un instrumento de análisis extraordinariamente versátil y flexible que proporciona la posibilidad de estudiar sistemas complejos que no pueden ser fácilmente modelados analíticamente. Habitualmente, tales sistemas son de naturaleza aleatoria, lo que implica generar (mediante técnicas de computación) fenómenos estocásticos y analizar estadísticamente las observaciones generadas por el modelo de simulación para estimar el comportamiento de las medidas de interés. Se utilizan estos ambientes simulados para poner a prueba los modelos científicos con el fin de probar o refutar su viabilidad y corrección (Kotushevski and Hawick, 2009).

El aumento constante de la potencia de las computadoras, la evolución de la ingeniería de software y el advenimiento de los sistemas inteligentes de transporte, han llevado a que el análisis de tráfico mediante simulaciones se convierta en uno de los enfoques más utilizados en apoyo al diseño y evaluación de los sistemas de tráfico. La posibilidad de emular la variabilidad temporal de los fenómenos del tráfico hacen de la simulación una herramienta única para capturar la complejidad de estos sistemas (Barceló, 2010).

En conformidad con lo visto en la Sección 3.2.2 sobre los modelos del flujo vehicular, se tiene que los simuladores de tráfico se pueden clasificar en *microscópicos*, *macroscópicos*, *mesoscópicos* y *nanoscópicos* (Barceló, 2010; Immers and Logghe, 2002):

- **Microscópicos:** se modelan las entidades (por ejemplo, vehículos, conductores, etc.) en forma individual con un alto nivel de detalle. Estos simuladores ofrecen resultados que dan una idea muy detallada del funcionamiento de la red de tráfico y son muy útiles para analizar estrategias de control, la estructura de la red y la sensibilidad del sistema ante cambios muy concretos en el tráfico. No obstante, las simulaciones microscópicas requieren de una gran cantidad de datos y sus costos computacionales son elevados.
- **Macroscópicos:** a diferencia del enfoque microscópico, se utiliza un modelo con menor nivel de detalle y no se consideran los vehículos individualmente, sino que se representa el flujo vehicular como un flujo compresible con las propiedades de tasa de flujo, velocidad y densidad. La principal ventaja de las simulaciones macroscópicas radica en los bajos costos computacionales debido a la baja complejidad, manteniendo las propiedades esenciales del tráfico.
- **Mesoscópicos:** los simuladores mesoscópicos llenan la brecha entre los macro y micro simuladores. Normalmente describen las entidades del tráfico con un

nivel de detalle mayor que los modelos macroscópicos aunque manejan los comportamientos e interacciones en forma macroscópica.

- **Nanoscópicos:** los simuladores nanoscópicos añaden mayor especificación al modelo microscópico, en especial respecto a los vehículos, los cuales se modelan como un conjunto de partes.

3.2.5. Formulación general de la planificación de ciclos de semáforos

En este proyecto se aborda el problema de planificación de ciclos de semáforos desde una perspectiva multiobjetivo, considerando mejoras en el flujo vehicular. El problema de optimización propone encontrar una configuración apropiada de ciclos de semáforos para una infraestructura dispuesta en una zona urbana, con el fin de mejorar las condiciones del tráfico y la polución.

El modelo propuesto considera escenarios de tráfico en los que no ocurren accidentes. Los escenarios están compuestos por una o más intersecciones gobernados por un único programa de ciclo. Los semáforos dentro de una misma intersección se encuentran sincronizados y no permiten configuraciones inviables.

Las configuraciones de semáforos son ingresadas en un microsimulador de tráfico junto con la descripción total de la zona en estudio y los viajes de cada uno de los vehículos para un tiempo de simulación establecido. El simulador devuelve las salidas necesarias para el cálculo de las emisiones emitidas y otros valores que permiten conocer las condiciones del tráfico durante la simulación.

A continuación se presenta una formulación matemática del problema de planificación de ciclos de semáforos.

3.2.6. Formulación matemática del problema

Dados los siguientes elementos:

- Una red dirigida $G(N, A)$ donde N es el conjunto de nodos y A es el conjunto de arcos dirigidos en la red.
- Un conjunto de nodos de entrada $N_e \subset N$ y un conjunto de nodos de salida $N_s \subset N$.
- Un conjunto de programas de ciclos $P = \{p_1, p_2, \dots, p_n\}$, con $n \leq |N|$. Para cada programa p_i existe un conjunto de estados asociado $E_{p_i} = e_{i1}, e_{i2}, \dots, e_{ik_i}$, donde $e_{ij} \in E$ con $1 \leq j \leq k_i$ y un conjunto de nodos $N_{p_i} \subseteq N$ semaforizados tal que p_i es la configuración de los semáforos.
- Una matriz $M \in \mathbb{N}^{|N_e| \times |N_s|}$ que represente la tasa de flujo vehicular para cada arco de la red de tráfico (se conocen los conjuntos de caminos y su proporción de elección entre cada par OD).

- Un conjunto de vehículos $V = \{v_1, v_2, \dots, v_m\}$ en conformidad con la matriz M . Cada vehículo v_h cuenta con tiempo de salida ts_h y ruta $r_h = l_{h1}, l_{h2}, \dots, l_{hq_h}$, donde $l_{hr} \in A$ con $1 \leq r \leq q_h$.
- Una función $VD(v, t)$ con $v \in V$ que indica si el vehículo llegó a destino en tiempo de simulación t .
- Una función $TP(v, t)$ con $v \in V$ que devuelve la cantidad de tiempo perdido por el vehículo v como consecuencia de ir por debajo de la velocidad ideal.
- Un valor para el tiempo total de simulación t_{SIM} .

El problema plantea la búsqueda de una función $f_1 : E \rightarrow \mathbb{R}$ tal que para cada estado de luces e_{ij} devuelva su duración d_{ij} y $f_2 : P \rightarrow \mathbb{R}$ tal que para cada programa p_i devuelva el tiempo de inicio del programa o_i (pueden existir restricciones de duración), de forma tal de maximizar la cantidad de vehículos que llega a destino al finalizar la simulación (Ecuación 3.4a) y minimizar el tiempo global perdido por ir debajo de la velocidad ideal para aquellos vehículos que llegaron a destino (Ecuación 3.4b).

$$\text{Maximizar } VDG_{t_{SIM}} = |\{v \in V | VD(v, t_{SIM})\}| \quad (3.4a)$$

$$\text{Minimizar } TPG_{t_{SIM}} = \sum_{v \in VD(t_{SIM})} TP(v, t_{SIM}) \quad (3.4b)$$

3.3. Estimación de una matriz OD utilizando conteos de tráfico

Convencionalmente, una matriz OD se estima a partir de una serie de extensas encuestas, incluyendo entrevistas en la calle, reconocimiento de matrículas y encuestas domiciliarias. Estos métodos son costosos, requieren mucho tiempo y mano de obra intensiva. Por consiguiente, en las últimas tres décadas muchos investigadores (Van Zuylen and Willumsen, 1980; Fisk, 1988; Yang et al., 1992; Bell et al., 1997; Hazelton, 2000; Chootinan et al., 2005; Ehlert et al., 2006) se enfocaron en la estimación de la matriz OD a través de conteos de tráfico. Este proceso puede ser visto como el problema inverso al *Problema de Asignación de Tráfico* (Bell and Iida, 1997) ya que consiste en estimar la matriz OD de forma tal que, al asignarla a la red, los viajes de origen a destino reproducen los conteos observados.

Los conteos de tráfico se recogen a menudo para monitorear la circulación del tráfico, midiendo el número de vehículos que pasan por un punto (o estación de

medición) durante un período de tiempo específico. Por lo general se llevan a cabo para controlar y describir las características del tráfico (Garber and Hoel, 1999), tales como la *media anual de tráfico diario*, *tráfico diario medio*, *volumen en horas pico*, *millas viajadas en automóvil*, etc. Además, los conteos de tráfico se pueden utilizar eficientemente para estimar una matriz OD. Esta es considerada una forma conveniente y confiable de obtener información actualizada de los patrones de viaje en una región (ver Cascetta and Postorino (2001) para una descripción de los métodos).

3.3.1. Localización de conteos de tráfico

El problema tradicional de *Localización de Conteos de Tráfico* (TCL, por sus siglas en inglés) refiere al problema de seleccionar la ubicación de obtención de los conteos de tráfico y se puede considerar como un preproceso del problema de estimación OD. Existen varios métodos para realizar los conteos que van desde métodos manuales, a conteos totalmente automáticos. Los métodos automáticos evitan la intervención humana y aceleran el proceso mediante el uso de instrumentos como tubos neumáticos o dispositivos de contacto electrónico, etc (ver ejemplos reales de instrumentos en Garber and Hoel (1999)). Los métodos manuales tienen la desventaja de requerir mano de obra intensiva y estar limitados por el error humano, además de ser por períodos cortos de tiempo. Sin embargo, se pueden realizar en cualquier lugar sin utilizar ningún instrumento.

Distintos estudios han revelado que la calidad de la matriz OD estimada depende en gran medida del número y la ubicación de las estaciones de conteo de tráfico (Yang et al., 1991; Yang and Zhou, 1998; Chootinan et al., 2005; Gan et al., 2005; Ehlert et al., 2006). Yang et al. (1991) observaron que conjuntos diferentes de conteos de tráfico no contienen la misma cantidad de información. La mejor opción para obtener una estimación confiable es realizar conteos en todos los arcos (segmento de ruta entre intersecciones), pero esto no siempre es posible debido a limitaciones de tiempo o recursos. Se debe, por lo tanto, realizar una selección que priorice algunos arcos de la red, aquellos arcos más informativos; es necesario encontrar un criterio de selección que permita elegir un número óptimo de ubicaciones de conteo. Los datos recogidos de estos arcos contribuyen significativamente a una estimación más precisa de los flujos OD entre otras características del tráfico en la zona de estudio.

Yang and Zhou (1998) llevaron a cabo una investigación exhaustiva sobre la importancia de la ubicación de los conteos de tráfico para una estimación efectiva de la tabla de viajes OD. Sobre la teoría del *error relativo máximo posible*, derivaron cuatro reglas para localizar los puntos de conteo:

- *Cobertura OD*: los puntos de conteo de tráfico deben estar ubicados de forma tal que se observe cierta porción de los viajes entre todo par OD.
- *Fracción máxima de flujo*: para un par OD en particular, los puntos de conteo deben estar ubicados de forma tal que la fracción de flujo entre este par OD sea lo más grande posible.

- *Intercepción máxima de flujo*: los arcos seleccionados deben interceptar el mayor flujo posible.
- *Regla de independencia de arcos*: los puntos de conteo de tráfico deben estar ubicados de forma tal que los conteos de tráfico resultantes sean linealmente independientes.

En la investigación señalada, el problema de localizar puntos de conteo en una red se formuló con un modelo de programación entera, donde la regla de cobertura OD y la regla de independencia de arcos son incorporadas como restricciones mientras que el flujo total de tráfico observado se toma como función objetivo a maximizar.

Yang et al. (2001) propusieron modelos TCL para seleccionar en forma óptima estaciones de conteo en una red basándose en la *regla de separación OD* sin necesidad de referencias explícitas a flujos OD existentes, flujos de rutas, proporciones de giro o suposiciones de comportamiento. Un par OD se considera *separado* si y sólo todos los viajes que conectan origen con destino pasan por al menos una estación de conteo seleccionada. Se generaron modelos de programación entera y se desarrolló un AG para resolver dos problemas: el primero busca un número dado de estaciones de conteo para separar la mayor cantidad de pares OD posible, el segundo busca determinar el número mínimo de estaciones de conteo para separar todos los pares OD.

Ehlert et al. (2006) debatieron sobre varias formulaciones de programación entera binaria para minimizar la cantidad de estaciones que satisfacen las reglas establecidas por Yang and Zhou (1998). El software OptTool desarrollado por Ehlert et al. (2006) incorpora la opción de asignar pesos a los pares OD sobre la base del contenido de información de los movimientos OD.

3.3.2. Ordenamiento de arcos basado en flujo previo

Como se mencionó en la sección anterior, la exactitud de la matriz OD estimada se ve afectada por la selección de los arcos de entrada. Se sabe que conjuntos diferentes de arcos de entrada producen matrices de viaje diferentes (Mussone et al., 2010). No obstante, la priorización de arcos es un aspecto pasado por alto en muchos estudios. La mayoría de los investigadores opina que la selección debe satisfacer la regla de cobertura OD, pero al existir restricciones presupuestarias, no siempre es posible realizar conteos en los arcos que satisfacen la regla. Por dicho motivo, el número óptimo de arcos que satisfacen la regla de cobertura se deben clasificar en orden de importancia de forma tal que sea posible elegir los arcos más importantes de acuerdo a los fondos disponibles. Generalmente, a los arcos se les asigna un peso correspondiente al costo de realizar el conteo (instalación de detectores, etc).

En el marco de este proyecto, se desarrolló una estrategia de ordenamiento o priorización de arcos en una red, con el objetivo de construir una matriz OD. Se reformuló la propuesta de Yang et al. (2001), asignando determinado *peso* o

ranking a los arcos siguiendo el enfoque señalado en Saraswathy and Isaac (2013) para los casos en los que se dispone de datos de flujo previo. Dados los elementos especificados en la Sección 3.2.3 se considera el siguiente problema de minimización:

$$\text{Minimizar } Z_1 = \sum_{l \in A} C(l) \gamma_l \quad (3.5)$$

sujeto a,

$$\sum_{l \in A} \delta_{r,l}^w \gamma_l \geq 1, \quad \forall r \in R_w, w \in W \quad (3.6)$$

donde $C(l)$ es el costo del conteo de tráfico para el arco l y $\gamma_l = 1$, si el arco l es seleccionado, 0 sino.

Al tratarse de un problema de minimización, cuanto menor sea el coeficiente de γ_l en la ecuación 3.5, más alta es la probabilidad de selección del arco l . Luego, si $C(l)$ es reemplazado por el inverso del peso del arco $P(l)$, el arco con más peso tiene más probabilidad de ser elegido. Se puede entonces reformular el problema de la siguiente forma:

$$\text{Minimizar } Z_2 = \sum_{l \in A} \frac{1}{P(l)} \gamma_l \quad (3.7)$$

sujeto a la Ecuación 3.6.

Para ajustar el peso $P(l)$ de cada arco se utiliza el *volumen por intercambio* del arco basado en el volumen de tráfico. Si se dispone de datos previos de flujo, estos pueden ser usados como valor de ajuste. Se tiene:

$$P(l) = \frac{V(l)}{OD(l)} \quad (3.8)$$

donde, $V(l)$ es el volumen de tráfico del arco l partir de datos anteriores y $OD(l)$ es el número de pares OD que usan el arco l .

La solución al problema de minimización propuesto resulta en la selección de un conjunto de arcos de costo mínimo $A_M \subseteq A$ que separan todos los pares OD. Esto es, todos los viajes que conectan un origen con un destino pasan por al menos uno de los arcos del conjunto. La regla de separación OD es un caso particular de la regla de cobertura OD donde se observa el total de viajes entre todo par OD. La solución al problema cumple entonces la regla de cobertura OD. Se puede probar además que cumple la regla de independencia de arcos por tratarse de un problema de minimización (ver demostración en Yang and Zhou (1998)). Es importante mencionar que no todas las reglas de locación citadas pueden ser completamente cumplidas. En general la regla de cobertura OD es considerada una regla fundamental para una estimación efectiva de la matriz OD y debe ser satisfecha. La regla de independencia de arcos excluye aquellos arcos que no aportan información nueva y por tanto también debe satisfacerse. Así pues, es razonable que estas reglas se traten como restricciones a la hora de determinar las locaciones de conteo.

Por otro lado, las reglas de fracción máxima de flujo e intercepción máxima de flujo, darán lugar a conflictos entre los diferentes pares OD y pueden no ser deseables en la selección de arcos. Por ejemplo, contar en un arco de bajo volumen puede ser poco justificado, sin embargo, este arco puede contener una fracción alta de viajes entre otro par OD. El peso volumen por intercambio fue propuesto en compromiso con estas dos reglas. Otra opción efectiva de priorización es establecer un valor umbral de fracción de flujo OD, a partir del cual el par OD se dice efectivamente cubierto. Luego los arcos candidatos son ordenados en forma descendiente de acuerdo al número de pares OD efectivamente cubiertos.

Una vez resuelto el problema de minimización y obtenido el conjunto A_M de arcos, se procede a realizar conteos en la mayor cantidad de arcos posibles del conjunto, comenzando por aquellos con mayor peso. Cuantos más conteos se realicen mejor será la calidad de la matriz OD estimada. La cantidad de conteos estará limitada por los recursos disponibles.

Sea $A_C \subseteq A_M$ el conjunto de arcos donde se realizaron conteos y sea $conteo(l)$ el resultado del conteo para el arco l . La obtención de la matriz a partir de los conteos se puede plantear como un problema de minimización que minimice la diferencia entre los conteos y el volumen generado por la matriz solución M :

$$\text{Minimizar} \quad \sum_{l \in A_C} |conteo(l) - volumen_M(l)|. \quad (3.9)$$

3.4. Contaminación atmosférica

Se entiende por contaminación atmosférica a la presencia en el aire de uno o más contaminantes o sus combinaciones, en cantidades tales y con duración tal que puedan o tiendan a afectar la vida humana, de animales o de plantas o de la propiedad que interfiera el goce de la vida o el ambiente (Wark and Warner, 1981). La contaminación atmosférica ocupa un lugar preferente en la atención de la comunidad científica y en las inquietudes de la ciudadanía por la preservación del medio ambiente, tema de capital importancia en las sociedades avanzadas de nuestro tiempo (Ataz and de Mera Morales, 2004).

Los problemas de contaminación atmosférica son tan antiguos como la Tierra misma. Los volcanes, fumarolas, incendios naturales, polvo y arena de tormentas en regiones áridas han contribuido a la contaminación del aire en forma natural (fuentes naturales). Sin embargo, a excepción de casos extremos como las erupciones volcánicas, la contaminación natural por sí misma no representa un peligro preponderante pues existen procesos naturales que logran asimilar estos contaminantes consiguiendo un equilibrio en los ecosistemas. Por otro lado, el ser humano ha emitido contaminantes al aire desde sus orígenes (fuentes antropogénicas), pero las emisiones se incrementaron de manera dramática a partir de la Revolución Industrial iniciada en el Reino Unido a finales del siglo XVII. Más recientemente, la quema de carbón, productos químicos, petróleo, gasolina, queroseno, diésel, combustible alcohólico y de aviación, gas natural, biomasa y residuos, así como la liberación de sustancias químicas en el medio ambiente, han contribuido en varios

de los principales problemas de contaminación atmosférica en un rango de escalas espaciales. Estos problemas incluyen el *smog* urbano, la contaminación del aire interior, la deposición ácida, el agujero de ozono antártico, la reducción del ozono estratosférico y el calentamiento global (Jacobson, 2012).

Países de todo el mundo, especialmente aquellos en desarrollo, están experimentando un rápido proceso de urbanización. La proporción de la población mundial que vive en ciudades ha crecido de un 35 por ciento en 1970 a casi el 50 por ciento en 2001, y se espera que esta cifra aumente a más del 60 por ciento para el 2030 (United Nations Centre for Human Settlements (Habitat), 2001). Una de las muchas consecuencias del aumento de la actividad económica que acompaña a la urbanización (particularmente el aumento del uso de vehículos, generación de electricidad y producción industrial) es el deterioro de la calidad del aire (Molina and Molina, 2004).

A pesar de la creciente evidencia epidemiológica que apoya la asociación entre la contaminación atmosférica y la morbilidad y mortalidad cardiovascular (Kappos et al., 2004; Brook et al., 2010), las concentraciones de contaminantes atmosféricos convencionales, continúan aumentando en muchas ciudades y en muchos casos están muy por encima de las guías para la calidad del aire de la Organización Mundial de la Salud (OMS). En el año 2012 alrededor de 7 millones de personas murieron –una de cada ocho del total de muertes en el mundo- como consecuencia de la exposición a la contaminación atmosférica (World Health Organization, 2014). Este dato confirma que la contaminación atmosférica constituye en la actualidad, por sí sola, el riesgo ambiental para la salud más importante del mundo.

La aprobación de varias leyes promulgadas por diferentes gobiernos en la mayoría de los países han logrado una disminución notable de la contaminación atmosférica de origen industrial y doméstica (calefacciones). Sin embargo, el crecimiento sustancial del tráfico motorizado ha aumentado los niveles de contaminación del aire en los centros urbanos (Sharma and Khare, 2001).

3.4.1. Contaminantes atmosféricos

Los contaminantes atmosféricos se definen por lo general como sustancias que alteran la composición de la atmósfera, ya sea porque se encuentran en cantidades diferentes a las normales o porque se encuentran en una región a la que no pertenecen, resultando en efectos adversos para los humanos o el ecosistema. Por ejemplo, el *ozono* (O_3) que en la estratosfera actúa como escudo contra los rayos UV, en la troposfera se considera un contaminante ya que resulta peligroso para la vida al ser un potente oxidante. Otros contaminantes pueden causar que importantes compuestos que están presentes de forma natural sean menos abundantes, como sucede con la reducción del ozono estratosférico debido a sustancias químicas hechas por el hombre. Unos pocos contaminantes se combinan con vapor de agua para convertirse en ácidos. El exceso de calor también puede ser considerado un contaminante, resultando en aumento de temperatura en zonas urbanas y tiempo climático más variable.

El Gobierno federal de Estados Unidos de América, a través de la Agencia de Protección Ambiental (EPA, por sus siglas en inglés), establece estándares para proteger el bienestar y la sanidad humana así como el medio ambiente de la contaminación atmosférica. Se controlan un total de 189 contaminantes, entre los cuáles se identifican los seis más importantes utilizando criterios específicos de acuerdo a las características de los contaminantes y sus posibles efectos en la salud y bienestar (a nivel mundial). Estos son: CO, *dióxido de azufre* (SO_2), NO_x , O_3 , *partículas o material particulado* (PM_{10} y $\text{PM}_{2.5}$) y *Plomo* (Pb). La EPA llama a estos los *seis contaminantes principales* o *seis contaminantes criterio* (US EPA, 2015a). Para minimizar el impacto a la salud o al medioambiente, EPA y la OMS han fijado la *máxima concentración atmosférica recomendable* para los contaminantes criterio. Estos límites de concentración se conocen como *Estándares Nacionales de Calidad del Aire Ambiental* (US EPA, 2015b) y varían con cada contaminante. Otros contaminantes importantes son los químicos orgánicos e inorgánicos y los hidrocarburos mal quemados (HC) los cuáles pertenecen a un grupo más grande de sustancias conocidas como compuestos volátiles orgánicos (VOC).

Existe una serie de elementos que intervienen en los procesos de contaminación del aire. La Figura 3.4 muestra los mecanismos de este tipo de contaminación. Se observa que el primer proceso corresponde a la emisión de contaminantes, cuyo nivel es variable en función de las características de los focos emisores y de las condiciones en que se produce dicha emisión. Los valores de emisión se miden a la salida de la fuente emisora. La evolución de los contaminantes en la atmósfera continúa a través de su transporte, difusión, mezcla o acumulación (mecanismos sumidero), procesos llevados a cabo por los fenómenos meteorológicos que determinan la capacidad dispersante de la atmósfera. Por otro lado ocurren un conjunto de reacciones químicas que dan lugar a la transformación de los contaminantes, puesto que la atmósfera es un medio oxidante, que recibe energía solar y contiene sustancias químicas de alto poder oxidante (radicales hidroxilo u otros radicales libres). El resultado final de estos procesos permite establecer los valores de inmisión de los contaminantes, que indicarán la calidad del aire. Se conoce como *inmisión* a la cantidad de contaminantes presentes en una atmósfera determinada, una vez que han sido transportados, difundidos, mezclados en ella y a los que están expuestos los seres vivos y los materiales que se encuentran bajo su influencia. Si los niveles de inmisión no son adecuados, disminuye la calidad del aire y se originan efectos negativos sobre los distintos receptores: el ser humano, los animales, los vegetales y los ecosistemas en su conjunto. Los factores que influyen en la dinámica de dispersión de los contaminantes, y por tanto en su concentración, son las características de las emisiones, las condiciones atmosféricas, las características geográficas y topográficas.

El tráfico vehicular, las industrias y el uso doméstico de la energía son las principales fuentes antropogénicas causantes de contaminación atmosférica urbana. En los tres casos, el problema deriva de la quema de combustibles fósiles (produce los seis contaminantes criterio), carbón o petróleo, como fuente de energía, pasando los residuos de la combustión a la atmósfera, que luego pueden retirarse por depo-

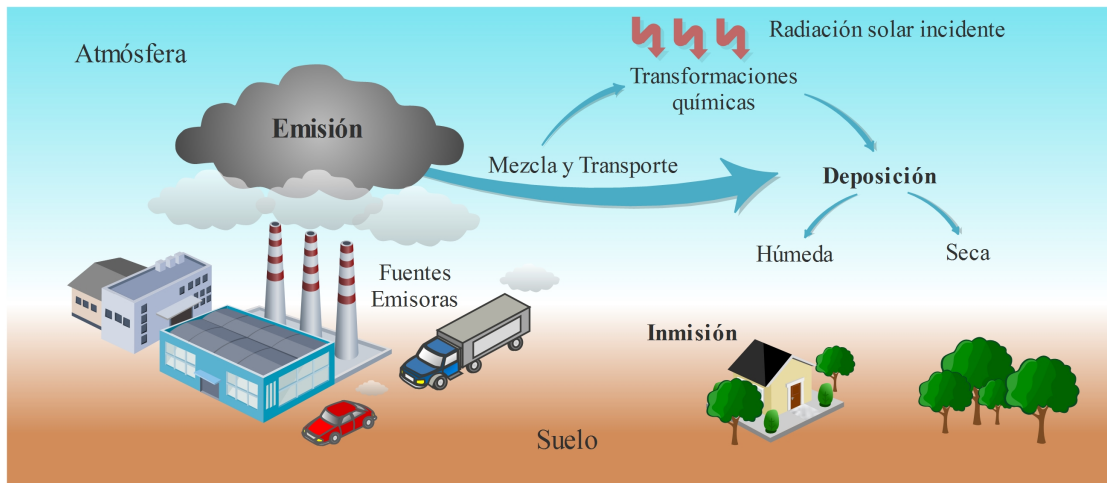


Figura 3.4: Mecanismos de la contaminación atmosférica.

sición seca o húmeda o reaccionar con otros componentes como muestra la Figura 3.4 (Molina and Molina, 2004).

3.4.2. Contaminación vehicular

Los altos ingresos, movilidad, expansión de ciudades y proliferación de centros de empleo han aumentado la demanda de transporte motorizado, resultando en una concentración desproporcionada de vehículos en los centros urbanos. Según un estudio realizado por la organización WardsAuto, el número mundial de automóviles registrados alcanzó la cifra de 1.015 millones en 2010 (vehículos livianos, medianos, camiones pesados y ómnibus), superando los 980 millones de 2009, lo que representa un incremento de producción/venta/circulación de un 3,6 %, el segundo incremento más alto en la historia triplicando la tasa de crecimiento de la población (1.18 %) para el mismo año (Sousanis, 2011; Bremner et al., 2010). Se estima además que para el año 2035 la flota mundial de vehículos livianos rebasará el total de 2 mil millones de los cuales solo un 2.5 % serán eléctricos, híbridos *plug-in* o de pilas de combustible, mientras que el resto funcionaran a gasolina o diésel (Navigant Research, 2015).

La contaminación atmosférica a causa del tráfico vehicular se ha convertido en motivo importante de preocupación en regiones de rápida urbanización caracterizadas por alta densidad de población e intensa actividad económica. Esto es debido al aumento en el número de vehículos en uso y la distancia recorrida cada año por cada vehículo (Fenger, 1999). En muchas áreas, las emisiones vehiculares se han vuelto una fuente dominante de contaminantes atmosféricos, incluyendo: CO, *dióxido de carbono* (CO_2), VOC, NO_x y PM_x (Transportation Research Board and National Research Council, 2002). Otros contaminantes vehiculares comunes son: SO_2 y Pb. En total, el tráfico vehicular es fuente emisora de 5 contaminantes criterio. Otros contaminantes secundarios asociados con las emisiones de vehículos de

motor incluyen oxidantes fotoquímicos (por ejemplo, el ozono), ácidos sulfúricos y nítricos (Onursal and Gautam, 1997).

A continuación se revisan brevemente las características de los principales agentes contaminantes atmosféricos vehiculares y sus efectos sobre las personas y los ecosistemas.

- **CO:** es un gas incoloro, insípido e inodoro a concentraciones atmosféricas, producido por la combustión incompleta de cualquier tipo de combustible, incluyendo la mayor parte de los combustibles empleados en el transporte. Esto generalmente sucede producto de una inadecuada proporción de aire-combustible debido a un equipo mal operado o con poco mantenimiento, limitaciones en el flujo de aire o bajas temperaturas. Los automóviles con motores de combustión interna son la principal fuente de emisión de CO en las zonas urbanas (hasta un 95 %). Por sí mismo, representa más del 50 % de la contaminación del aire en peso a nivel mundial. El CO puede persistir en la atmósfera durante uno o dos meses antes de convertirse en CO₂ -el CO contribuye por lo tanto a CO₂ en la atmósfera- (Díaz and Pantoja, 2011; Mihelcic and Zimmerman, 2012; Hill, 2010).

Este compuesto es de los más peligrosos por su especial capacidad para combinarse con la hemoglobina y causar asfixia química, que ocasiona reducción del oxígeno en el corazón y en el cerebro, lo cual es crítico para las personas que tienen padecimientos en el corazón, en los pulmones o anemia. La exposición a esta sustancia ocasiona, dolores de cabeza, fatiga y reflejos retardados. En concentraciones altas puede causar la muerte en tiempos de exposición relativamente cortos (Díaz and Pantoja, 2011; Meyer et al., 2011).

- **CO₂:** es un gas incoloro, inodoro y vital para la vida en la Tierra, por lo que no se considera generalmente un contaminante atmosférico. Sin embargo, existe gran preocupación por las crecientes emisiones de este gas como producto de la combustión de combustibles fósiles y el uso de la tierra. El CO₂ es un gas de efecto invernadero que absorbe la radiación térmica emitida por la superficie de la tierra, evitando que escape al espacio, y la irradia en todas direcciones aumentando la temperatura de la atmósfera (Zaror, 2002).

El efecto combinado de aumento de la generación de CO₂ en la actividad humana y la destrucción de áreas boscosas, resulta en un desequilibrio que aumenta la concentración media de CO₂ en la atmósfera. Esto a su vez afecta el balance térmico terrestre dando lugar al calentamiento global, que refiere al aumento observado en los últimos siglos de la temperatura media del sistema climático de la Tierra y sus efectos relacionados (ver Weart (2006); Pardos (2010) para más información).

- **HC y VOC:** como se mencionó anteriormente, los HC pertenecen al grupo de los VOC. Mientras que los HC son compuestos exclusivamente de hidrógeno y carbón, los VOC pueden contener otros elementos. Algunos VOC se forman

naturalmente pero la mayoría se producen por la combustión incompleta de HC y también por su evaporación.

Al existir cientos de compuestos diferentes, los HC y VOC muestran una amplia gama de propiedades. Unos, como el benceno, son cancerígenos; algunos son tóxicos, mientras que otros son inofensivos para la salud. Muchos conductores y peatones desarrollan dolores de cabeza y otros síntomas al estar muy expuestos a hidrocarburos volátiles procedentes de los tubos de escape de los vehículos de motor. Las personas sensibles pueden reaccionar con ataques de asma u otros problemas respiratorios. Altas concentraciones de una sustancia química como el benceno pueden causar mareos, náuseas y otros síntomas. Muchos VOC son también contaminantes de aire peligrosos y juegan un papel importante en la formación de ozono a nivel del suelo (Hill, 2010; Ferrante et al., 2012).

- **NO_x**: los tres óxidos de nitrógeno que se encuentran comúnmente en la atmósfera son: óxido nitroso (N₂O), óxido nítrico (NO) y dióxido de nitrógeno (NO₂). Estos se denominan genéricamente NO_x. El N₂O es un gas relativamente poco reactivo, y es un componente traza de la atmósfera “natural”. El NO es un gas incoloro e inodoro; mientras que el NO₂ es café-rojizo y tiene un olor desagradable. Los óxidos de nitrógeno son producidos directa e indirectamente por procesos de combustión a altas temperaturas. En dichos procesos el N₂ presente en el aire se oxida para formar principalmente NO, el que se transforma en NO₂ mediante reacciones fotoquímicas. En las zonas urbanas, las concentraciones máximas de NO_x coinciden con los volúmenes máximos de tráfico. Los gases de escape de los motores de automóviles son ricos en NO y pobres en NO₂. Generalmente, una pequeña fracción del total de NO_x está presente como NO₂.

Los NO_x son de los principales componentes del smog urbano y su concentración aumenta en los meses de invierno. En grandes concentraciones puede irritar los pulmones, producir constricción bronquial y disminuir la resistencia ante infecciones respiratorias. Estos efectos pueden ser mayores cuando existen otros compuestos alergénicos presentes en el aire. Es también un importante precursor del ozono y de precipitaciones ácidas (Zaror, 2002; Keoleian et al., 1997).

- **PM_x**: consiste de una compleja mezcla de partículas sólidas y líquidas que están dispersas en el aire y pueden mantenerse suspendidas por largo tiempo debido a su baja velocidad de sedimentación. Se origina de fuentes naturales y antropogénicas siendo el tráfico rodado (en particular los vehículos diésel) la principal fuente de emisiones en las ciudades. El PM_x varía extensamente en su composición física y química, fuente y el tamaño de partícula. Se define PM₁₀ como la fracción de partículas en el aire de tamaño inferior a 10 μm (micrones), y partículas PM_{2,5} las de menor a 2,5 μm (Onursal and Gautam, 1997).

Tanto $PM_{2,5}$ como PM_{10} son las de mayor interés, ya que debido a su pequeño tamaño pueden penetrar profundamente en los pulmones, con potencial peligro para la salud. Una vez que las partículas entran al tracto respiratorio, algunas pueden ser atrapadas por el flujo mucoso que baña una porción del tracto respiratorio. El mucus es tragado y, por lo tanto, las partículas llegan al estómago y a los intestinos. Otras permanecen en los pulmones donde pueden rodearse de tejido y, eventualmente, ser incorporadas al flujo sanguíneo (Zaror, 2002).

- **O₃:** a diferencia de otros contaminantes, el ozono no proviene de ninguna fuente directa. En vez de eso, el ozono es creado por una compleja secuencia de reacciones químicas que tienen lugar cuando los NO_x e hidrocarburos reactivos entran en contacto en presencia de luz solar (Mihelcic and Zimmerman, 2012). Debido a que en áreas urbanas, el tráfico vehicular es responsable de más del 50 % de emisiones de NO_x y 40 % de VOC, este se establece como la principal causa de ozono. El ozono es también un gas de efecto invernadero (Hill, 2010) y un componente importante del smog que contiene otros oxidantes fotoquímicos y PM_x haciéndolo visible a los ojos (ver Figura 3.5).



Figura 3.5: En verano, una espesa capa de smog se esparce sobre la ciudad de São Paulo (Brasil).

El smog se caracteriza por la presencia de bruma y sus efectos en la salud humana son graves. Una exposición a concentraciones elevadas puede causar tos, molestias en la garganta, constricción de las vías respiratorias, laringitis, alteración de los glóbulos rojos e irritación en ojos y nariz. Se calcula que la exposición al ozono causa mas de 21.000 muertes prematuras al año en 25 países de la Unión Europea (Amann, 2008). También tiene un efecto perjudicial en los ecosistemas, interfiriendo con el metabolismo de las plantas y

dañando las cosechas. Tal contaminación puede reducir la biodiversidad en un ecosistema (Díaz and Pantoja, 2011).

3.5. Planificación de ciclos de semáforos considerando emisiones vehiculares

En esta sección se modifica la formulación general realizada en la sección 3.2.5 para considerar emisiones vehiculares. Nuevamente se aborda el problema de planificación de ciclos de semáforos siguiendo una perspectiva multiobjetivo y se mantienen los objetivos que mejoran las condiciones del tráfico. En este caso, el micro-simulador de tráfico devuelve además las salidas necesarias para el cálculo de las emisiones emitidas por los vehículos.

A continuación se presenta una formulación matemática del problema de planificación de ciclos de semáforos considerando emisiones vehiculares.

3.5.1. Formulación matemática del problema considerando emisiones vehiculares

Dados los siguientes elementos:

- Una red dirigida $G(N, A)$ donde N es el conjunto de nodos y A es el conjunto de arcos (links) dirigidos en la red.
- Un conjunto de nodos de entrada $N_e \subset N$ y un conjunto de nodos de salida $N_s \subset N$.
- Un conjunto de programas de ciclos $P = \{p_1, p_2, \dots, p_n\}$, con $n \leq |N|$. Para cada programa p_i existe un conjunto de estados asociado $E_{p_i} = e_{i1}, e_{i2}, \dots, e_{ik_i}$, con $e_{ij} \in E$ y un conjunto de nodos $N_{p_i} \subseteq N$ semaforizados tal que p_i es la configuración de los semáforos.
- Una matriz $M_f \in \mathbb{N}^{|N_e| \times |N_s|}$ para cada tipo de vehículo tv considerado que represente la tasa de flujo vehicular para cada arco de la red de tráfico.
- Un conjunto de vehículos $V = \{v_1, v_2, \dots, v_m\}$ en conformidad con las matrices M_f . Cada vehículo v_h cuenta con tiempo de salida ts_h , tipo de vehículo tv_h y ruta $r_h = l_{h1}, l_{h2}, \dots, l_{hq_h}$, con $l_{hr} \in A$.
- Una función $VD(v, t)$ con $v \in V$ que indica si el vehículo llegó a destino en tiempo de simulación t .
- Una función $ET(c, t)$ que devuelve la cantidad total de contaminante c emitido al tiempo de simulación t .
- Una función $TP(v, t)$ con $v \in V$ que devuelve la cantidad de tiempo perdido por el vehículo como consecuencia de ir por debajo de la velocidad ideal.

- Un valor para el tiempo total de simulación t_{SIM}

El problema plantea la búsqueda de una función $f_1 : E \rightarrow \mathbb{R}$ tal que para cada estado de luces e_{ij} devuelva su duración d_{ij} y $f_2 : P \rightarrow \mathbb{R}$ tal que para cada programa p_i devuelva el tiempo de inicio del programa o_i (pueden existir restricciones de duración), de forma tal de maximizar la cantidad de vehículos que llega a destino al finalizar la simulación (Ecuación 3.10a), minimizar el tiempo global perdido por ir debajo de la velocidad ideal para aquellos vehículos que llegaron a destino (Ecuación 3.10b) y minimizar la emisión total para todos los contaminantes c considerados (Ecuación 3.10c).

$$\text{Maximizar } VDG = |\{v \in V | VD(v, t_{SIM})\}| \quad (3.10a)$$

$$\text{Minimizar } TPG = \sum_{v \in VD(t_{SIM})} TP(v, t_{SIM}) \quad (3.10b)$$

$$\text{Minimizar } ETG_c = ET(c, t_{SIM}) \quad \forall c \in C \quad (3.10c)$$

3.6. Trabajos relacionados

La combinación de metaheurísticas y algoritmos bioinspirados con otras técnicas se ha vuelto popular para resolver problemas relacionados a la planificación de ciclos de semáforos. Un primer enfoque es el brindado por Rouphail et al. (2000), donde un AG es combinado con el microsimulador CORSIM (Halati et al., 1997) para optimizar los ciclos de semáforos de nueve intersecciones en la ciudad de Chicago en EE.UU.. Los resultados, en términos de tamaño total de cola están limitados debido la convergencia retardada del algoritmo propuesto por los autores.

En Teklu et al. (2007), se analiza el impacto de los cambios de tiempo de una señal con respecto a los conductores de los vehículos. Más precisamente, los autores analizan el problema de determinar el tiempo óptimo de los ciclos de semáforos mientras anticipan la respuesta de los conductores a dichos cambios. Para resolver el problema utilizan el paquete SATURN (Simulation-Assignment Modeling Software, Van Vliet (1982)) y una visión macroscópica del flujo vehicular mientras emplean un AG para computar la duración de los ciclos.

En Sánchez et al. (2008), siguiendo el modelo propuesto por Brockfeld et al. (2001), los autores diseñan un AG con el objetivo de optimizar los ciclos de semáforos. Este algoritmo es testado sobre el área comercial de Santa Cruz de Tenerife en España. Cada intersección cuenta con un ciclo de semáforos independiente y se utiliza una representación de soluciones similar a la de Teklu et al. (2007). El cálculo que determina los estados válidos de colores en las intersecciones debe realizarse para cada escenario antes de utilizar el algoritmo y depende fuertemente del escenario elegido. Turkey et al. (2009) utilizan un AG para mejorar el rendimiento de los semáforos y los peatones en un área definida por 4 calles de dos vías. Otro

trabajo relacionado	consideración de emisiones	método de optimización
Rouphail et al. (2000)	No	AG
Chen and Xu (2006)	No	PSO
Teklu et al. (2007)	No	AG
Sánchez et al. (2008)	No	AG
Turky et al. (2009)	No	AG
Peng et al. (2009)	No	PSO
Li et al. (2009)	Si	sistema de alerta (ADAS)
Stevanovic et al. (2009)	Si	AG (VISGAOST)
Tielert et al. (2010)	Si	comunicación semáforo a vehículo
Asadi and Vahidi (2011)	Si	control basado en optimización
De Coensel et al. (2012)	Si	onda verde
Lv et al. (2013)	Si	AG
Zhou and Cai (2014)	Si	AG
Olivera et al. (2015)	Si	PSO

Cuadro 3.1: Resumen de trabajos relacionados al problema de planificación de ciclos de semáforos

trabajo relacionado con algoritmos bioinspirados es el de Chen and Xu (2006), donde se aplica un algoritmo de optimización por enjambre de partículas (PSO, por sus siglas en inglés) para entrenar una lógica difusa (*fuzzy logic*) localizada en cada intersección a fin de determinar el tiempo efectivo en verde para cada ciclo de semáforos. Para testear este PSO se utilizó una red muy sencilla con dos intersecciones. Recientemente, Peng et al. (2009) presentaron un PSO con separación de nichos para la planificación de ciclos de semáforos. En esta aproximación, se adopta una visión microscópica del flujo vehicular con el objetivo de evaluar las soluciones. Una sola instancia académica con una calle de una vía y dos intersecciones fue utilizada para testear el PSO. Sin embargo, el estudio esta enfocado en la capacidad de los nichos separados de mantener la diversidad de la población del PSO y no en un problema concerniente al tráfico urbano.

Si bien estudios empíricos han demostrado la relación entre las emisiones vehiculares y los retrasos causados por las señales de control de tráfico (Unal et al., 2003; Coelho et al., 2005), existen pocos trabajos que consideren directamente la planificación de ciclos de semáforos como forma de disminuir la polución urbana. Entre los más relevantes relacionados con la problemática se encuentra a Li et al. (2009), donde se propone un avanzado sistema de alerta para los conductores con el objetivo de evitar frenadas bruscas en las intersecciones y potencialmente reducir las emisiones vehiculares. Se define además un método de evaluación del consumo de combustible y de las emisiones.

Por su parte, Stevanovic et al. (2009) utilizan tres herramientas del estado del arte comerciales (VISSIM, CMEM y VISGAOST) para optimizar las señales de tráfico y minimizar tanto el consumo de combustible como disminuir las emisiones.

Este trabajo estudia 14 intersecciones semaforizadas en Utah, EE.UU. Tielert et al. (2010) muestran que las comunicaciones de tráfico a vehículos permiten reducir el impacto ambiental del tráfico vehicular ayudando a los conductores a evitar que frenen o aceleren en intersecciones semaforizadas. Asadi and Vahidi (2011) obtienen una reducción del 47 % del consumo de combustible al evaluar 9 intersecciones en fila. Se propone un algoritmo de control basado en optimización que utiliza información de ciclos por adelantado dentro del sistema de auto-crucero de los vehículos. De Coensel et al. (2012) combinan el simulador microscópico de tráfico Paramics (Cameron and Duncan, 1996) con modelos de emisiones y muestran que la introducción de una onda verde puede reducir las emisiones consideradas hasta en un 10 %-40 % en condiciones favorables. Lv et al. (2013) desarrollan una metodología de optimización de ciclos utilizando un AG cuya función objetivo considera tanto demoras como emisiones. Se utiliza un modelo de emisiones que considera la velocidad de los vehículos segundo a segundo y se analizan los beneficios en la calidad del aire bajo distintas condiciones de tráfico. Zhou and Cai (2014) optimizan los ciclos para reducir las emisiones vehiculares, el consumo de combustible y la demora de los vehículos en las intersecciones. El modelo de optimización multiobjetivo combina un AG con el simulador Paramics y un modelo de emisiones.

Por último, Olivera et al. (2015) utiliza un PSO para la planificación de ciclos de semáforos en la ciudades de Sevilla y Málaga. La función mono-objetivo contempla a la polución como un aspecto a optimizar aunque posee problemas para manejar la visión global del tráfico por ejemplo, en el caso de atascos que disminuyan el flujo vehicular —los automóviles no se mueven—, una solución puede ser considerada muy buena desde el punto de vista de la reducción de contaminación dependiendo del peso que tenga este término.

En el Cuadro 3.1 se exponen los distintos estudios mencionados. Todos estos enfoques están ligados a diferentes aspectos del problema de planificación del tráfico. En comparación con la propuesta planteada en este proyecto, los estudios están focalizados en instancias pequeñas o requieren de una inversión de capital para el vehículo o para el municipio. Además, por lo general el tratamiento del tráfico se realiza con algoritmos mono-inspirados que no consideran el impacto de la polución urbana dentro del problema global del tráfico y las que lo consideran muestran un impacto limitado en el aprovechamiento de las programaciones de los semáforos para reducir las emisiones.

Capítulo 4

Estrategia de resolución

El propósito de este capítulo es describir la estrategia de resolución desarrollada para abordar el problema de planificación de ciclos de semáforos considerando emisiones vehiculares. Se describen aquellos componentes de software principales implementados y/o utilizados incluyendo los AEMO seleccionados para la obtención de soluciones.

4.1. Introducción

La investigación desarrollada en el proyecto propone una estrategia de optimización utilizando AEMO para obtener ciclos de semáforos eficientes desde un punto de vista ambiental, siguiendo la formulación propuesta en la Sección 3.5. Se aplicaron cuatro algoritmos evolutivos: los clásicos NSGA-II y GDE3, y los más modernos, MOEA/D y ϵ -NSGA-II (ver Sección 2.4).

En la siguiente sección se expone el procedimiento general utilizado por los AEMO para la evaluación de los individuos de la población. En la Sección 4.3 se estudia el simulador de tráfico SUMO, seleccionado para realizar las simulaciones. Luego, en la Sección 4.4 se describe el proceso de modelado de la red de tráfico, el cual consta de dos etapas bien distinguidas y que permite obtener el escenario SUMO requerido para la evaluación de los individuos. La Sección 4.5 describe la metodología desarrollada para la obtención de las matrices OD. Finalmente, en la Sección 4.6, se detallan algunos aspectos del diseño e implementación de los algoritmos aplicados, como ser los operadores evolutivos y las técnicas de paralelización empleadas. Se especifica además, la representación de las soluciones candidatas utilizada.

4.2. Procedimiento general de evaluación

Durante el proceso evolutivo los AEMO mantienen una población de individuos que representan las posibles soluciones al problema. Los individuos evolucionan de acuerdo a operaciones ejecutadas por los algoritmos aplicados. Cada vez que un algoritmo produce un nuevo individuo, éste debe ser evaluado para conocer

su nivel de aptitud. Para ello, se sigue el procedimiento ilustrado en la Figura 4.1: el individuo es decodificado y traducido al formato utilizado por el simulador de tráfico microscópico SUMO para ser evaluado mediante una simulación. Una vez finalizada la simulación, el simulador retorna los datos pertinentes para la evaluación de la solución representada: la cantidad total de contaminantes emitidos (ETG_c), cantidad de vehículos que llegaron a destino (VDG) y tiempo perdido por los vehículos como consecuencia de ir debajo de la velocidad ideal (TPG). Como se puede observar, el simulador de tráfico requiere de un escenario (escenario SUMO), que consiste de un modelo de la red de tráfico en estudio (en el formato requerido por SUMO). Este escenario contiene todos los elementos destacados en la Sección 3.2.1, los cuales deben ser tenidos en cuenta al momento de obtener una solución realista al problema de optimización.

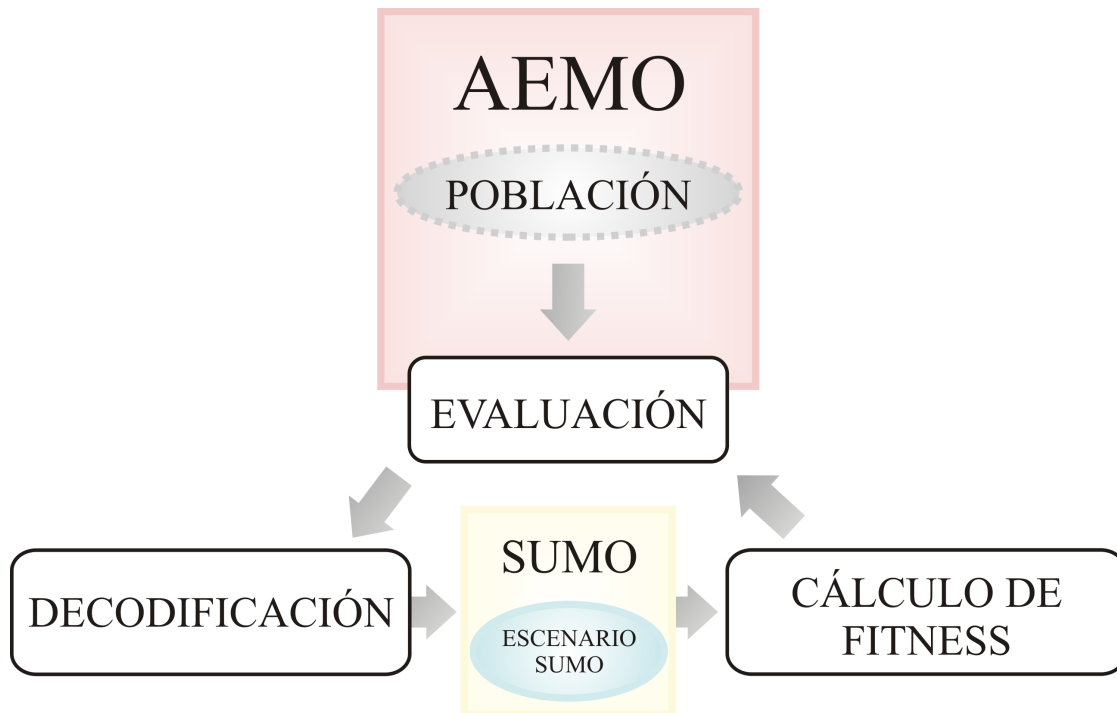


Figura 4.1: Procedimiento general utilizado por los algoritmos evolutivos para evaluar los individuos de la población.

El tráfico vehicular se encuentra fuertemente condicionado por el deseo privado de movilidad de los conductores así como de la infraestructura subyacente. Los tiempos de salida de los conductores y la velocidad de sus movimientos afectan la tasa de flujo y viceversa, haciendo imposible una descripción del tráfico mediante fórmulas matemáticas. Debido a esta complejidad, las simulaciones se mantienen como la única forma de estudiar una red de tráfico. Si bien el uso de simulaciones proporciona varias ventajas, como el estudio de configuraciones que serían difíciles o incluso imposibles de aplicar en la práctica, es posible querer utilizar el modelo de simulación fuera de los límites para los que fue construido causando una falsa apreciación del problema. Resulta prudente entonces entender con detalle la he-

ramienta de simulación para poder configurarla de tal forma que los resultados obtenidos imiten a los resultados que se obtendrían al realizar pruebas reales. Se presenta a continuación el simulador utilizado para las simulaciones.

4.3. SUMO: Simulation of Urban Mobility

SUMO (acrónimo de *Simulation of Urban MObility*) es un simulador de tráfico puramente microscópico, de código abierto y altamente portátil, diseñado para manejar grandes redes viales. Cada vehículo es manejado de forma explícita mediante un identificador, su hora de salida y una ruta propia dentro de la red (Krajzewicz et al., 2002).

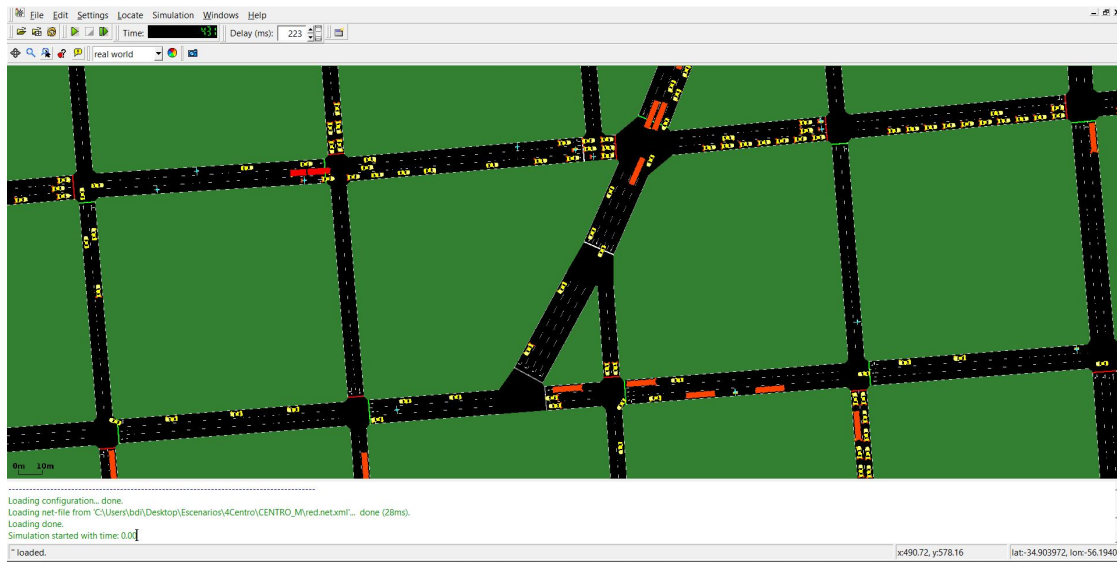


Figura 4.2: Captura de una simulación en SUMO-GUI.

En la Figura 4.2 se puede ver una captura de una simulación en SUMO-GUI, una interfaz gráfica provista dentro del paquete SUMO. En cada momento de la simulación los vehículos cuentan con determinada posición y velocidad. Estos parámetros son actualizados en cada ciclo de reloj (1 segundo). Se utiliza una extensión de modelo de seguimiento-de-vehículo desarrollado por Krauß (1998) donde, en cada ciclo de reloj, la velocidad del vehículo (v_f) es actualizada teniendo en cuenta la velocidad del vehículo que va por delante (v_l), dando lugar a un sistema libre de colisiones. A la velocidad actualizada se la denomina v_{safe} y se calcula como se indica en la Ecuación 4.1, donde $g(t)$ es la distancia al vehículo precedente en el tiempo t , τ el tiempo de reacción del conductor (generalmente 1 segundo), b la función de desaceleración y $\bar{v} = \frac{(v_l + v_f)}{2}$.

$$v_{\text{safe}}(t) = v_l(t) + \frac{g(t) - v_l(t)\tau}{\frac{\bar{v}}{b(\bar{v})} + \tau} \quad (4.1)$$

Otras de las características de SUMO incluyen: movimiento de vehículos en tiempo discreto y espacio continuo, semaforización de intersecciones, restricciones de velocidad, soporte para diferentes tipos de vehículos, multi-vía, calles con cambio de carril, alta velocidad de ejecución, alta interoperabilidad, entre otras (Krajzewicz et al., 2012). El simulador proporciona además información constante sobre el flujo de vehículos y permite simular factores medioambientales basados en el modelo *HandBook Emission Factors for Road Transport* (HBEFA) (Keller and Wüthrich, 2014). A través de este modelo es posible simular diferentes condiciones de vehículos con información sobre aceleraciones, deceleraciones, frenado, velocidades máximas, combustible consumido, así como su repercusión en las emisiones basadas en HBEFA: CO, CO₂, HC, PM_x y NO_x.

Dentro del paquete SUMO se tienen tres módulos principales:

- *SUMO*: lee la información de entrada, procesa la simulación, agrupa los resultados y produce los archivos de salida.
- *NETCONVERT*: herramienta que simplifica la creación de redes para SUMO a partir de redes provenientes de otros simuladores de tráfico como VISUM, Vissim o MATsim u otros formatos comunes como los mapas digitales de OpenStreetMap. También permite asignar, de forma automática, programas de ciclos en las intersecciones semaforizadas.
- *DUAROUTER*: es una aplicación de línea de comando que, dados el tiempo de partida, origen y destino, computa la ruta de un vehículo a través de la red mediante el algoritmo de *Dijkstra* para el cálculo de caminos más cortos.

4.4. Modelado de la red de tráfico

El modelado de la red de tráfico en estudio tiene como objetivo la obtención de un escenario SUMO que represente fielmente la realidad, de modo que el simulador devuelva resultados similares a los que se obtendrían si se hicieran pruebas reales. Como se señaló en la Sección 3.2.1, una red de tráfico cuenta con una serie de aspectos físicos y reglamentarios, como ser las calles, los carriles, sentidos, programas de ciclos, etc., y se tiene además el flujo vehicular propio de la zona en estudio. Debido a que el flujo vehicular depende de la red subyacente, se debe dividir el proceso de modelado de la red en dos etapas consecutivas (ver Figura 4.3).

A continuación se describe el modelado de los elementos físicos y reglamentarios.

4.4.1. Modelado físico y reglamentario

Una red SUMO describe las calles e intersecciones sobre las que circulan los vehículos durante la simulación. A gran escala, una red SUMO es un grafo dirigido. Los nodos, generalmente denominados *cruces* en el contexto de SUMO, representan las intersecciones y los arcos (unidireccionales) las calles. En concreto, una red SUMO contiene la siguiente información:

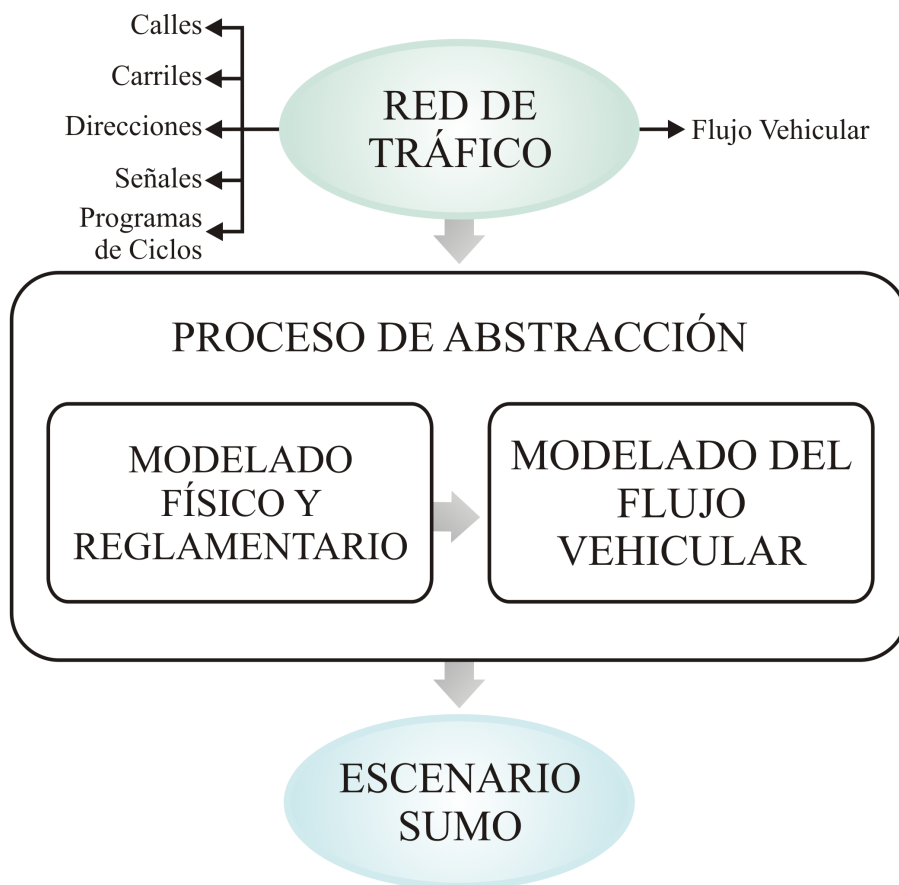


Figura 4.3: Proceso de abstracción de la red de tráfico.

- Calles (arcos) descritas como una colección de carriles, incluyendo la posición, forma y velocidad de cada carril.
- Programas de ciclos asociados a los cruces.
- Cruces, incluyendo regulación de derecho al paso.
- Conexiones entre carriles en cada cruce.

Dada una determinada red de tráfico, se busca construir la red SUMO correspondiente. Una red SUMO se codifica como un archivo XML y se obtiene usualmente con la herramienta NETCONVERT. Esta herramienta permite construir la red a partir de archivos descriptores más simples, o bien convirtiendo una red existente en otro formato.

El servicio de datos OpenStreetMap (2015) permite seleccionar un área específica de un mapa digital mundial y descargar los datos a un archivo XML (datos OSM). La herramienta *Java OpenStreetMap Editor* (JOSM) es un editor de datos OSM, que puede ser utilizado para refinar el mapa descargado y conservar exactamente los arcos pertenecientes a la red de tráfico en estudio. También es posible corregir el número de carriles por calle, sentidos, restricciones de giro, tipos de

calle (principal, secundaria, residencial, etc.), cuáles cruces se encuentran semaforizados, la velocidad máxima de cada vía de tránsito, etc. Muchas veces los datos que se descargan se encuentran desactualizados o son erróneos, por lo que se debe hacer un estudio presencial de la red de tráfico para luego realizar las correcciones correspondientes.

Una vez que se cuenta con el mapa OSM deseado, es posible convertirlo al formato de red SUMO mediante NETCONVERT. Para todos los cruces, las conexiones entre carriles son determinadas en forma automática y según las restricciones de giro indicadas. En aquellos cruces marcados como semaforizados, NETCONVERT asigna programas de ciclos por defecto. En la mayoría de los casos, las conexiones y los programas se corresponden con la realidad; sin embargo, en caso de intersecciones complejas, se deben corregir tanto las conexiones entre carriles como los programas de ciclos. En la Figura 4.4 se puede observar una intersección semaforizada con giro a la izquierda antes y después de aplicada la corrección de conexiones. Mientras que al convertir la red OSM se obtuvieron 6 cruces con 6 programas de ciclos independientes, se trata en verdad de un único cruce con un único programa de ciclo. Fue necesario unificar los cruces, corregir las conexiones y ajustar el programa de ciclo en forma manual (ver el Apéndice A para una descripción más detallada del uso de NETCONVERT y la aplicación de correcciones).

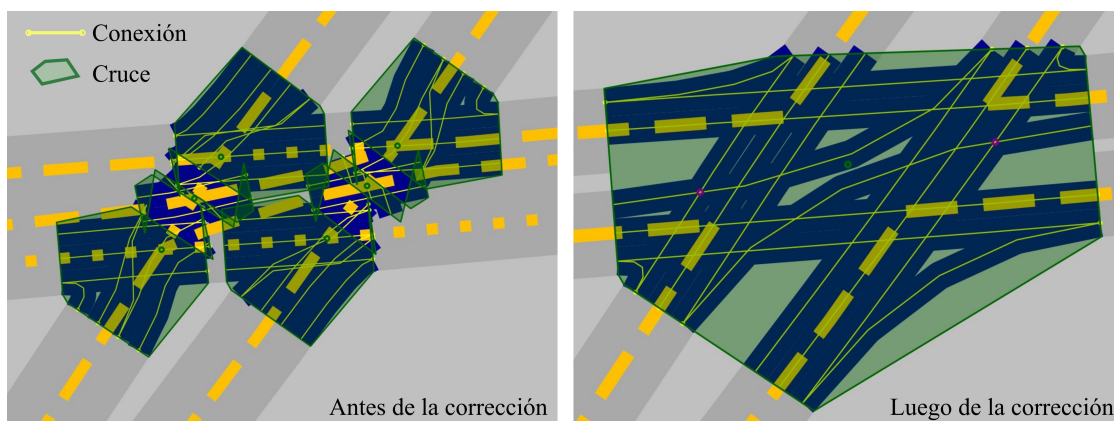


Figura 4.4: Corrección de conexiones y cruces en una intersección de una red SUMO.

En una red SUMO, un programa de ciclos se compone básicamente de:

- Un identificador, el cuál se utiliza para indicar qué cruces controla el programa correspondiente.
- Un conjunto de estados ordenados, compuestos de una cadena de caracteres y un valor que indica la duración de cada luz. A diferencia de lo expuesto en la Sección 3.2.1, se tiene un carácter por cada conexión entre carriles y los valores son “G”, “r” y “y” para las luces verde, roja y amarilla respectivamente (ver Figura 4.5).

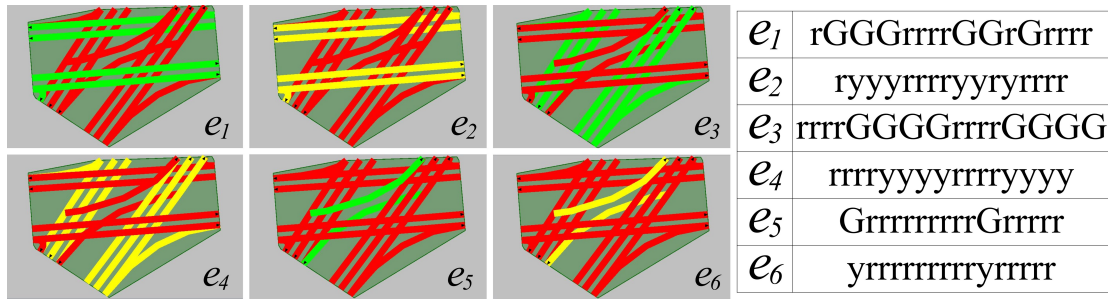


Figura 4.5: Conexiones y estados del programa de ciclos en una intersección de una red SUMO.

- Un tiempo de *offset* o tiempo de inicio del programa, que permite un comienzo retardado de la primera fase respecto al estado inicial del programa de una intersección.

Determinar los estados de un programa es en la mayoría de los casos una tarea trivial. Sin embargo, cuando se tiene una gran cantidad de cruces bastante distanciados unos de otros, no es posible observar todos los cambios de luces en forma simultánea y determinar los estados no resulta sencillo. En estos casos, el estudio del programa se puede realizar mediante una filmación que enfoque al mismo tiempo todos los semáforos, permitiendo para un momento determinado, determinar las luces de todos los semáforos que componen el programa. La Figura 4.6 muestra una intersección perteneciente a una de las redes de tráfico analizadas, la cual se compone de 7 cruces con un único programa de ciclos. Para su estudio se realizó una filmación aérea (mediante un dron) durante un período de tiempo suficiente para determinar los estados del programa y su respectiva duración. Se determinaron un total de 14 estados.

Una vez se tiene la red SUMO que modela fielmente los aspectos físicos y reglamentarios de la red de tráfico es posible continuar con el modelado del flujo vehicular.

4.4.2. Modelado del flujo vehicular

Si bien el flujo vehicular se encuentra en constante cambio y nunca es exactamente el mismo, las variables macroscópicas estudiadas (tasa, velocidad y densidad) se mantienen aproximadamente constantes durante períodos de tiempo específico. Resulta razonable entonces asociar las matrices OD dadas en la formulación matemática planteada en la Sección 3.5 a una determinada franja horaria, como por ej. de lunes a jueves, de 17:00hs a 19:00hs.

Según se presentó en la Sección 3.2.3, la variable macroscópica tasa de flujo queda determinada por la matriz OD en cada arco de la red. Además, la velocidad de los vehículos es controlada por SUMO siguiendo los delineamientos de la reglamentación vial. Se puede deducir entonces que, de acuerdo a la ecuación de continuidad ($q = ku$), al estimar una matriz OD y traducirla a tráfico microscópico

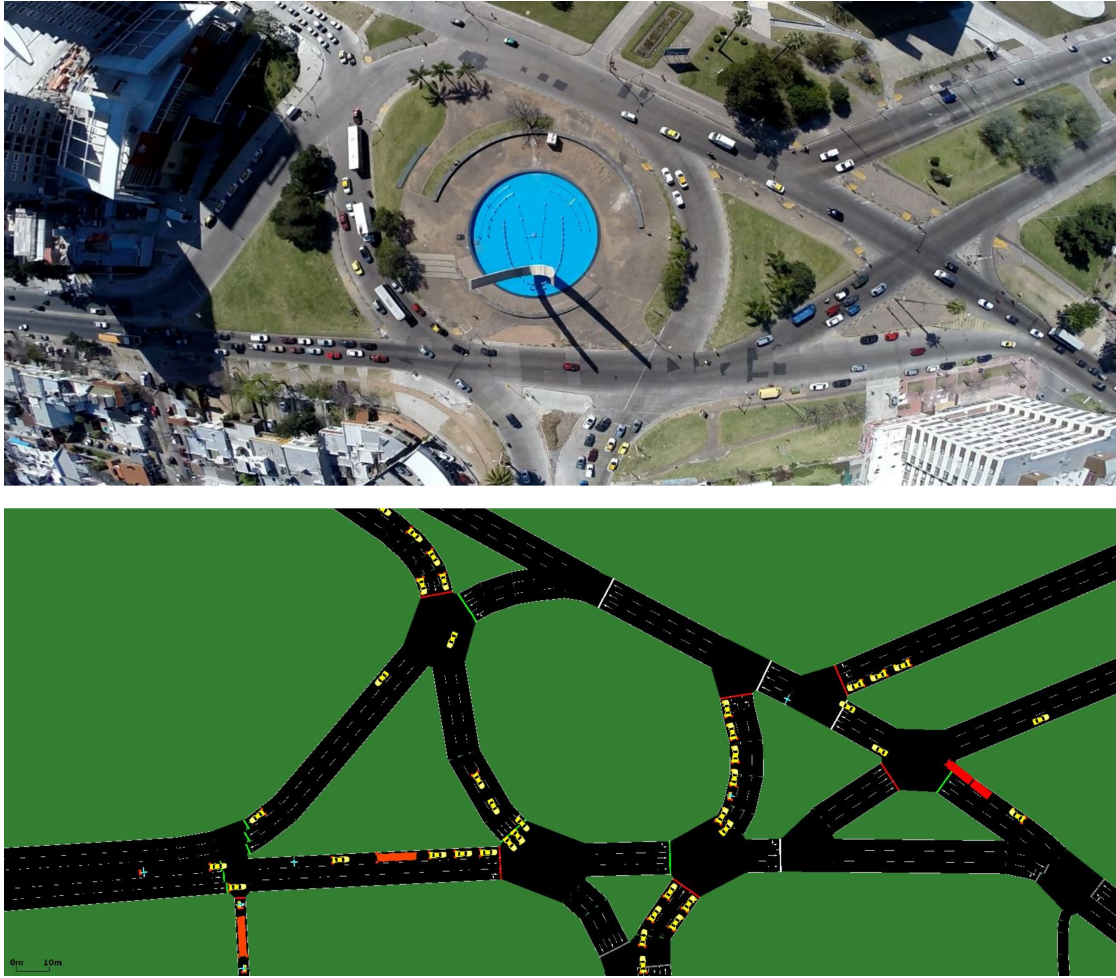


Figura 4.6: Imagen aérea de una intersección compleja estudiada y la correspondiente red SUMO durante una simulación.

de SUMO, se obtiene una representación macroscópica completa del flujo vehicular en la red de tráfico. Es decir, el tráfico microscópico de SUMO mantiene las características macroscópicas, tasa de flujo, velocidad y densidad, de la red de tráfico en estudio.

El proceso de traducción de una matriz OD a tráfico en una red se conoce como *asignación de tráfico* (Barceló, 2010). En el presente trabajo, las instancias de evaluación analizadas (ver Sección 5.3) se corresponden con áreas relativamente pequeñas. Por este motivo fue posible suponer, en todos los casos, que la ruta para cada par OD $w = (i, j)$ esta dada por el camino más corto posible entre el nodo de entrada n_i y el nodo de salida n_j . Dado un elemento $M[i, j]$, se generan entonces $M[i, j]$ vehículos de n_i a n_j siguiendo la ruta más corta y con tiempos de salida distribuidos de acuerdo a una distribución uniforme durante el tiempo de simulación t_{SIM} . Para el cálculo de rutas se utilizó la herramienta DUAROUTER sobre la red SUMO determinada en la etapa anterior.

En SUMO, un vehículo se especifica básicamente con:

- Tiempo de partida.
- Tipo de vehículo.
- Ruta de viaje: lista ordenada de arcos.

En la próxima sección se describe la metodología utilizada para la obtención de las matrices OD y la posterior traducción a tráfico SUMO.

4.5. Generación de tráfico SUMO mediante integración GPS/GIS

Los últimos años han visto un aumento dramático en la presencia de vehículos con dispositivos de posicionamiento global (GPS, por sus siglas en inglés). Además de proporcionar servicios de localización y navegación, estos dispositivos abren una ventana sin precedentes hacia la dinámica y movilidad de una red vial. Desde este punto de vista, el uso de datos basados en trazas de GPS (secuencia ordenada de puntos con latitud, longitud, fecha y hora, entre otros parámetros), emerge como una forma confiable y económica de recabar tiempos de viaje y velocidades en una red vial. En particular, taxis equipados con GPS se pueden considerar como sensores móviles ubicuos que constantemente sondean el ritmo y pulso de una ciudad. Estos se han utilizado para la construcción automática de mapas, detección de puntos calientes, computación urbana, y estrategias de encuentro de pasajeros, entre otros, demostrando ser una fuente de datos extremadamente útil para descubrir el comportamiento subyacente del tráfico (Castro et al., 2012).

En la Sección 3.3 se analizó un método manual de estudio del flujo vehicular para una red de tráfico. Este método permite estimar una matriz OD utilizando conteos de tráfico, no sin antes realizar un ordenamiento de los arcos de la red (ver Sección 3.3.2). De acuerdo a la Ecuación 3.8, esto es posible siempre que se cuente con datos previos del volumen de tráfico. Sin embargo, debido a limitaciones de recursos, infraestructura sensorial o restricciones de distribución, es siempre un desafío contar con este tipo de datos. Como veremos en el próximo capítulo, los datos que fueron recabados para el análisis del flujo vehicular, se componen de una gran cantidad de trazas de GPS las cuales cubren las zonas estudiadas. Por consiguiente, se procedió a desarrollar una estrategia que permitiera, a partir de las trazas de GPS, obtener valores estimativos del volumen de flujo en los arcos de la red y de este modo hacer uso del método analizado. En la Figura 4.7 es posible ver un esquema de la estrategia desarrollada.

En primer lugar, se analizan las trazas de GPS para obtener valores estimativos del volumen de tráfico en la red. Luego, siguiendo la propuesta de la Sección 3.3, se priorizan los arcos, se realizan conteos manuales y se estima la matriz OD. Finalmente, se genera tráfico para SUMO. Para llevar a cabo estas tareas, fue necesaria la implementación de varios componentes de software que permitieron automatizar gran parte del procedimiento:

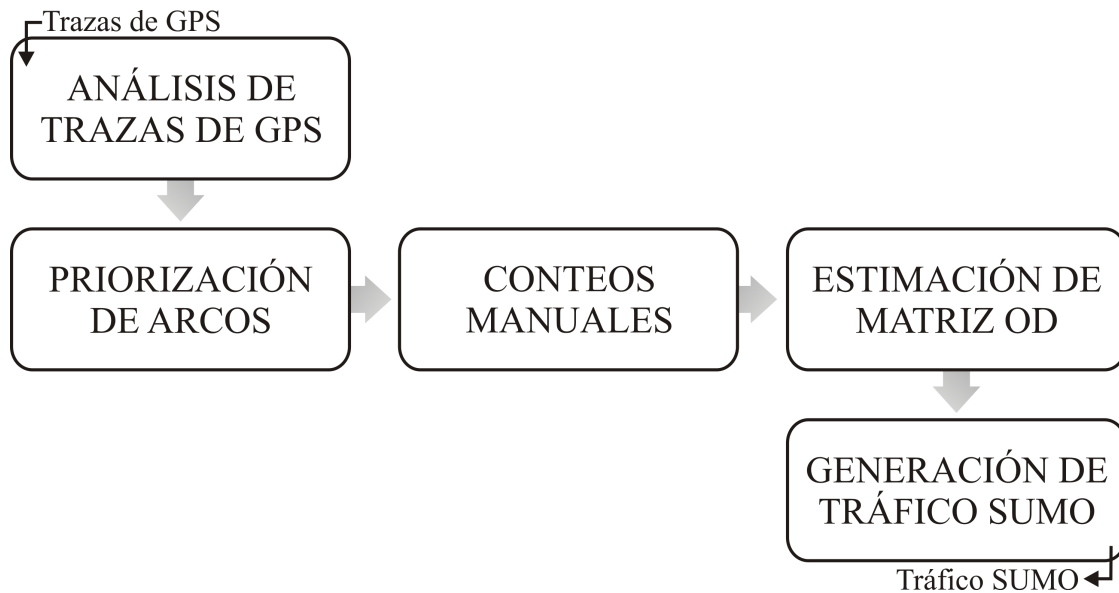


Figura 4.7: Esquema de la estrategia desarrollada para la generación de tráfico SUMO a partir de trazas de GPS

- **PowerGen:** Este programa fue implementado para llevar a cabo la primer etapa de la estrategia desarrollada. Analiza las trazas de GPS y devuelve un valor estimativo del volumen de tráfico para un conjunto de arcos de la red. A este valor le llamaremos *ranking*.
- **PriorWeight:** Prioriza los arcos de la red lo que permite decidir donde realizar los conteos según los recursos disponibles.
- **TrafficGenerator:** Devuelve una matriz OD para cada tipo de vehículo considerado y genera el tráfico SUMO correspondiente.

Estos componentes se describen más a fondo en las siguientes secciones. En el Apéndice A se realiza una descripción detallada de como ejecutar las herramientas.

4.5.1. PowerGen

El programa PowerGen, implementado en .NET Framework 4.0, determina un valor $R(l)$ estimativo del volumen de tráfico para cada arco $l \in A$ de una red, analizando trazas de GPS. En la Figura 4.8 puede verse un diagrama de componentes en el que se observa la base de datos de trazas de GPS (MySQL), la herramienta OSM2Routing y los sub-componentes de PowerGen.

Dada una franja horaria F (hora inicial y hora final), el componente GPXGenerator filtra todas las trazas de GPS de la base de datos, conservando los puntos del recorrido que tienen lugar durante la franja. Por ejemplo, si una de las trazas contiene un recorrido que comienza a las 4pm y finaliza a las 8pm y la franja horaria es de 5pm a 7pm, solo se conservan los puntos entre las 5 pm y las 7 pm, descartándose todos los demás. Las trazas filtradas son luego traducidas al

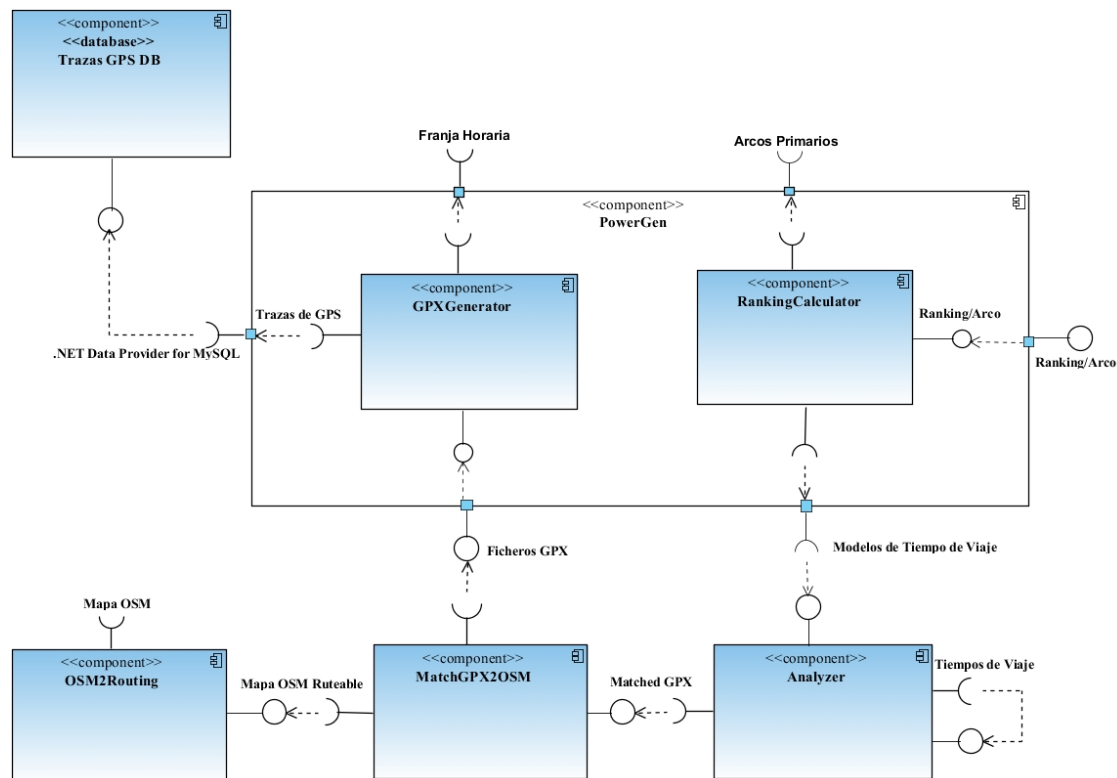


Figura 4.8: Diagrama de componentes para el programa PowerGen.

formato GPX (esquema XML pensado para transferir datos GPS entre aplicaciones), y analizadas utilizando las herramientas del proyecto *Travel Time Analysis* (TTA) (Kabrt, 2010). Este proyecto ofrece varias opciones para procesar y analizar archivos GPX. A continuación, se explica el orden de ejecución de las herramientas.

- **OSM2Routing**: esta herramienta se utiliza para convertir un mapa OSM de la red de tráfico a un formato amigable (*ruteable*), extrayendo los tipos de rutas (autopistas, residenciales, etc) y otros elementos irrelevantes. En la Figura 4.9 se puede ver un mapa OSM y su correspondiente ruteable obtenido al aplicar la conversión.
- **MatchGPX2OSM**: se utiliza para ajustar los archivos GPX al mapa ruteable obtenido en el paso anterior. A este proceso se le conoce como *map-matching* y consiste en alinear la secuencia de posiciones a la red de rutas del mapa. Este es un paso de procesamiento previo fundamental para muchas aplicaciones, en particular para el análisis del flujo vehicular. En la práctica existen una gran cantidad de trayectorias GPS con baja tasa de muestreo (por ejemplo, un punto por cada 2-5 minutos). La mayoría de los enfoques map-matching actuales solo se ocupan de trayectorias con alta tasa de muestreo (por lo general de un punto cada 10-30s) y se vuelven menos efectivos para tasas bajas a medida que incrementa la incertidumbre en los datos. MatchGPX2OSM utiliza un algoritmo de map-matching llamado ST-Matching para tasas de

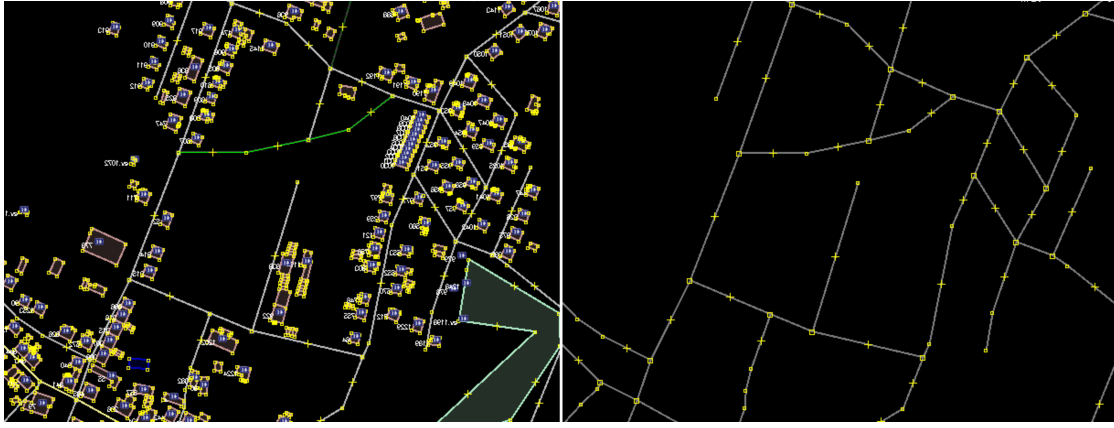


Figura 4.9: Ejemplo de mapa OSM y su correspondiente ruteable.

muestreo baja. ST-Matching considera la geometría espacial y las estructuras topológicas de la red de calles así como las restricciones de velocidad de las trayectorias. Basado en un análisis espacio-temporal, construye un grafo candidato del cuál se identifica la secuencia que se ajusta mejor (Lou et al., 2009). La Figura 4.10 muestra el resultado (color verde) de ajustar una traza (color rojo) a un mapa ruteable utilizando MatchGPX2OSM.

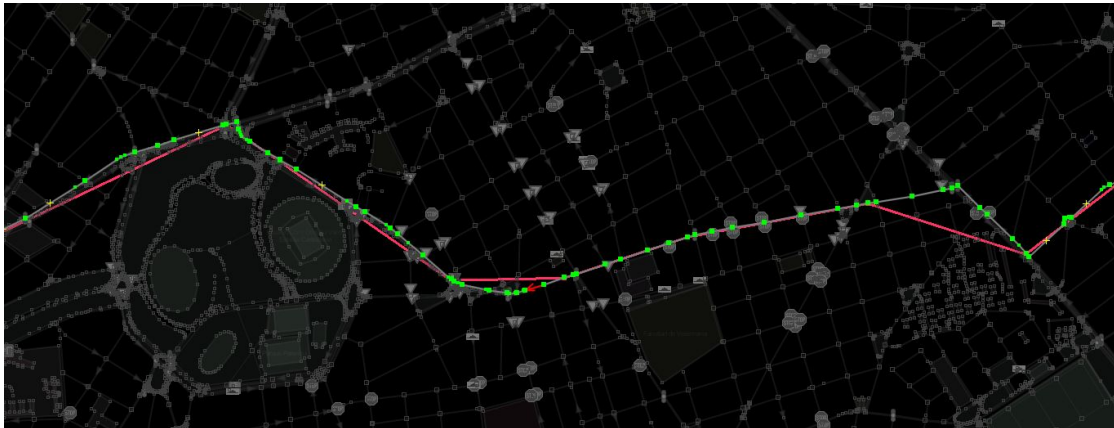


Figura 4.10: Ejemplo de ajuste de una traza de GPS a un mapa ruteable utilizando la herramienta MatchGPX2OSM.

- *Analyzer*: a partir del conjunto de trazas filtradas ajustadas genera primero una base de datos de tiempos de viaje (TTDB) y luego modelos de tiempo de viaje $TT(l)$ para cada arco *suficientemente cubierto* por un número mínimo (configurable) de trazas GPX.

Finalmente, el componente *RankingCalculator*, calcula el valor ranking $R(l)$ para los arcos de la red suficientemente cubiertos de la siguiente manera:

$$R(l) = \frac{d(l) \times c(l)}{d(l)/TT(l)} = c(l) \times TT(l) \quad (4.2)$$

donde $c(l)$ es el número de carriles y $d(l)$ la distancia en metros del arco l . Este cálculo fue determinado a priori estudiando las relaciones entre tasa de flujo, densidad y velocidad de la Sección 3.2.2. Aquellos arcos que cuenten con un *área grande donde la velocidad promedio sea baja* tendrán un ranking alto. A los arcos que no estén suficientemente cubiertos por las trazas se les asigna un ranking de 0.

4.5.2. PriorWeight

Dado el conjunto de arcos A , junto a su respectivo ranking, el programa PriorWeight asigna a cada arco el peso $P(l) = R(l)/OD(l)$ (que se obtiene al sustituir $V(l)$ por $R(l)$ en la Ecuación 3.8) y luego, determina el conjunto mínimo de arcos $A_M \subseteq A$ que separan todos los pares OD resolviendo el problema de minimización Z_2 de la Sección 3.3.2, Ecuación 3.7. Ver diagrama de componentes en la Figura 4.11.

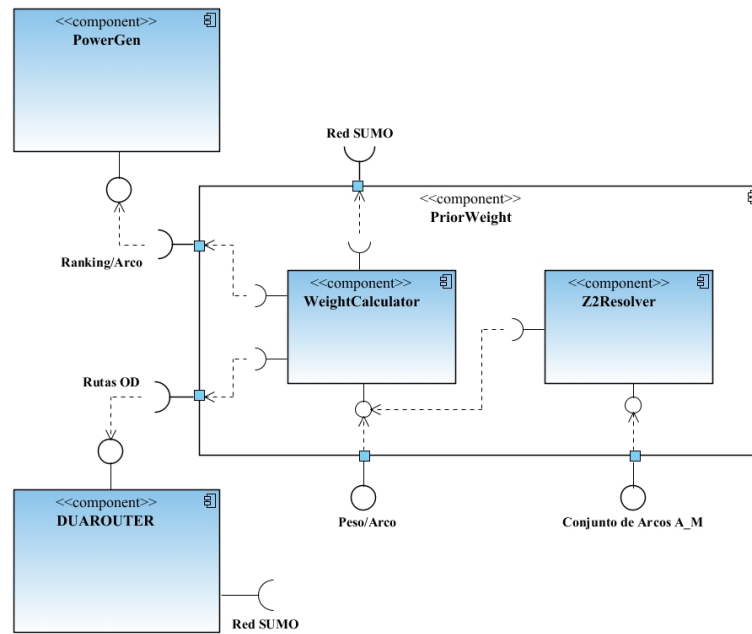


Figura 4.11: Diagrama de componentes para PriorWeight.

PriorWeight fue implementado en .NET Framework 4.0. Para resolver el problema de minimización se utiliza el algoritmo de búsqueda local *hill climbing* o *ascenso de colina* provisto por la librería Microsoft Solver Foundation 3.1. Para el cálculo de la ruta más corta posible entre cada par OD se utiliza la herramienta DUAROUTER sobre la red SUMO.

4.5.3. TrafficGenerator

Dados los conteos para el conjunto de arcos $A_C \subseteq A$, TrafficGenerator estima una matriz OD M por cada tipo de vehículo considerado durante los conteos. A partir de las matrices genera el tráfico SUMO correspondiente con tiempos de

salida distribuidos en forma uniforme y utilizando la ruta más corta posible entre el origen y destino (ver Figura 4.12).

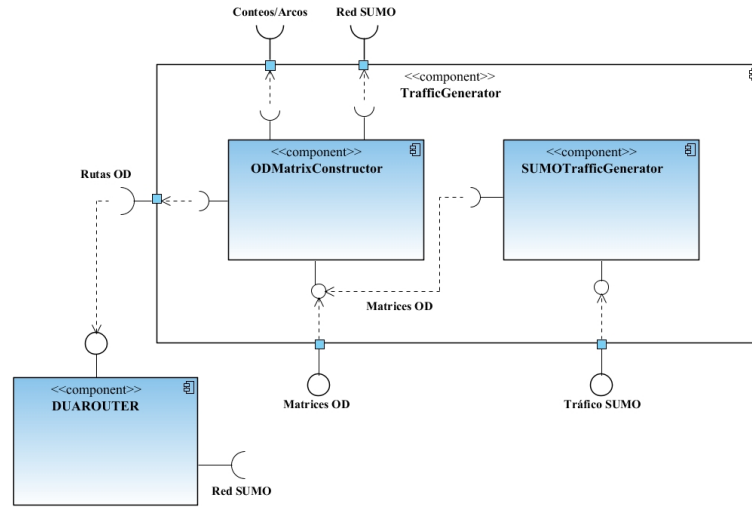


Figura 4.12: Diagrama de componentes para TrafficGenerator.

Para la obtención de las matrices, se resuelve una versión multiobjetivo del problema de minimización propuesto en la Sección 3.3.2, Ecuación 3.9 (para cada tipo de vehículo), utilizando el AEMO NSGA-II. El algoritmo es ejecutado hasta obtener una diferencia despreciable entre los conteos y el volumen. El uso de un algoritmo evolutivo para este procedimiento es sugerido en Saraswathy and Isaac (2013). TrafficGenerator fue implementado en Java.

4.6. Diseño e implementación de los AEMO propuestos

En conformidad con el esquema presentado en la Sección 4.2, en la Figura 4.13 puede verse un diagrama con los componentes principales en torno a la estrategia de optimización implementada. A continuación se describe el entorno de programación.

4.6.1. Entorno de programación

Para la implementación de los AEMO se utilizaron las funcionalidades provistas por el *MOEA framework* (Hadka, 2015), una biblioteca de código abierto implementada en Java, utilizada para el desarrollo y experimentación con AEMO y otros algoritmos de optimización de propósito general. El MOEA framework proporciona las herramientas necesarias para el rápido diseño, desarrollo y ejecución de un gran número de algoritmos, incluyendo los utilizados en este trabajo: NSGA-II, GDE3, MOEA/D y ϵ -NSGA-II.

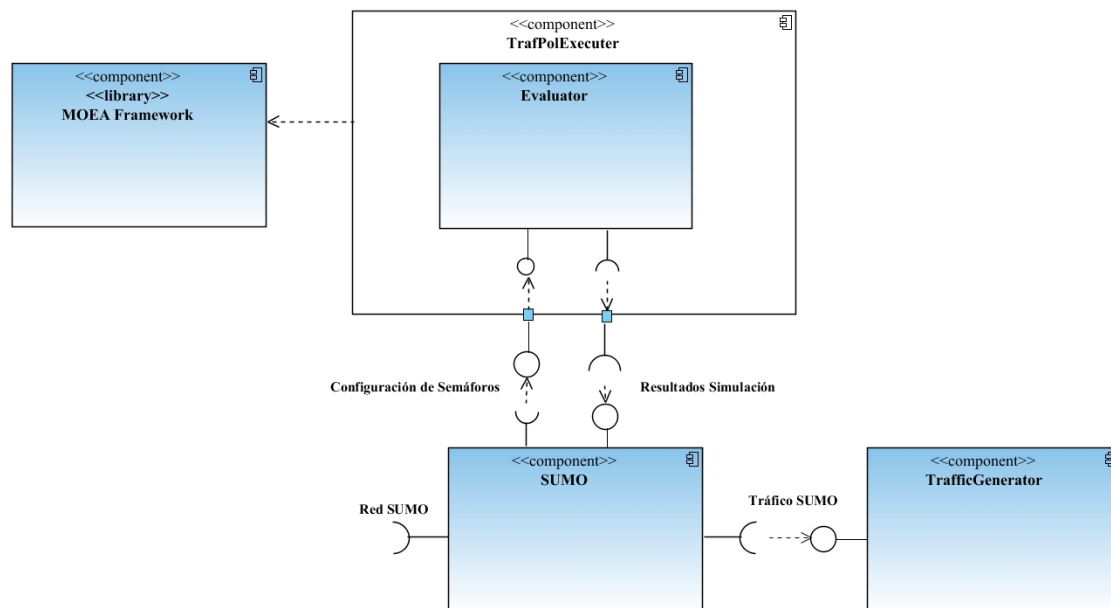


Figura 4.13: Diagrama de los componentes principales en la resolución del problema de planificación de ciclos de semáforos.

La clase *Executor* perteneciente al framework, es la responsable de construir y ejecutar varias veces un algoritmo. Una ejecución requiere de tres piezas de información: 1) un problema; 2) el nombre del algoritmo elegido; y 3) el número máximo de evaluaciones (cálculo de fitness) como condición de parada del algoritmo. Opcionalmente se pueden especificar propiedades de parametrización propias del algoritmo, de lo contrario se utilizan valores por defecto. Por ejemplo, en el caso del NSGA-II, se puede especificar el tamaño de la población, la probabilidad del cruzamiento, probabilidad de mutación, etc.

Para definir un problema, se debe implementar la interfaz *Problem* la cuál define todos los métodos que caracterizan un problema como ser la representación, evaluación de soluciones, etc. En nuestro caso, sin embargo, fue más conveniente extender la clase *AbstractProblem* que provee implementaciones por defecto de muchos de los métodos requeridos por la interfaz *Problem*. Para ello se implementó la clase *GenotipoEvaluator* y se sobrescribieron los métodos de creación y evaluación de nuevos individuos. La función de creación genera soluciones aleatorias, mientras que la evaluación de un individuo requiere ejecutar el simulador SUMO utilizando los programas de ciclos codificados en el individuo. Al terminar la simulación se leen las salidas del simulador y se determinan los valores de las funciones objetivo.

El *Executor* provee también otras funcionalidades avanzadas, como ser la distribución de la evaluación de las funciones objetivo entre varios núcleos de procesamiento (ya sea en una o en múltiples computadoras en red). Como se mencionó en la Sección 2.5, este método se corresponde con el modelo de paralelización maestro-esclavo. Sin embargo, solo algunos de los algoritmos provistos soportan este tipo de paralelización (los llamados *algoritmos nativos*). Otro tipo de paralelización pro-

vista por el framework es el modelo de islas. También es posible aplicar técnicas de hibridación.

A continuación se describen los operadores evolutivos y las técnicas de paralelización aplicadas para la ejecución de los AEMO utilizados.

4.6.2. Operadores evolutivos

Los operadores utilizados para el NSGA-II fueron:

- *Cruzamiento Binario Simulado*: o SBX por sus siglas en inglés, fue propuesto por Deb and Agrawal (1994) con el objetivo de simular el comportamiento del operador de cruce de 1-punto en codificaciones binarias para codificaciones reales. Para ello, primero se definió el *poder de búsqueda* de un operador de cruzamiento en términos de la probabilidad de distribución de que dos cadenas hijas arbitrarias sea creadas por dos cadenas padre cualesquiera. Luego se desarrolló el operador SBX con un poder de búsqueda similar al operador de cruzamiento de 1-punto.
- *Mutación Polinomial*: propuesto por Deb and Goyal (1996), dado un individuo padre, genera una solución hija cercana, con una probabilidad más alta que una solución lejana, de la siguiente manera:

$$x^{t+1} = x^t + (x^{(U)} - x^{(L)}) \cdot \delta \quad (4.3)$$

donde $\delta = (2 \cdot u)^{1/(\eta+1)} - 1$ si $u < 0,5$ ó $\delta = 1 - [2 \cdot (1 - u)]^{1/(\eta+1)}$ en otro caso y $u \sim U[0, 1]$.

Para el GDE3, se utiliza el proceso de evolución diferencial (mutación y cruzamiento) visto en la Sección 2.3.4. Para el MOEA/D, se utiliza el cruzamiento visto en la Sección 2.3.4 y mutación polinomial. En el caso del ϵ -NSGA-II, al igual que en el NSGA-II, se utiliza el operador de cruzamiento SBX y mutación polinomial.

4.6.3. Técnicas de paralelización

Debido a que el NSGA-II, GDE3 y el ϵ -NSGA-II entran dentro de la categoría algoritmos nativos del framework fue posible aplicarles la paralelización maestro-esclavo. Las evaluaciones de individuos son distribuidas entre todos los núcleos disponibles de la computadora utilizada para la ejecución del algoritmo. Esto significa que el simulador SUMO se ejecuta en forma paralela durante la evaluación de los individuos.

Debido a que el MOEA/D no es un algoritmo nativo, no fue posible aplicar la paralelización maestro-esclavo provista por el framework. Por este motivo, se decidió distribuir las ejecuciones independientes del algoritmo entre todos los núcleos de procesamiento, aplicando un modelo de islas sin migraciones.

4.6.4. Representación de soluciones

Considérese un conjunto $P = \{p_1, p_1, \dots, p_n\}$ de programas de ciclos de semáforos, cada uno correspondiente a un conjunto de nodos N_{p_i} de la red en estudio. Los semáforos gobernados por este programa están sincronizados y no permiten combinaciones inválidas. Cada programa p_i consta de una serie de estados, con determinada duración, que se repiten en forma secuencial indefinidamente. Cada estado es una combinación válida de colores, que debe seguir reglas específicas establecidas. Un ejemplo de programa se presenta en la Figura 4.14. La representación utilizada es una cadena de naturales con las duraciones de los estados de los programas dispuestos en forma sucesiva, junto al tiempo de *offset* o tiempo de inicio o_i del programa p_i :

$$[o_1, d_{11}, d_{12}, \dots, d_{1k_1}], [o_2, d_{21}, d_{22}, \dots, d_{2k_2}], \dots, [o_n, d_{n1}, d_{n2}, \dots, d_{nk_n}] \quad (4.4)$$

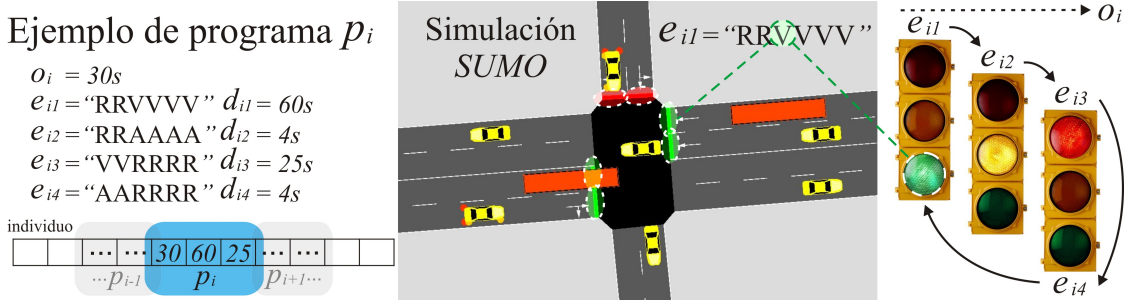


Figura 4.14: Ejemplo de un programa de ciclos p_i , representación y simulación SUMO en estado e_{i1} .

Los estados con luces amarillas tienen una duración constante de 4 segundos, por lo que no se consideran en la representación. Los estados con luces verdes o rojas tienen duración $d_{ij} \in [5, 60] \subset N$, en segundos. A los offsets se les asigna un valor dentro del intervalo $[0, 60k_i + 5k_i] \subset N$, donde $60k_i + 5k_i$ se corresponde con la duración total del programa. Esto permite que los programas inicien en cualquier momento del ciclo.

4.6.5. Funciones objetivo

Se propone minimizar la emisión total de contaminantes vehiculares (CO , CO_2 , HC , PM_x y NO_x), maximizar el número de vehículos que llega a destino en tiempo de simulación y minimizar el tiempo global perdido por todos los vehículos como consecuencia de ir por debajo de la velocidad ideal, según el modelo matemático definido en la Sección 3.5.1. Se tienen en total 7 funciones objetivo.

Capítulo 5

Análisis experimental

El objetivo de este capítulo consiste en describir y analizar las pruebas realizadas durante el desarrollo del proyecto y los resultados obtenidos, explicando en cada caso los métodos empleados y las conclusiones deducidas.

5.1. Introducción

A lo largo de este capítulo se presentan y analizan los resultados obtenidos en los experimentos de validación, configuración y evaluación de los AEMO NSGA-II, GDE3, MOEA/D y ϵ -NSGA-II, propuestos sobre un conjunto de instancias de pruebas pertenecientes a la ciudad de Montevideo, Uruguay.

En la siguiente sección se describe la plataforma de desarrollo y ejecución de los AEMO desarrollados. La Sección 5.3 describe las instancias de evaluación seleccionadas sobre la ciudad de Montevideo y la información vehicular recabada con el propósito de modelar el tráfico de las distintas instancias. Se evalúa además la calidad de $R(l)$ (ver Sección 4.5) como estimador del volumen de tráfico. En la Sección 5.4 se describe la calibración paramétrica llevada a cabo para los algoritmos. Finalmente, en la Sección 5.5, 5.6 y 5.7 se presentan y analizan los resultados numéricos de los algoritmos ejecutados.

5.2. Plataforma de desarrollo y ejecución

Como se mencionó en el capítulo anterior, los algoritmos ejecutados fueron desarrollados en Java 1.6 utilizando la biblioteca MOEA Framework 2.4. La fase de simulación se realizó utilizando SUMO versión 0.21.0 para el sistema operativo CentOS 6. El análisis experimental fue realizado en un HP Proliant DL385 G7 con procesador AMD Opteron 6172 2.10 GHz, 24 núcleos, 24GB RAM y CentOS Linux 6.5, del Cluster FING, la infraestructura de cómputo de alto desempeño de la Universidad de la República (Nesmachnow, 2010).

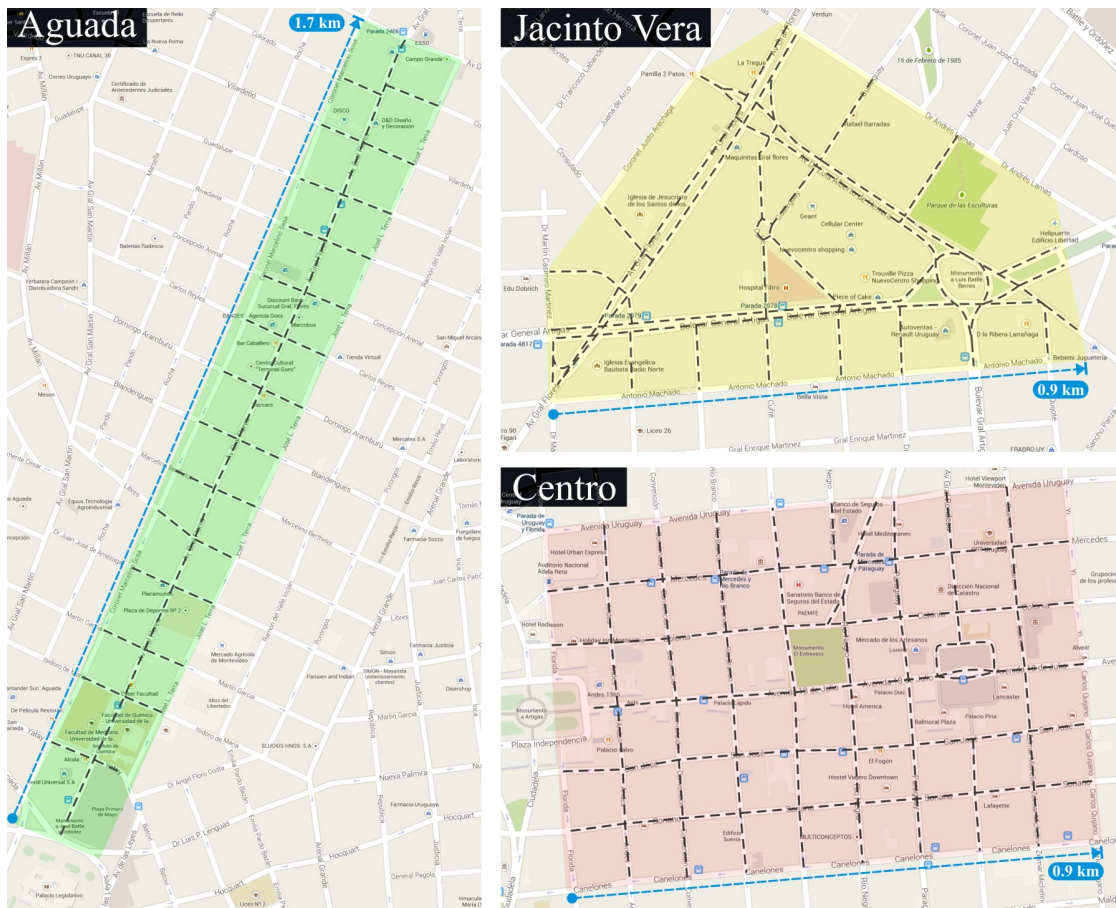


Figura 5.1: Zonas de la ciudad de Montevideo elegidas para la evaluación de los algoritmos.

5.3. Instancias de evaluación

Para la realización de este trabajo se eligieron tres grandes zonas de la ciudad de Montevideo con distintivas características de tramado y tráfico vehicular. Los escenarios comprenden parte de los barrios Aguada, Jacinto Vera y Centro respectivamente (ver Figura 5.1):

- **Aguada:** escenario de trama simple compuesto por una única avenida, abarcando un total de 14 intersecciones semaforizadas.
- **Jacinto Vera:** escenario de trama compleja compuesto por tres grandes intersecciones semaforizadas.
- **Centro:** se trata del escenario más grande y complejo compuesto por avenidas y calles residenciales de distinta longitud. Posee un total de 44 intersecciones de las cuales 41 están semaforizadas. La gran concentración de servicios, oficinas públicas y privadas, comercios, cines y teatros, lo convierten en una de las zonas más populosas de Montevideo.

Se utilizaron las reglas del tráfico regulado, señales, sentidos, carriles, etc. El tiempo de cada simulación en SUMO se estableció en 8 minutos, determinado a priori suficiente para que un vehículo pueda circular desde cualquier intersección del mapa a cualquier otra a una velocidad promedio de 45 km/h. Para cada escenario se consideraron además tres franjas horarias: *mañana* (8:00-10:00 hs), *tarde* (17:00-19:00 hs) y *noche* (20:00-22:00 hs), dando lugar a un total de 9 instancias.

5.3.1. Estudio del tráfico

Se consideró el estudio de las zonas seleccionadas y dichos horarios para poder observar el comportamiento de los algoritmos frente a distintos parámetros de tráfico vehicular. En primer lugar, se procedió a recabar información relacionada al tráfico montevideano que permitiera modelar las situaciones de tráfico en los distintos horarios de la forma más precisa posible. A través de la información suministrada por la Intendencia de Montevideo, en conformidad con la *ley de acceso a la información pública N° 18.381*, se relevaron los siguientes datos:

- **Trazas de GPS CUTCSA:** se trata de la posición global cada 30 segundos aproximadamente de ómnibus de la *Compañía Uruguay de Transportes Colectivos S.A.* (CUTCSA), actualmente la empresa de transporte más grande de Uruguay. Se conformó un grupo de 37 líneas con recorridos en las zonas seleccionadas para la generación de escenarios y se solicitaron las trazas de todo un mes en el que no haya ocurrido ningún evento que pudiera influir en los datos, desviándolos de la tendencia habitual. Se obtuvieron 59.520 recorridos correspondientes al mes de septiembre de 2014, un total de 11.857.521 posiciones de GPS. En la Figura 5.2 se pueden ver las trazas de GPS CUTCSA dispuestas sobre las zonas seleccionadas. Es importante observar que no logran cubrir todas los segmentos de calle en estudio dado que estos no forman parte de los recorridos de las líneas solicitadas.
- **Trazas de GPS Flotilla:** se trata de las trazas de GPS de un servicio de reparto privado que opera en la ciudad de Montevideo (ver Figura 5.3).

Siguiendo la estrategia de análisis de flujo vehicular descrita en la Sección 4.5, una vez clasificados los arcos de la red, se realizaron conteos manuales en los de mayor prioridad. Para cada escenario se realizaron aproximadamente 30 conteos en cada horario. Con el objetivo de determinar con mayor exactitud la emisión de contaminantes, los vehículos se clasificaron en *light* (vehículo liviano), *pcar* (auto de pasajeros promedio), *heavy* (camión pesado) y *bus* (ómnibus).

5.3.2. Validación del valor Ranking

En la Sección 4.5 se utilizó el valor $R(l)$ como estimativo del volumen de tráfico para cada arco $l \in A$ de una red de tráfico. A continuación, evaluamos la calidad de

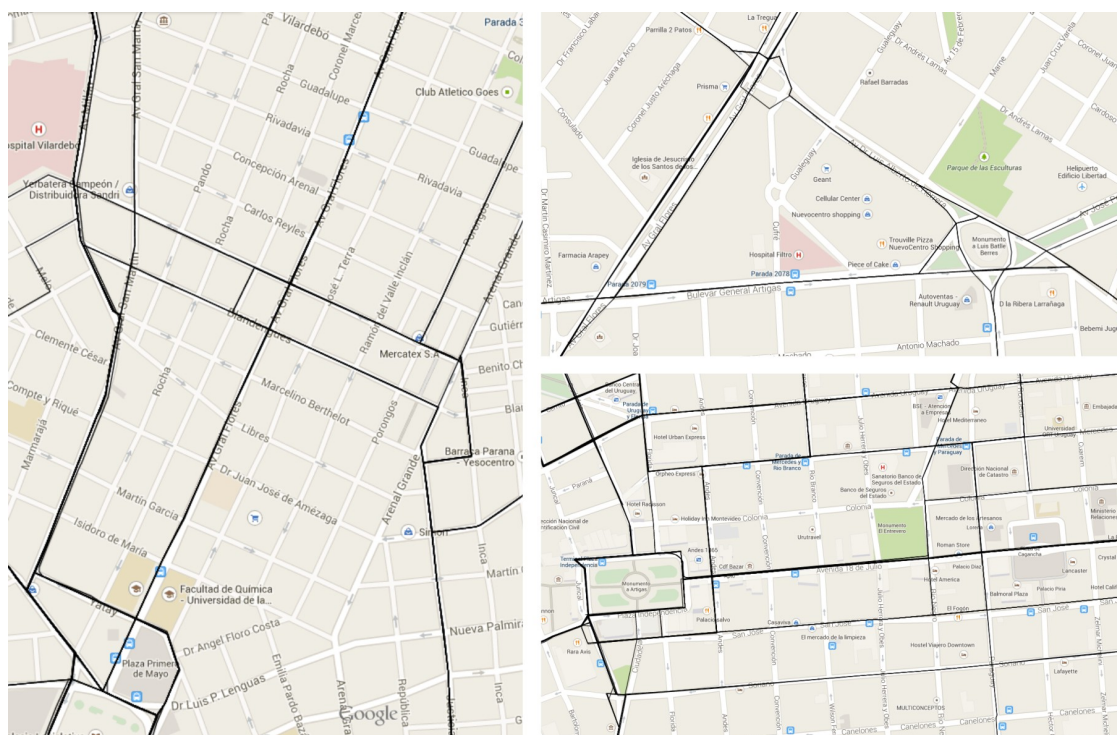


Figura 5.2: Trazas de GPS de la compañía CUTCSA dispuestas sobras las zonas de evaluación.

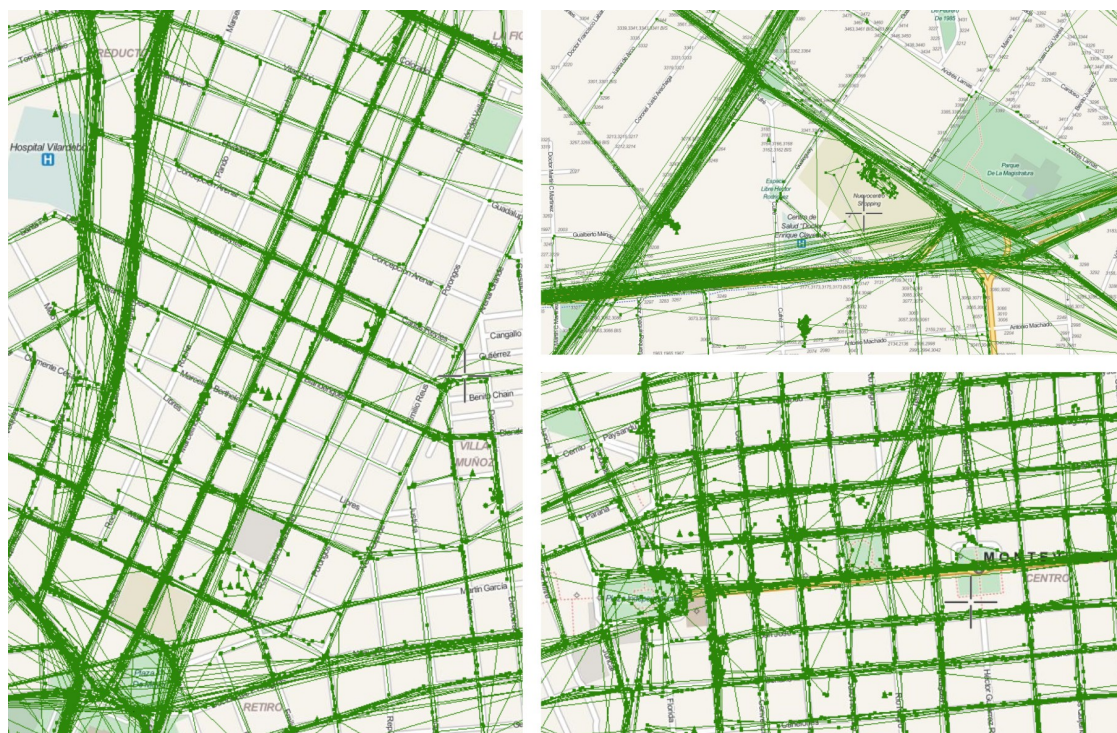


Figura 5.3: Trazas de GPS Flotilla dispuestas sobras las zonas de evaluación.

escenario	horario	α	β_1	β_2	β_3	β_4	RMSE	PA
Aguada	mañana	0.8	0.1	0.6	2.6	2.6	26.1	61.9 %
	tarde	0.8	0.1	0.6	2.6	2.6	13.3	77.3 %
	noche	0.2	0.1	0.6	2.6	2.6	10.8	60.2 %
Jacinto Vera	mañana	1.0	0.1	0.6	2.6	2.6	26.1	76.5 %
	tarde	1.1	0.1	0.6	2.6	2.6	61.0	72.5 %
	noche	1.0	0.1	0.6	2.6	2.6	30.5	71.9 %
Centro	mañana	1.9	0.1	1.2	3.4	2.6	10.2	85.9 %
	tarde	0.9	0.1	0.6	9.9	2.6	23.7	73.0 %
	noche	0.5	0.5	0.6	7.5	2.6	8.1	70.9 %

Cuadro 5.1: Resultados de validación para el estimador de volumen vehicular ranking.

$R(l)$ como estimador mediante los indicadores *raíz cuadrada del error cuadrático medio* (RMSE) y *porcentaje de acierto* (PA) dados por las Ecuaciones 5.1 y 5.2.

$$RMSE = \sqrt{\frac{\sum_{l \in A_C} (\alpha R(l) - C_\beta(l))^2}{|A_C|}} \quad (5.1)$$

$$PA = \frac{\sum_{l_1 \in A_C} \sum_{l_2 \in A_C} f_a(l_1, l_2)}{|A_C|^2} \quad (5.2)$$

Primero se determinan los valores $\alpha, \beta_1 \in [0.1, 0.5]$, $\beta_2 \in [0.6, 2.5]$, $\beta_3 \in [2.6, 9.9]$ y $\beta_4 \in [2.6, 9.9]$. Estos valores son ajustados resolviendo el problema de minimización planteado en la Ecuación 5.3, donde $C_{tipo}(l)$ es el conteo de vehículos de tipo *tipo* en el arco l . Entre los conteos realizados para cada instancia de evaluación, se encuentran los conteos en todos los arcos origen $A_O \subset A_C$ (independientemente del resultado de priorización). Estos conteos permitieron conocer las proporciones de tipos de vehículo utilizadas para la generación de las instancias de configuración.

$$\text{Minimizar } \sum_{l \in A_O} |\alpha R(l) - (\beta_1 C_{light}(l) + \beta_2 C_{pcar}(l) + \beta_3 C_{heavy}(l) + \beta_4 C_{bus}(l))| \quad (5.3)$$

Luego, se calcula RMSE y PA, donde A_C son todos aquellos arcos donde se realizaron conteos, $C_\beta(l) = \beta_1 C_{light}(l) + \beta_2 C_{pcar}(l) + \beta_3 C_{heavy}(l) + \beta_4 C_{bus}(l)$ y $f_a(l_1, l_2)$ está dada por la Ecuación 5.4.

$$f_a(l_1, l_2) = \begin{cases} 1 & \text{si } R(l_1) \geq R(l_2) \text{ y } C_\beta(l_1) \geq C_\beta(l_2) \\ 1 & \text{si } R(l_1) < R(l_2) \text{ y } C_\beta(l_1) < C_\beta(l_2). \\ 0 & \text{en otro caso.} \end{cases} \quad (5.4)$$

En el Cuadro 5.1 se pueden ver los resultados del cálculo de RMSE y PA para todas las instancias. El RMSE alcanza su valor máximo de 61.0 para la instancia Jacinto Vera/tarde, y se mantiene por debajo de 30.5 para las demás instancias. Si bien esto equivale a una diferencia de 30 pcar aproximadamente para un $\beta_2 = 1.0$, en la mayoría de los casos el RMSE es favorable al valor ranking como estimador del

volumen vehicular. Se obtuvo además, un porcentaje de acierto mayor al 60.0 % en todas las instancias (llegando incluso al 85.9 %). Este valor complementa los resultados anteriores y se puede concluir que el ranking $R(l)$ utilizado es un buen estimador del volumen vehicular en el arco l .

5.4. Calibración paramétrica

Se llevó a cabo un análisis paramétrico de los AEMO propuestos utilizando 3 instancias de configuración (distintas a las instancias de evaluación, de modo de evitar sesgos en la parametrización). Estas instancias se asemejan a las zonas en estudio en términos de estructura y diversidad de tráfico, pero abarcan una zona geográfica menor y tienen menos semáforos (ver Figura 5.4). En conformidad con la instancias de evaluación se consideran vehículos de tipo light, pcar, heavy y bus. Las simulaciones tienen una duración de 5 minutos con ingreso de vehículos cada 0.8, 0.7 y 0.6 segundos, respectivamente. Tenemos:

- *Instancia de configuración 1 (IC1)*: se trata de un escenario pequeño con tramado simple que cuenta con una avenida principal semaforizada. Se tienen en total 7 intersecciones simples semaforizadas. La simulación tiene una duración de 5 minutos durante los cuales ingresa 1 vehículo cada 0.8 segundos al escenario (375 vehículos en total).
- *Instancia de configuración 2 (IC2)*: cuenta con una intersección compleja semaforizada y otras 10 intersecciones simples de las cuales 4 están semaforizadas. La simulación dura 5 minutos e ingresa 1 vehículo cada 0.7 segundos (428 vehículos).
- *Instancia de configuración 3 (IC3)*: escenario de trama más compleja que IC1 e IC2. Abarca un total de 15 intersecciones semaforizadas. La simulación dura 5 minutos e ingresa 1 vehículo cada 0.6 segundos (500 vehículos).

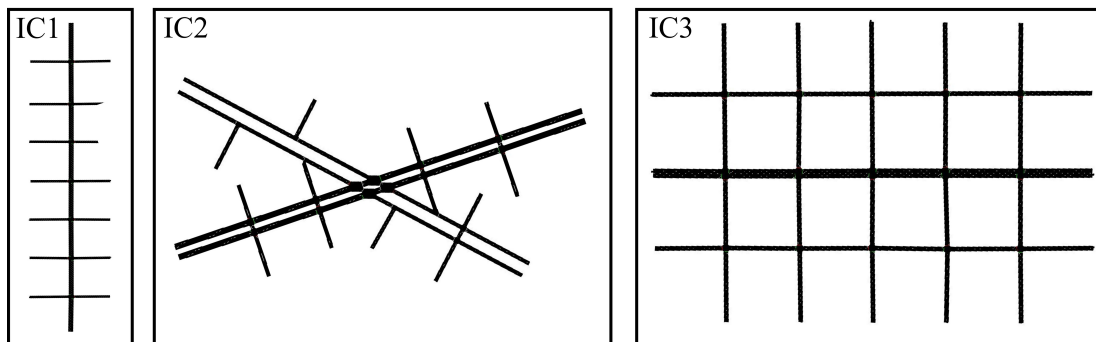


Figura 5.4: Instancias utilizadas para la calibración de los algoritmos.

etapa	AEMO	configuraciones	parámetros fijados	elegido
1	NSGA-II	($\#P=50$, $mce=10000$), ($\#P=75$, $mce=15000$), ($\#P=100$, $mce=20000$)	$p_{SBX}=0.7$, $p_{PM}=0.01$	$\#P=100$, $mce=20000$
1	GDE3	($\#P=100$, $mce=10000$), ($\#P=150$, $mce=15000$), ($\#P=200$, $mce=20000$)	$p_{CR}=0.1$, $F=0.5$	$\#P=200$, $mce=20000$
1	MOEA/D	($\#P=100$, $mce=10000$), ($\#P=150$, $mce=15000$), ($\#P=200$, $mce=20000$)	$p_{CR}=0.1$, $p_{PM}=0.01$	$\#P=200$, $mce=20000$
2	NSGA-II	$p_{SBX} \in \{0.7, 0.8, 0.9\}$, $p_{PM} \in \{0.005, 0.01, 0.05\}$	$\#P=100$, $mce=20000$	$p_{SBX}=0.9$, $p_{PM}=0.05$
2	GDE3	$p_{CR} \in \{0.1, 0.2, 0.9\}$, $F \in \{0.2, 0.4, 0.5\}$	$\#P=200$, $mce=20000$	$p_{CR}=0.1$, $F=0.2$
2	MOEA/D	$p_{CR} \in \{0.1, 0.2, 0.9\}$, $p_{PM} \in \{0.005, 0.01, 0.05\}$	$\#P=200$, $mce=20000$	$p_{CR}=0.2$, $p_{PM}=0.05$
3	ϵ-NSGA-II	$\#P \in \{10, 20\}$, $\epsilon \in \{1, 0.1\}$	$p_{SBX}=0.9$, $p_{PM}=0.05$	$\#P=10$, $\epsilon=1$

Cuadro 5.2: Resumen de la calibración paramétrica aplicada a los algoritmos.

La calibración paramétrica fue realizada en tres etapas. La primer etapa tuvo como objetivo determinar el tamaño de la población ($\#P$) y la máxima cantidad de evaluaciones (mce) de los algoritmos NSGA-II, GDE3 y MOEA/D fijando los demás parámetros. Luego de determinado $\#P$ y mce , en la segunda etapa se procedió a estudiar las probabilidades de cruzamiento (p_{SBX} , p_{CR}) y de mutación (p_{PM}) de los distintos operadores evolutivos (en el caso de GDE3 se estudió el peso diferencial F y p_{CR}), con 3 valores candidatos por cada parámetro (se probaron todas las combinaciones posibles). Por cada tupla [algoritmo, instancia de configuración, configuración] se realizaron 30 ejecuciones independientes. Finalmente, en la tercer etapa se estudiaron los parámetros $\#P$ y ϵ del algoritmo ϵ -NSGA-II fijando los parámetros elegidos para el NSGA-II. En el Cuadro 5.2 es posible ver un resumen con las configuraciones estudiadas y las configuraciones elegidas para las tres etapas de configuración. Para ver los resultados arrojados por los algoritmos para cada instancia de configuración consultar el Apéndice C.

5.5. Experimentos sobre la zona de Aguada

En esta sección analizaremos los experimentos realizados sobre el escenario Aguada para los tres horarios: mañana, tarde y noche. En la Figura 5.5 se muestra la red SUMO generada para el escenario.

Al igual que en la calibración de parámetros y en el resto de los experimentos, cada algoritmo es ejecutado 30 veces de forma independiente, obteniendo distribuciones de resultados para las métricas: HV, DG, DGI, $I_{\epsilon+}$ y ESP (ver Sección 2.6.1). También se estudia la *contribución* (CON) al frente de Pareto y el tiempo de ejecución (t_e). Para determinar si dos distribuciones son similares o diferentes

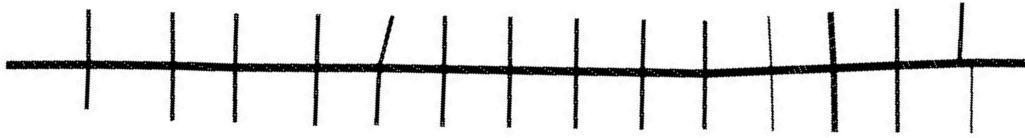


Figura 5.5: Red SUMO para el escenario Aguada.

se aplican los tests estadísticos de Mann-Whitney y Kruskal-Wallis (ver Sección 2.6.2). A partir de las mejores soluciones de todas las ejecuciones se genera el frente de Pareto aproximado FP^* .

5.5.1. Comparación entre AEMO

En primer lugar, comparamos los AEMO entre sí, con el objetivo de conocer cuál obtuvo el mejor rendimiento. En el Apéndice D se pueden consultar los resultados de las ejecuciones. Los Cuadros D.1, D.2 y D.3 reportan el valor mínimo (min), máximo (max), mediana ($\tilde{x}$) y desviación estándar (σ) de las métricas para cada AEMO en los tres horarios. En negrita se indica si se trata de la mejor distribución de resultados (varias medianas en negrita significa que no se encontró diferencia estadísticamente significativa entre estos resultados). De los resultados, resulta fácil inferir la clara superioridad del algoritmo GDE3 frente a los demás AEMO, tanto en eficiencia como en efectividad. No solo mantiene un tiempo de ejecución muy por debajo de los algoritmos MOEA/D y ϵ -NSGA-II sino que obtiene los mejores resultados para las métricas HV, DGI, $I_{\epsilon+}$ y contribución en los tres horarios, indicando una buena aproximación al frente de Pareto conservando buena diversidad. Sin embargo, no debemos olvidar que no se ejecutó una versión paralela del MOEA/D, sino que se paralelizaron las ejecuciones independientes y por eso se registra un tiempo de ejecución tan elevado (correspondiente a la ejecución serial del algoritmo) en contraste con los demás algoritmos. Otra observación es la baja contribución del MOEA/D al frente de Pareto.

5.5.2. Comparación entre AEMO y configuración real

Para evaluar la calidad de las soluciones obtenidas por los AEMO propuestos, se comparan los resultados con los que se obtienen al simular la configuración real actual de los semáforos. Esto es posible dado que se determinaron exactamente los estados de los programas de ciclo de la realidad y su duración (ver Sección 4.4.1). En el Cuadro 5.3 se muestran los porcentajes de mejora de cada uno de los AEMO respecto al resultado arrojado por la configuración real. Analizamos los horarios por orden de volumen vehicular. Si bien, el MOEA/D obtiene en promedio el mejor porcentaje para las emisiones y el tiempo perdido por los vehículos por ir debajo de la velocidad ideal (*tiempo perdido* o TPG), se debe tener en cuenta que la cantidad de vehículos que llegan a destino en tiempo de simulación (*vehículos en destino*) es, para todas las soluciones devueltas, muy inferior a la configuración real. Además,

sabemos por la comparación anterior que su contribución al frente de Pareto es casi nula, por lo que los demás resultados son irrelevantes.

Horario de la noche: en el horario nocturno es donde se observan menos vehículos: la matriz OD estimada cuenta con un total de 463 vehículos que ingresan al escenario a lo largo de la simulación (59 de tipo light, 379 pcar, 3 heavy y 22 bus). En el mejor caso, el algoritmo GDE3 es capaz de mejorar la situación real hasta en un **26.2 %** las emisiones de HC y superior para las demás emisiones. Se mejora hasta en un **62.7 %** (26.3 % en promedio) el tiempo perdido y en un **63.8 %** la cantidad de vehículos en destino. Los algoritmos NSGA-II y ϵ -NSGA-II obtuvieron resultados bastante similares en cuanto a las emisiones aunque inferiores. En el mejor caso, la cantidad de vehículos en destino es baja (5.2 % para el NSGA-II y 3.9 % para el ϵ -NSGA-II).

Horario de la mañana: en este horario, la matriz OD estimada consta de 866 vehículos. De estos, 115 son light, 666 son pcar, 32 son heavy y 53 son bus. El GDE3 alcanza, en el mejor caso, mejoras de hasta un **31.5 %** en la reducción de HC llegando a un **55.0 %** en la reducción de NO_x . La mejora en el tiempo perdido es de **77.1 %** en el mejor caso y 48.9 % en promedio. El NSGA-II alcanza el mejor porcentaje de mejora para los vehículos en destino (**10.6 %**).

Horario de la tarde: para el horario de la tarde, la matriz OD consta de 950 vehículos, aproximadamente el doble de lo obtenido para el horario de la noche. De estos, 144 son light, 749 son pcar, 12 son heavy y 45 son bus. En el mejor caso del NSGA-II, aproximadamente 69 vehículos más, llegan destino en tiempo de simulación, representando un **11.6 %** de mejora. Se obtiene además hasta un **77.0 %** de mejora en el tiempo perdido. A pesar de ser una horario con gran volumen de tráfico, el GDE3 logra reducir las emisiones de CO, CO_2 , HC y PM_x hasta en un **36.4 %**, **46.7 %**, **34.1 %** y **43.2 %** respectivamente (la mayor reducción alcanzada para este escenario).

En las Figuras 5.6, 5.7 y 5.8 se muestran los frentes de Pareto generados por la ejecución de cada algoritmo, combinando los resultados de las emisiones (normalizadas) y graficados contra la cantidad de vehículos que llegan a destino en tiempo de simulación y luego contra el tiempo perdido. También se muestra la referencia correspondiente a la configuración real. Si bien, la solución real es buena en cuanto a los vehículos en destino, desde un punto de vista, se observa claramente un amplio margen de mejora, que los algoritmos pueden explotar adecuadamente. Observando los frentes es posible entender la razón por la cuál el MOEA/D tiene un porcentaje alto de mejora respecto a la solución real. Esto se debe a que, al no conseguir la movilidad de los vehículos, la mayoría no comience su viaje, y por lo tanto no acumulen tiempo perdido ni emitan contaminantes.

5.5.3. Comparación con técnicas alternativas

Debido a la estructura de la red de tráfico surge, en forma inmediata, la comparación con la técnica de *onda verde* u *ola verde*. Esta consiste en una práctica habitual de coordinación de dos o más programas de ciclos a lo largo de una misma calle, para permitir el flujo de tráfico continuo en una única dirección. Cualquier

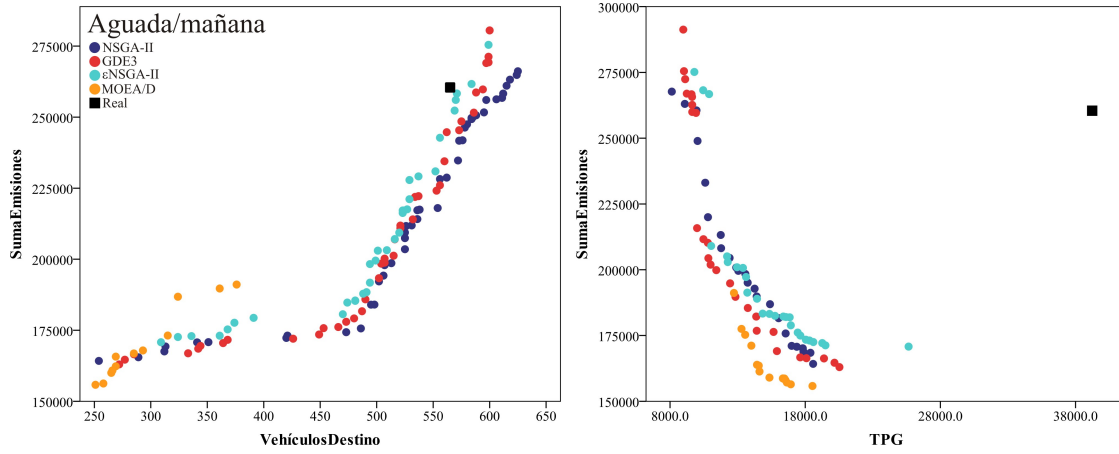


Figura 5.6: Frentes de Pareto de AEMO junto a la referencia real para la instancia Aguada/mañana.

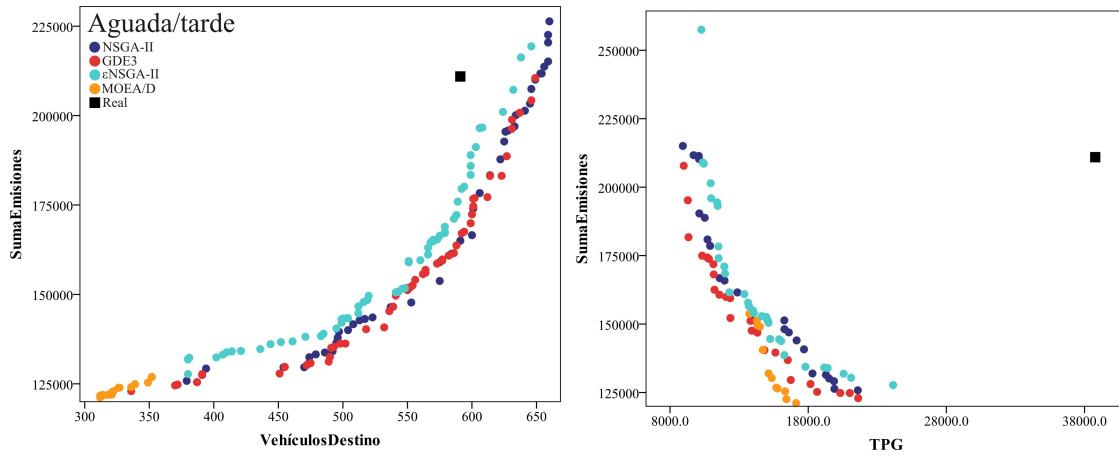


Figura 5.7: Frentes de Pareto de AEMO junto a la referencia real para la instancia Aguada/tarde.

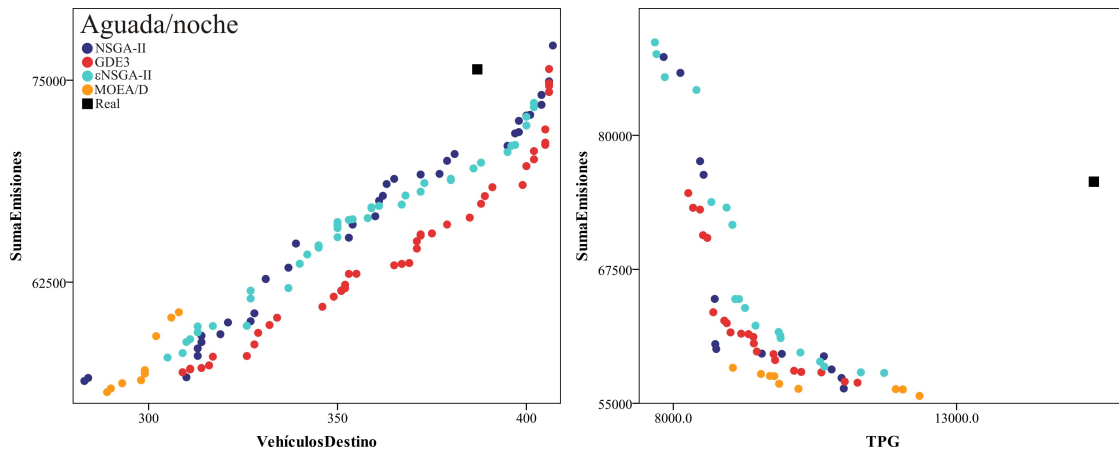


Figura 5.8: Frentes de Pareto de AEMO junto a la referencia real para la instancia Aguada/noche.

% de mejora		NSGA-II		GDE3		MOEA/D		ϵ-NSGA-II	
horario mañana									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	5934.5g	34.1 %	1.8 %	35.2 %	6.6 %	33.6 %	28.0 %	31.5 %	11.1 %
CO ₂	252717.1g	45.6 %	23.8 %	45.2 %	27.3 %	48.4 %	46.5 %	43.1 %	30.1 %
HC	421.6g	30.5 %	-16.6 %	31.5 %	-10.3 %	22.8 %	14.2 %	28.4 %	-4.9 %
PM _x	44.5g	38.7 %	14.9 %	38.7 %	18.8 %	41.8 %	39.4 %	36.4 %	22.2 %
NO _x	891.3g	52.8 %	28.1 %	55.0 %	31.4 %	57.1 %	52.4 %	52.0 %	33.4 %
VDG	565	10.6 %	-39.0 %	6.1 %	-40.5 %	-33.4 %	-56.9 %	6.0 %	-43.4 %
TPG	39232.8s	79.3 %	47.6 %	77.1 %	48.9 %	67.6 %	54.5 %	75.0 %	48.2 %
horario tarde									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	6026.4g	36.1 %	6.5 %	36.4 %	13.0 %	36.1 %	32.8 %	34.0 %	14.1 %
CO ₂	207726.5g	46.2 %	21.9 %	46.7 %	27.9 %	48.9 %	46.4 %	43.9 %	29.8 %
HC	443.9g	32.9 %	-6.1 %	34.1 %	1.8 %	27.8 %	23.0 %	29.4 %	3.2 %
PM _x	36.4g	41.7 %	17.3 %	43.2 %	24.1 %	45.0 %	42.2 %	41.0 %	25.5 %
NO _x	491.1g	61.9 %	33.8 %	62.9 %	43.1 %	63.1 %	59.6 %	61.9 %	45.4 %
VDG	591	11.6 %	-28.3 %	9.8 %	-29.7 %	-40.4 %	-48.2 %	9.3 %	-31.6 %
TPG	38773.5s	77.0 %	43.5 %	76.8 %	44.5 %	64.5 %	52.1 %	73.5 %	43.9 %
horario noche									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	2048.5g	27.1 %	7.0 %	27.2 %	14.4 %	24.5 %	18.6 %	25.6 %	10.2 %
CO ₂	77785.4g	29.3 %	14.1 %	39.4 %	18.4 %	31.7 %	28.8 %	26.3 %	15.3 %
HC	151.8g	25.8 %	2.0 %	26.2 %	11.7 %	22.3 %	12.9 %	24.4 %	6.5 %
PM _x	13.2g	25.7 %	11.4 %	35.5 %	16.2 %	27.2 %	24.5 %	23.6 %	12.7 %
NO _x	190.7g	44.0 %	18.1 %	64.0 %	20.4 %	44.5 %	36.2 %	40.5 %	16.6 %
VDG	387	5.2 %	-16.6 %	63.8 %	-16.5 %	-20.4 %	-28.4 %	3.9 %	-18.2 %
TPG	15422.5s	49.2 %	20.6 %	62.7 %	26.3 %	41.3 %	24.2 %	50.2 %	23.8 %

Cuadro 5.3: Porcentaje de mejora de AEMO y configuración real para la instancia Aguada.

vehículo que viaja *dentro* de la onda verde (a una velocidad aproximada definida para la onda en particular), se encontrará con una cascada progresiva de luces verdes sin tener que detenerse en las intersecciones. El uso de esta técnica mejora, entre otras cosas, la seguridad en las calles, reduce el consumo de combustible y la emisión de contaminantes, entre otros aspectos (Wu et al., 2014; Kelly, 2012; Warberg et al., 2008).

Para implementar esta técnica en la red de tráfico, se hizo uso del *offset* de los programas de ciclo asignando un tiempo relativo al *offset* anterior:

$$o_i = o_{i-1} + d(p_i, p_{i-1})/v_{ov} \quad (5.5)$$

donde $d(p_i, p_{i-1})$ es la distancia del programa p_i a p_{i-1} y v_{ov} es una velocidad determinada para la onda verde. En nuestro caso se consideraron todas las velocidades entre 15Km/h y 45Km/h en ambas direcciones. Para que la onda verde se repita constantemente, la duración de los estados verdes y rojos debe ser igual para todos los programas. Para cada tupla [velocidad, dirección] se probaron todas las duraciones $d \in \{10s, 20s, 30s, 40s, 50s\}$. Es importante mencionar que solo un grupo de

% de mejora - onda verde - Aguada									
	horario mañana			horario tarde			horario noche		
obj.	real	<i>max</i>	$\bar{x}$	real	<i>max</i>	$\bar{x}$	real	<i>max</i>	$\bar{x}$
CO	5934.5	13.7 %	-6.7 %	6026.4	14.5 %	-7.7 %	2048.5	16.2 %	-13.3 %
CO ₂	252717.1	27.8 %	5.3 %	207726.5	21.7 %	3.9 %	77785.4	16.4 %	-4.5 %
HC	421.6	12.4 %	-16.5 %	443.9	14.5 %	-16.0 %	151.8	15.3 %	-19.9 %
PM _x	44.5	20.8 %	0.2 %	36.4	17.4 %	-0.8 %	13.2	14.6 %	-8.0 %
NO _x	891.3	36.4 %	6.1 %	491.1	42.4 %	9.0 %	190.7	22.0 %	-5.2 %
VDG	565.0	5.7 %	-23.0 %	591.0	7.3 %	-18.5 %	387.0	-2.6 %	-15.8 %
TPG	39232.8	56.7 %	13.3 %	38773.5	65.7 %	14.4 %	15422.5	24.4 %	-11.5 %

Cuadro 5.4: Porcentaje de mejora de la técnica onda verde sobre la configuración real para la instancia Aguada.

vehículos, puede hacer uso de la onda verde antes de que esta vuelva a comenzar y dar lugar a otro flujo ininterrumpido de tráfico.

En el Cuadro 5.4 se puede ver el porcentaje de mejora alcanzado por la onda verde sobre la configuración real para los tres horarios. Si bien en el mejor caso, se obtienen mejoras considerables para las emisiones y el tiempo perdido, no se superan los valores obtenidos para los AEMO NSGA-II, GDE3 y ϵ -NSGA-II en ninguno de los horarios. Esta superioridad puede explicarse debido a que los AEMO no mejoran solo el flujo de una sucesión de calles sino de todo un conjunto lo que permite fluidez en todas ellas y resulta en mejores resultados a nivel global. En el horario de la noche, la cantidad de vehículos en destino es incluso inferior al caso real (-2.6 %) para todas las soluciones.

Las Figuras 5.9, 5.10 y 5.11 presentan el frente de Pareto aproximado FP^* generado por la ejecución de todos los algoritmos y la técnica onda verde junto a la referencia correspondiente a la configuración real. En el horario de la mañana se observan algunos puntos no dominados pertenecientes la técnica onda verde. En las demás, es notoria la superioridad de las soluciones producidas por los AEMO.

5.5.4. Análisis de una solución en el frente de Pareto

Para finalizar el análisis sobre el escenario Aguada, elegiremos una solución particular provista por un AEMO y la aplicaremos a una de las instancias. En este caso, lo haremos para horario de la tarde, debido a que es donde se presenta mayor volumen de tráfico.

La solución elegida se obtuvo con el algoritmo GDE3 y en el Cuadro 5.5 se pueden ver los porcentajes de mejora de esta solución particular sobre la configuración real. En la Figura 5.12 se observa una sección del escenario al finalizar la simulación. Para la configuración elegida, se observa claramente una menor cantidad de vehículos en circulación. Por lo tanto, las soluciones obtenidas por los AEMO mejora la fluidez del tráfico en escenarios reales.

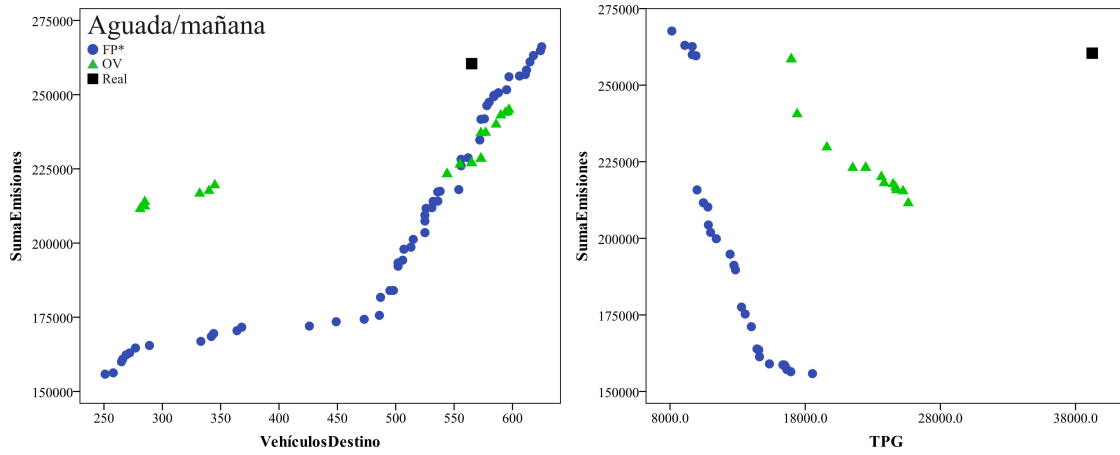


Figura 5.9: Frente de Pareto FP^* y de onda verde junto a la referencia real para la instancia Aguada/mañana.

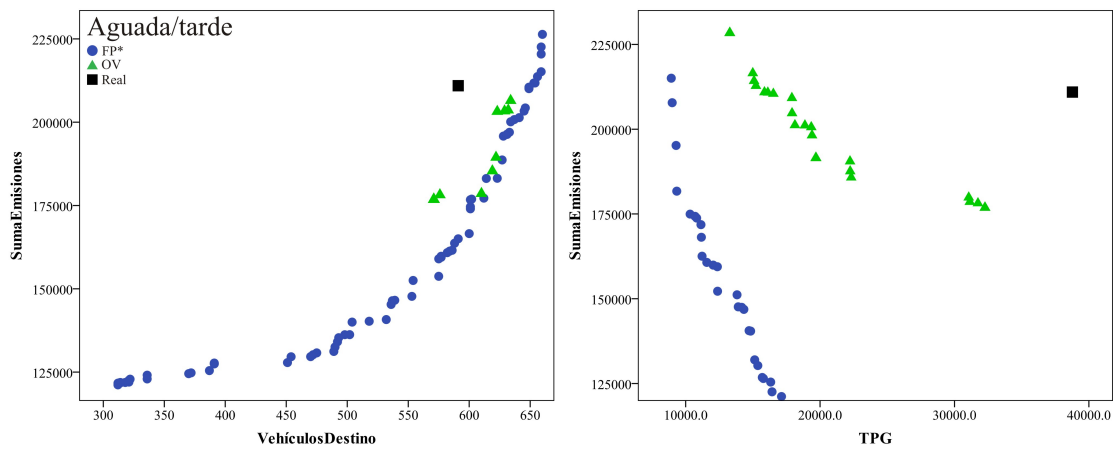


Figura 5.10: Frente de Pareto FP^* y de onda verde junto a la referencia real para la instancia Aguada/tarde.

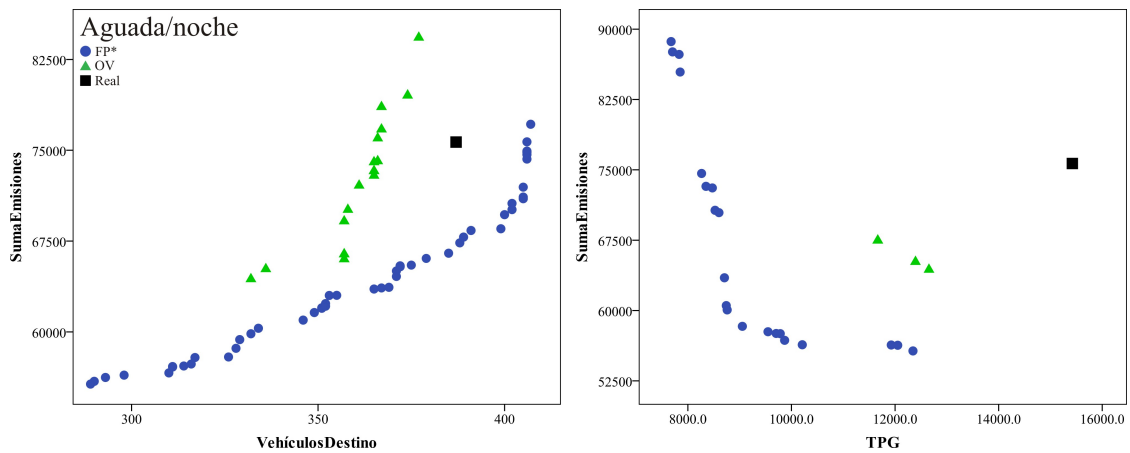


Figura 5.11: Frente de Pareto FP^* y de onda verde junto a la referencia real para la instancia Aguada/noche.

obj.	real	% mejora elegida
CO	6026.4g	9.3 %
CO ₂	207726.5g	12.6 %
HC	443.9g	7.9 %
PM _x	36.4g	12.2 %
NO _x	491.1g	23.6 %
VDG	591.0	5.4 %
TPG	38773.5s	9.2 %

Cuadro 5.5: Porcentaje de mejora de la configuración elegida sobre la real para la instancia Aguada/tarde.

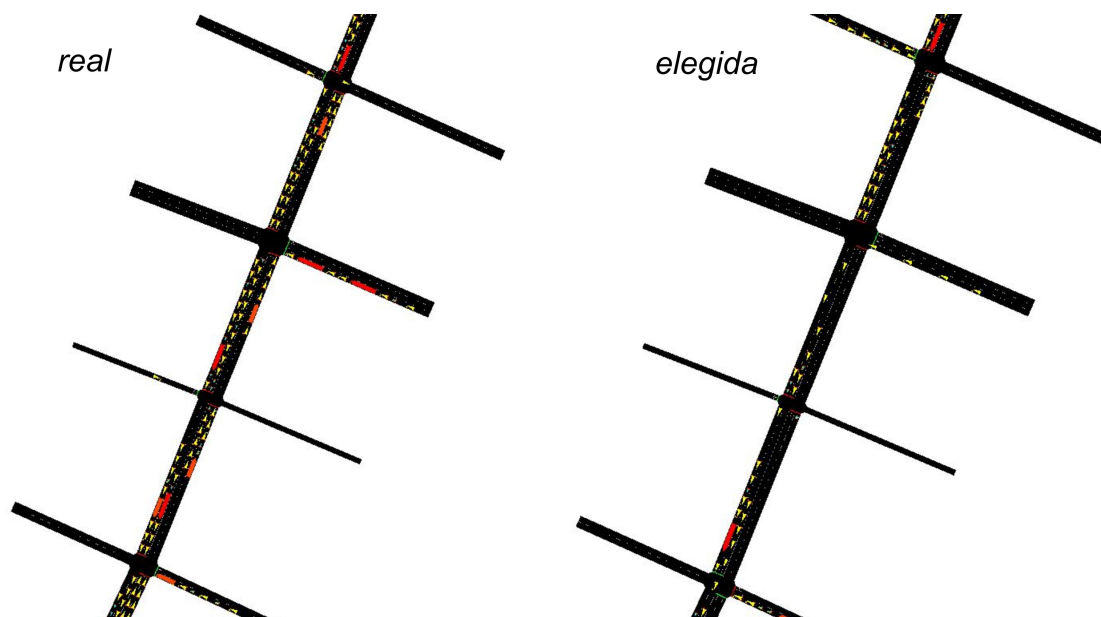


Figura 5.12: Captura del final de la simulación con (a) configuración real y (b) configuración elegida para la instancia Aguada/tarde.

5.6. Experimentos sobre la zona de Jacinto Vera

A continuación se describen los resultados de los experimentos realizados sobre el escenario Jacinto Vera para los horarios: mañana, tarde y noche. En la Figura 5.13 se puede observar la red SUMO generada para el escenario.

Nuestro interés por el estudio de esta zona de Montevideo, recae en su simple pero a la vez compleja red vial. A pesar de abarcar un área relativamente pequeña y con pocas intersecciones semaforizadas posee 3 focos constantes de tráfico. Se trata de tres grandes intersecciones con programas de ciclos complejos. Surge entonces la pregunta de hasta que punto es posible, mediante la configuración de sus pocos programas de ciclo, mejorar el flujo de tráfico de la zona.

Al igual que en la sección anterior, volvimos a comparar los AEMO entre sí (consultar los resultados en el Apéndice D, Cuadros D.4, D.5 y D.6), de modo

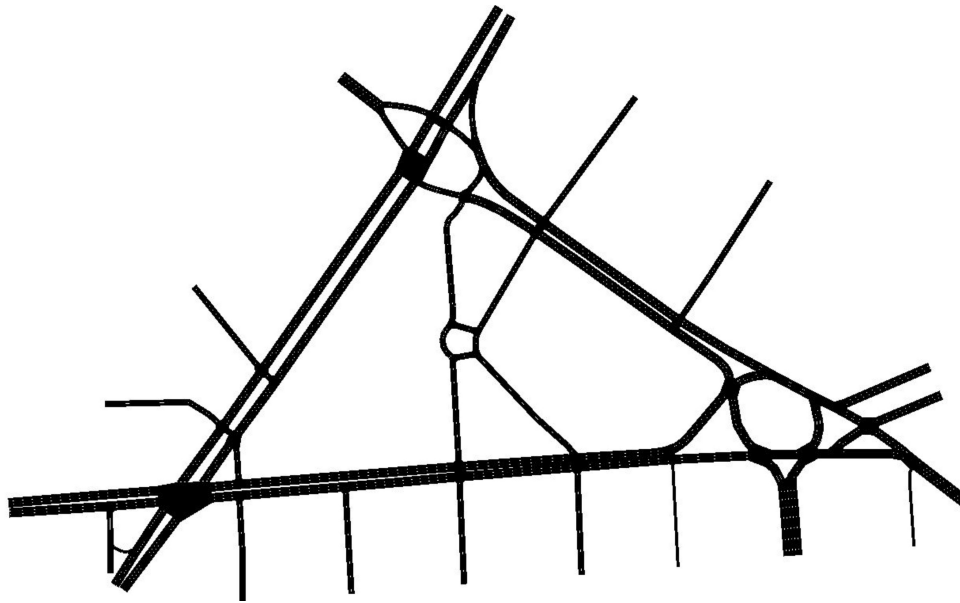


Figura 5.13: Red SUMO para el escenario Jacinto Vera.

de determinar el AEMO de mejor rendimiento para este escenario en particular. Nuevamente el algoritmo GDE3 presenta resultados que denotan su utilidad para resolver el problema de planificación de ciclos de semáforos. En los tres horarios, obtiene los mejores resultados para las métricas HV, DGI, $I_{\epsilon+}$ y contribución con un tiempo de ejecución relativamente bajo. En segundo lugar recaen los algoritmos NSGA-II y ϵ -NSGA-II. MOEA/D presenta los peores resultados (alta DG, DGI, $I_{\epsilon+}$, espaciamiento y baja contribución al frente de Pareto).

5.6.1. Comparación entre AEMO y configuración real

En el Cuadro 5.6 se muestran los porcentajes de mejora de cada uno de los AEMO respecto al resultado arrojado por la configuración real. Por orden de volumen vehicular, tenemos:

Horario de la noche: en el caso nocturno, la matriz OD estimada cuenta con un total de 719 vehículos que ingresan al escenario a lo largo de la simulación (67 de tipo light, 584 pcar, 14 heavy y 54 bus). En el mejor caso, los algoritmos NSGA-II, GDE3 y ϵ -NSGA-II arrojan resultados similares, logrando una mejora superior al **20.0 %** para todas las emisiones vehiculares, más del **7.0 %** respecto a los vehículos que llegan a destino (31 vehículos) y más del **52.0 %** en el tiempo perdido global (4.4 hs).

Horario de la mañana: en este horario, la matriz OD consta de 943 vehículos. De estos, 89 son light, 770 son pcar, 41 son heavy y 43 son bus. Todos los AEMO llegan hasta el **30.0 %** de mejora en reducción de emisiones. El NSGA-II alcanza el **21.5 %** en vehículos en destino (103 vehículos) y **57.0 %** en tiempo perdido (14.1 % en promedio). El GDE3 obtiene hasta un **33.2 %** de mejora en reducción de material particulado (20.7 % en promedio) y superior para las demás emisiones.

% de mejora		NSGA-II		GDE3		MOEA/D		ϵ-NSGA-II	
horario mañana									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	5707.5g	34.4 %	4.3 %	34.4 %	8.0 %	34.5 %	25.9 %	30.9 %	7.6 %
CO ₂	204606.6g	34.4 %	12.6 %	34.3 %	15.2 %	37.3 %	33.8 %	32.7 %	15.8 %
HC	477.1g	35.9 %	-0.5 %	36.4 %	4.0 %	35.6 %	20.5 %	31.9 %	3.2 %
PM _x	35.6g	32.7 %	11.6 %	33.2 %	14.2 %	35.3 %	31.4 %	31.7 %	14.3 %
NO _x	604.1g	43.5 %	18.9 %	45.5 %	20.7 %	47.6 %	37.3 %	41.1 %	21.8 %
VDG	478	21.5 %	-16.7 %	21.3 %	-15.8 %	-11.1 %	-36.1 %	19.2 %	-16.8 %
TPG	32503.7s	57.0 %	14.1 %	55.5 %	14.6 %	39.4 %	11.7 %	52.1 %	14.7 %
horario tarde									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	8237.0g	33.4 %	4.6 %	34.3 %	5.8 %	34.0 %	26.0 %	32.0 %	7.1 %
CO ₂	240943.5g	29.3 %	7.9 %	29.9 %	8.9 %	32.0 %	27.2 %	27.9 %	9.4 %
HC	726.2g	35.7 %	2.7 %	35.9 %	4.5 %	34.4 %	23.0 %	32.8 %	5.8 %
PM _x	42.4g	29.9 %	6.4 %	30.5 %	7.4 %	32.8 %	25.2 %	27.4 %	7.6 %
NO _x	519.2g	34.0 %	8.1 %	34.4 %	9.5 %	31.6 %	19.5 %	30.1 %	7.6 %
VDG	367	40.3 %	-7.6 %	38.4 %	-6.0 %	-12.0 %	-34.5 %	36.5 %	-5.5 %
TPG	36800.3s	68.0 %	24.7 %	69.4 %	27.7 %	57.3 %	26.8 %	67.5 %	25.3 %
horario noche									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	4723.8g	23.8 %	4.2 %	23.7 %	6.9 %	23.3 %	16.6 %	21.7 %	6.6 %
CO ₂	173565.9g	21.1 %	10.1 %	21.3 %	12.4 %	22.6 %	20.9 %	20.1 %	11.3 %
HC	396.1g	25.8 %	-0.4 %	26.3 %	2.7 %	24.0 %	13.0 %	24.1 %	3.0 %
PM _x	30.1g	23.3 %	11.6 %	22.8 %	14.3 %	24.4 %	21.7 %	22.4 %	13.2 %
NO _x	493.6g	31.3 %	15.5 %	31.0 %	18.3 %	30.1 %	24.0 %	29.5 %	16.5 %
VDG	442	9.5 %	-16.6 %	9.3 %	-17.1 %	-10.6 %	-19.4 %	7.7 %	-15.4 %
TPG	30651.5s	54.9 %	31.2 %	55.7 %	32.6 %	47.3 %	26.3 %	52.6 %	30.0 %

Cuadro 5.6: Porcentaje de mejora de AEMO sobre la configuración real para la instancia Jacinto Vera.

Horario de la tarde: el horario de la tarde constituye el horario con mayor flujo vehicular. Se registran un total de 121 vehículos de tipo light, 979 pcar, 25 heavy y 43 bus (suman 1168 vehículos). Si bien, los resultados promedio de mejora de los AEMO para la cantidad de vehículos en destino siguen siendo negativos (en promedio la cantidad de vehículos en destino es menor que para la configuración real, al igual que en las instancias del escenario Aguada), el ϵ -NSGA-II obtiene un porcentaje de -5.5 %, el mejor hasta el momento, a pesar de tratarse de la instancia con mayor volumen de tráfico. Además, la mejora promedio en reducción de emisiones es superior al **2.6 %** para todos los AEMO. Para el tiempo perdido, el GDE3 alcanza el mejor porcentaje (**69.4 %**).

En las Figuras 5.14, 5.15 y 5.16 se muestran los frentes de Pareto generados por la ejecución de todos los algoritmos y la referencia correspondiente a la configuración real. Como ya se hacía notar en los comentarios anteriores, existe una mejora sustancial para la cantidad de vehículos en destino en comparación con el escenario Aguada. Esto se puede observar en los frentes los cuáles están considerablemente más alejados de la referencia real. Es evidente además la buena proximidad del GDE3 al frente de Pareto aproximado.

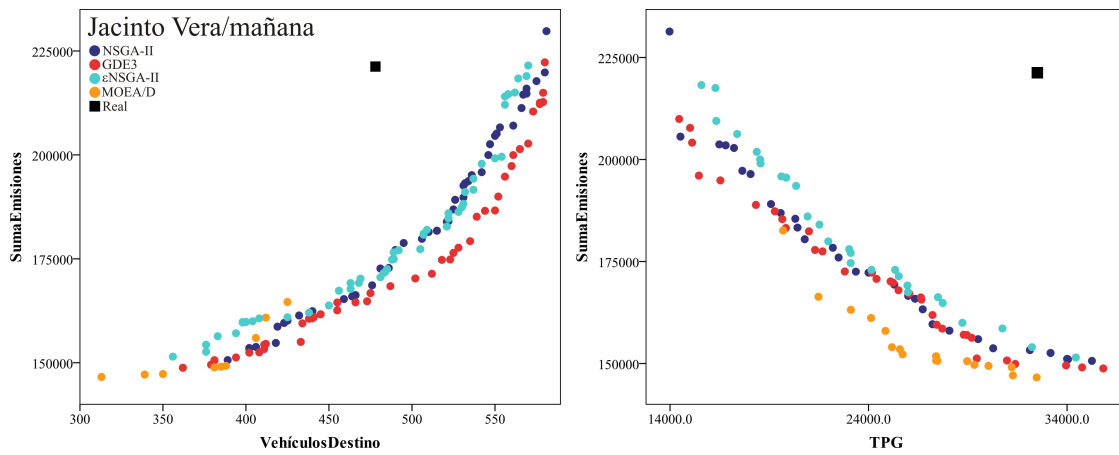


Figura 5.14: Frentes de Pareto de AEMO junto a la referencia real para la instancia Jacinto Vera/mañana.

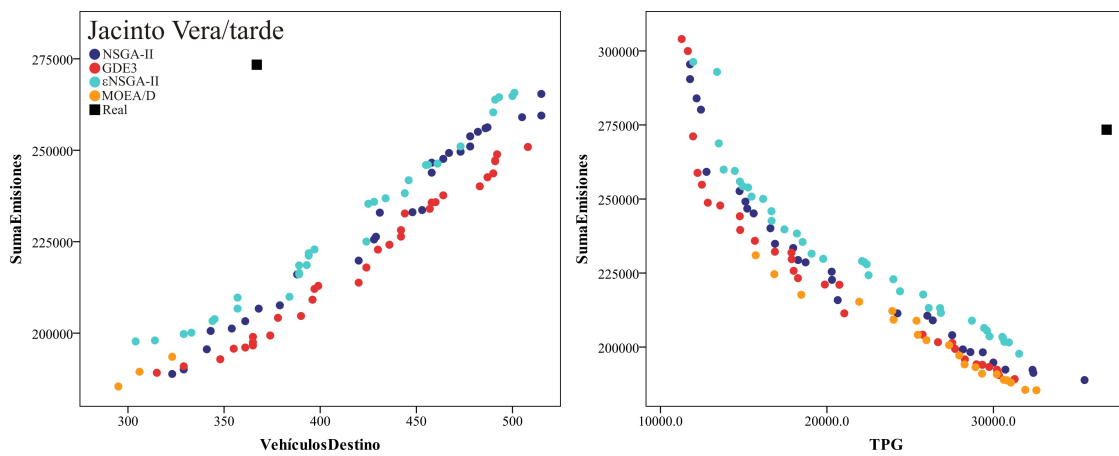


Figura 5.15: Frentes de Pareto de AEMO junto a la referencia real para la instancia Jacinto Vera/tarde.

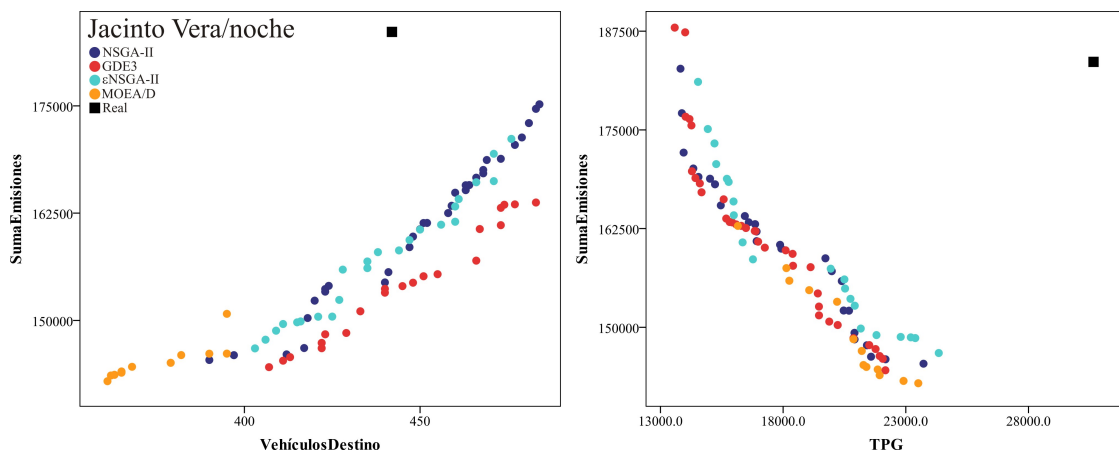


Figura 5.16: Frentes de Pareto de AEMO junto a la referencia real para la instancia Jacinto Vera/noche.

obj.	real	% mejora elegida
CO	8237.0g	15.5 %
CO ₂	240943.5g	7.9 %
HC	726.2g	20.5 %
PM _x	42.4g	9.7 %
NO _x	519.2g	1.3 %
VDG	367	21.8 %
TPG	36800.3s	1.0 %

Cuadro 5.7: Porcentaje de mejora de la configuración elegida sobre la real para la instancia Jacinto Vera/tarde.

5.6.2. Análisis de una solución en el frente de Pareto

Como caso particular elegiremos una solución para el horario de la tarde, donde a pesar del gran volumen vehicular, se obtuvieron soluciones con mejoras importantes respecto a la configuración actual de la red de tráfico.

La solución elegida pertenece al algoritmo GDE3 y en el Cuadro 5.7 se pueden ver los porcentajes de mejora de esta solución particular sobre la configuración real. En la Figura 5.17 se observa una de las grandes intersecciones del escenario al finalizar la simulación. Para la configuración elegida, se observa una red más despejada con menor cantidad de vehículos en circulación.

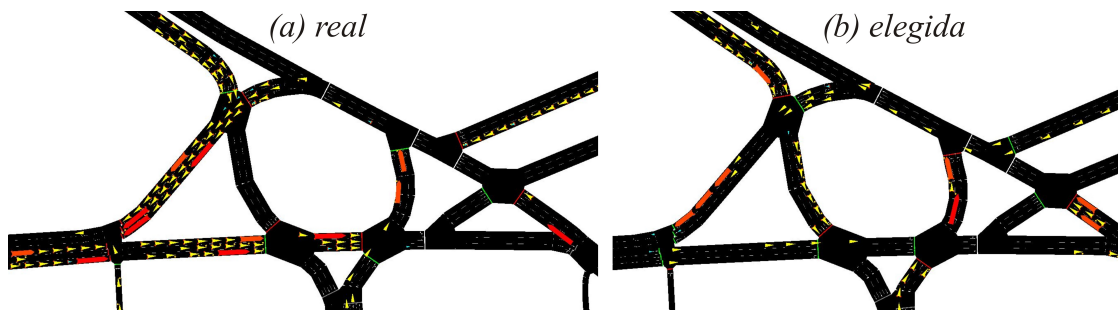


Figura 5.17: Captura del final de la simulación con (a) configuración real y (b) configuración elegida para la instancia Jacinto Vera/tarde.

5.7. Experimentos sobre la zona Centro

A continuación se describen los resultados de los experimentos realizados sobre el escenario Centro para los horarios: mañana, tarde y noche. En la Figura 5.18 se puede observar la red SUMO generada para el escenario. Este escenario constituye la etapa más importante dentro del análisis experimental, debido a la gran cantidad de intersecciones semaforizadas (41 programas de ciclos). Esto implica, entre otras cosas, el manejo de una gran cantidad de variables a optimizar por parte de los algoritmos evolutivos (genotipos con una gran cantidad de genes). Si bien,

esto no debería representar un desafío mayor, el número de objetivos a optimizar podría resultar en una disminución general de la calidad de las soluciones obtenidas (Praditwong and Yao, 2007).

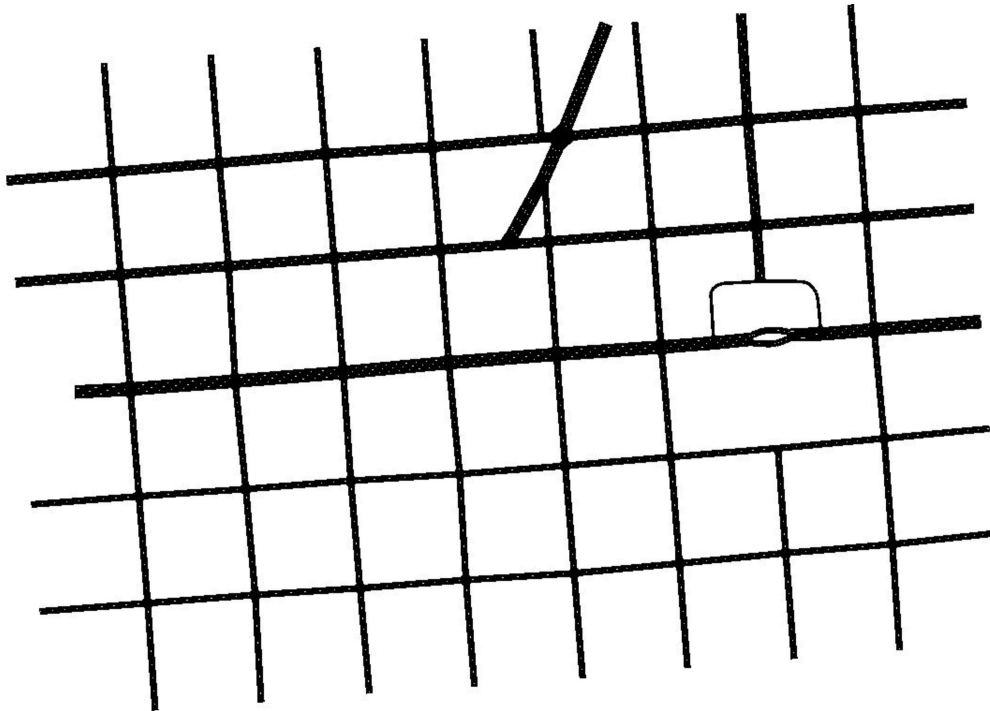


Figura 5.18: Red SUMO para el escenario Centro.

Los resultados de la ejecución de los AEMO se puede consultar en el Apéndice D, Cuadros D.7, D.8 y D.9. El GDE3 volvió a lograr los mejores resultados de HV, DGI, I_{c+} y contribución. En este caso devolvió además los mejores tiempos de ejecución, superando ampliamente al NSGA-II (hasta el momento el AEMO más eficiente).

5.7.1. Comparación entre AEMO y configuración real

El Cuadro 5.8 reporta los porcentajes de mejora de cada uno de los AEMO respecto al resultado arrojado por la configuración real. Por orden de volumen vehicular, tenemos:

Horario de la noche: en este caso, se observa un flujo vehicular liviano: la matriz OD estimada tiene un total de 586 vehículos que ingresan al escenario a lo largo de la simulación (41 de tipo light, 485 pcar, 2 heavy y 58 bus). En el mejor caso, el NSGA-II es capaz de mejorar la situación real hasta en **8.6 %** en las emisiones de CO, y superior para las demás emisiones. Se mejora hasta en un **46.2 %** (25.9 % en promedio) el tiempo de los vehículos que transitan por debajo de la velocidad ideal y en un **21.8 %** los vehículos que llegan a destino (4.7 % en promedio). Esta es la primera vez que se observa un porcentaje de mejora promedio mayor a cero para los vehículos en destino. El GDE3 y el ϵ -NSGA-II obtienen resultados similares. Si bien,

% de mejora		NSGA-II		GDE3		MOEA/D		ϵ-NSGA-II	
horario mañana									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	7396.3g	19.3 %	2.7 %	20.4 %	9.5 %	18.8 %	15.0 %	17.6 %	7.8 %
CO ₂	304699.0g	23.8 %	8.5 %	25.9 %	14.6 %	29.8 %	26.9 %	21.0 %	12.3 %
HC	631.6g	23.9 %	1.2 %	23.6 %	9.9 %	16.8 %	11.7 %	20.6 %	8.6 %
PM _x	53.9g	17.9 %	4.3 %	20.5 %	10.3 %	23.3 %	20.4 %	16.4 %	8.4 %
NO _x	1060.0g	30.3 %	9.1 %	32.6 %	16.2 %	35.8 %	31.2 %	28.9 %	13.5 %
VDG	633	43.6 %	13.7 %	37.2 %	13.0 %	-6.9 %	-32.7 %	35.5 %	12.2 %
TPG	57307.6s	57.6 %	30.8 %	52.8 %	30.8 %	45.4 %	32.2 %	53.0 %	25.9 %
horario tarde									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	6369.5g	17.7 %	-6.0 %	18.4 %	1.4 %	13.8 %	10.2 %	15.0 %	-1.0 %
CO ₂	268274.7g	20.9 %	5.6 %	21.3 %	9.5 %	25.0 %	21.7 %	18.8 %	7.3 %
HC	532.1g	16.9 %	-14.9 %	17.3 %	-5.0 %	10.8 %	2.6 %	15.2 %	-7.7 %
PM _x	46.6g	20.9 %	3.9 %	20.6 %	8.3 %	22.2 %	18.3 %	17.2 %	6.0 %
NO _x	890.7g	28.0 %	8.9 %	30.2 %	11.7 %	32.2 %	25.1 %	28.1 %	9.1 %
VDG	652	11.3 %	-30.7 %	10.1 %	-25.3 %	-12.1 %	-40.5 %	8.6 %	-27.5 %
TPG	50626.9s	58.1 %	26.7 %	54.5 %	26.9 %	37.2 %	22.7 %	55.8 %	22.6 %
horario noche									
obj.	real	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$	max	$\bar{x}$
CO	3282.1g	8.6 %	-0.7 %	8.5 %	0.4 %	4.6 %	0.9 %	8.0 %	-2.2 %
CO ₂	144205.6g	9.7 %	4.2 %	9.9 %	4.6 %	13.2 %	9.9 %	7.8 %	3.0 %
HC	268.2g	13.4 %	0.5 %	13.8 %	1.7 %	5.7 %	0.3 %	12.5 %	-1.5 %
PM _x	24.8g	10.0 %	3.4 %	9.7 %	3.7 %	11.8 %	8.8 %	8.1 %	1.9 %
NO _x	483.7g	22.2 %	9.5 %	20.9 %	9.0 %	27.8 %	19.0 %	19.5 %	7.3 %
VDG	439	21.8 %	4.7 %	18.7 %	-5.1 %	0.4 %	-11.1 %	19.1 %	-6.9 %
TPG	30012.0s	46.2 %	25.9 %	42.0 %	26.6 %	32.5 %	20.0 %	39.5 %	25.6 %

Cuadro 5.8: Porcentaje de mejora de AEMO sobre la configuración real para la instancia Centro.

el MOEA/D consigue en el mejor caso una mejora de vehículos en destino mayor a cero, sigue presentando un promedio bastante inferior a los demás algoritmos.

Horario de la tarde: se cuentan 75 light, 814 pcar, 4 heavy, 114 bus (1007 vehículos). Es importante señalar que mientras el NSGA-II se destaca en maximizar los vehículos que llegan a destino y minimizar el tiempo perdido, el GDE3 reduce considerablemente las emisiones. El ϵ -NSGA-II se encuentra en un término intermedio. El mayor porcentaje alcanzado por el NSGA-II para el tiempo perdido es de **58.1 %**. Con esta mejora el tiempo de viaje promedio de los vehículos se ve reducido a más de la mitad.

Horario de la mañana: la matriz OD consta de 1130 vehículos, aproximadamente el doble de lo obtenido para el horario de la noche. De estos, 94 son light, 892 son pcar, 7 son heavy y 137 son bus. Los resultados indican que NSGA-II alcanza, en el mejor caso, mejoras de hasta **19.3 %** en la reducción de CO y **30.3 %** en la reducción de NO_x. En el mejor caso, 276 vehículos más llegan a destino en el tiempo de simulación, representando un **43.6 %** de mejora. Promediando sobre las soluciones del frente de Pareto aproximado, se mejora en un 13.7 % el número de vehículos en promedio y un 30.8 % en la demora.

onda verde - Centro									
obj.	horario mañana			horario tarde			horario noche		
	real	max	$\bar{x}$	real	max	$\bar{x}$	real	max	$\bar{x}$
CO	7396.3g	11.3 %	0.0 %	6369.5g	7.8 %	-10.1 %	3282.1g	8.5 %	-6.7 %
CO ₂	304699.0g	5.9 %	-5.6 %	268274.7g	6.3 %	-11.0 %	144205.6g	5.4 %	-8.8 %
HC	631.6g	17.7 %	1.9 %	532.1g	13.9 %	-10.4 %	268.2g	14.3 %	-4.1 %
PM _x	53.9g	4.7 %	-6.2 %	46.6g	8.7 %	-10.9 %	24.8g	6.6 %	-9.2 %
NO _x	1060.0g	5.8 %	-12.8 %	890.7g	17.8 %	-14.9 %	483.7g	12.1 %	-12.5 %
VDG	633	14.8 %	-0.7 %	652	10.9 %	-6.4 %	439	6.4 %	-1.9 %
TPG	57307.6s	27.7 %	3.7 %	50626.9s	23.9 %	-0.6 %	30012.0s	22.2 %	1.6 %

Cuadro 5.9: Porcentaje de mejora de la técnica onda verde sobre la configuración real para la instancia Centro.

Las Figuras 5.19, 5.20 y 5.21 se presentan los frentes de Pareto generados por la ejecución de los AEMO y la referencia real. Se observa claramente un amplio margen de mejora en todos los objetivos, que los algoritmos GDE3, NSGA-II e incluso el ϵ -NSGA-II pueden explotar adecuadamente.

5.7.2. Comparación con técnicas alternativas

Al igual que en el escenario Aguada, compararemos el resultado de los AEMO frente a la técnica de onda verde. En este caso, sin embargo, se tienen 3 ondas verdes en forma simultánea una por cada avenida principal del escenario. En todos los casos se estudian las velocidades de onda entre 15km y 45km. Por otro lado, se simulará un método de descongestión manual utilizado actualmente en la ciudad de Montevideo (le llamaremos método DCI). Este método consiste en aplicar la luz verde por dos o tres minutos cuando se detectan, a través de cámaras inteligentes, cantidades extraordinarias de vehículos en una esquina. El método se aplicará en forma individual para cada semáforo del escenario sobre el método de onda verde.

En los cuadros 5.9 y 5.10 se reportan los porcentaje de mejora alcanzados por las técnicas onda verde y onda verde + DCI sobre la configuración real para los tres horarios. La técnica de onda verde obtiene, en el mejor caso, mejoras considerables para todos los objetivos. Al aplicar el método de descongestión DCI a las soluciones devueltas por la onda verde se obtienen mejoras aún mayores. Sin embargo, en comparación con los valores obtenidos para los AEMO, se observan grandes diferencias tanto para las emisiones como para el tiempo perdido.

Las Figuras 5.22, 5.23 y 5.24 presentan el frente de Pareto FP^* generado por los AEMO, el frente de Pareto generado por la técnica de onda verde y el correspondiente a la técnica de onda verde + DCI para cada horario. Se grafican las emisiones normalizadas contra la cantidad de vehículos que llegan a destino en tiempo de simulación y luego contra el tiempo perdido. Se muestra también la referencia real. Si bien ambas técnicas alternativas se acercan bastante a FP^* , obteniendo algunos puntos no nominados en el caso de la tarde, es bastante notoria la superioridad de las soluciones producidas por los AEMO para los objetivos de emisiones y tiempo perdido, como ya se hizo notar anteriormente.

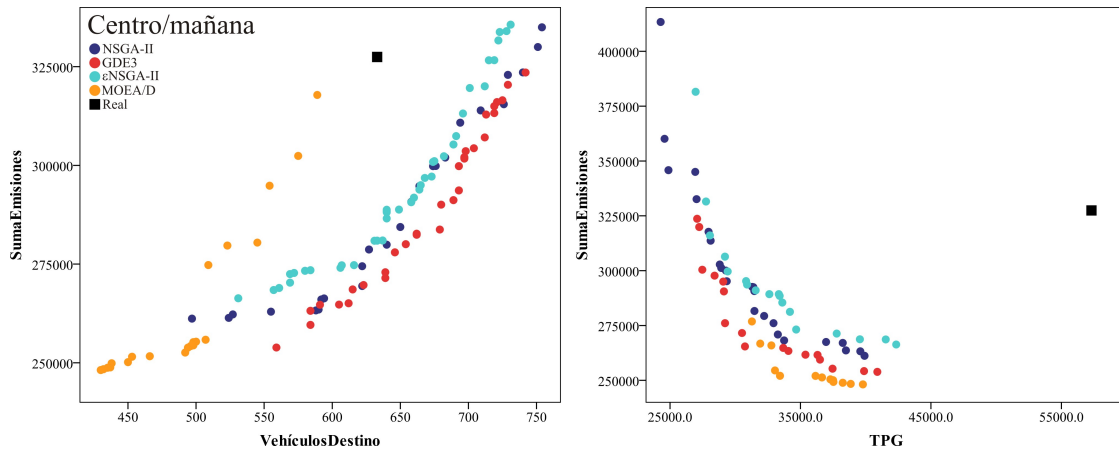


Figura 5.19: Frentes de Pareto de AEMO junto a la referencia real para la instancia Centro/mañana.

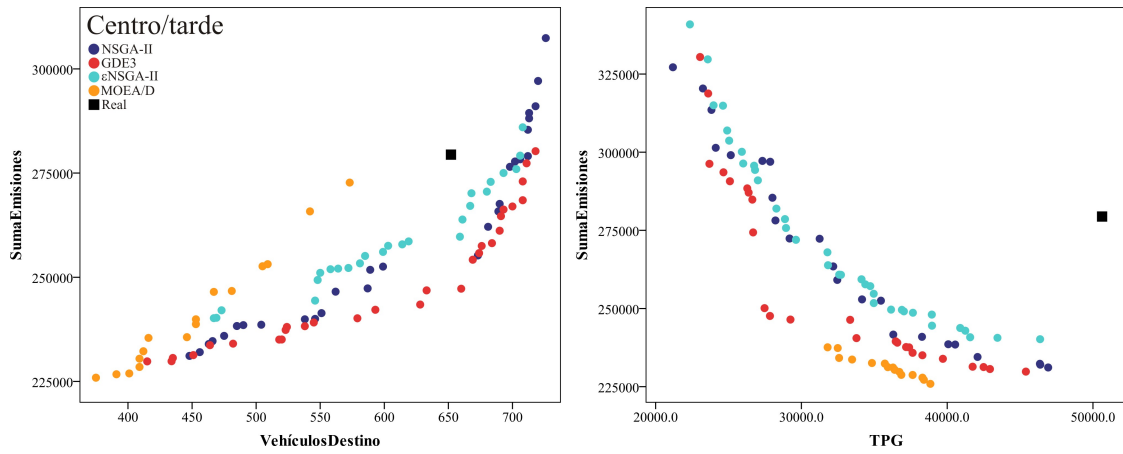


Figura 5.20: Frentes de Pareto de AEMO junto a la referencia real para la instancia Centro/tarde.

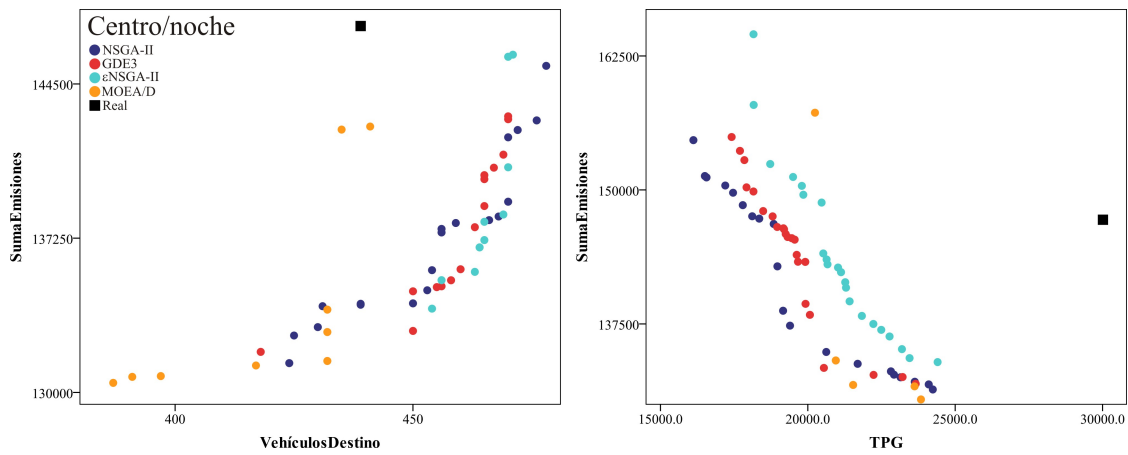


Figura 5.21: Frentes de Pareto de AEMO junto a la referencia real para la instancia Centro/noche.

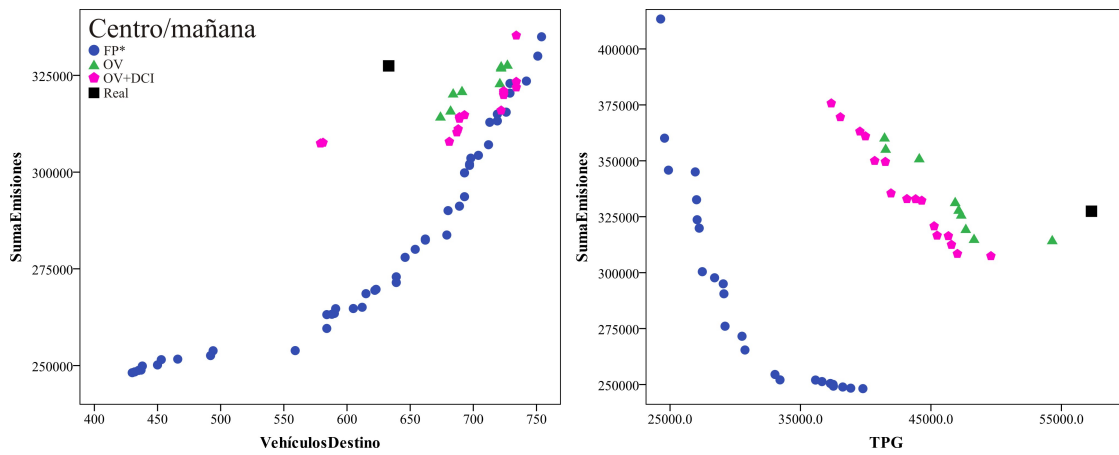


Figura 5.22: Frente de Pareto FP^* , de onda verde y de onda verde + DCI junto a la referencia real para la instancia Centro/mañana.

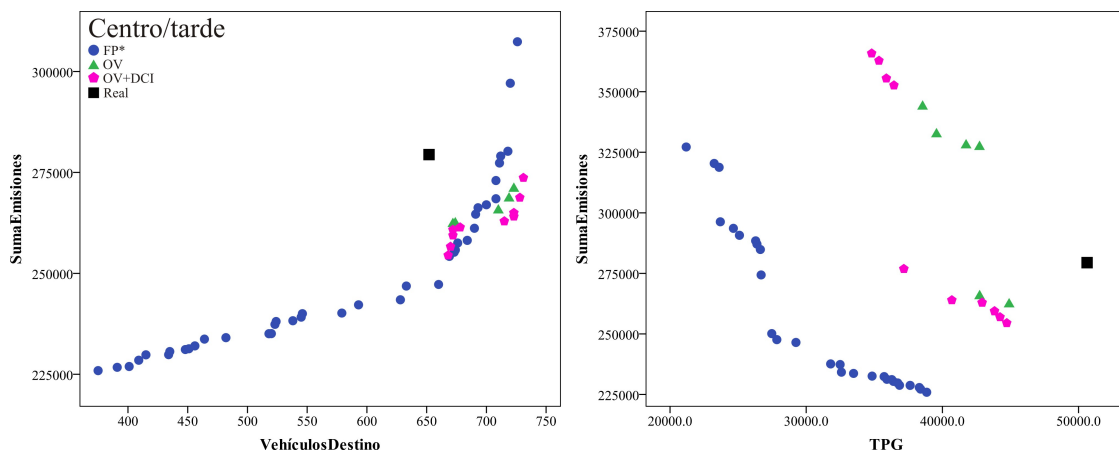


Figura 5.23: Frente de Pareto FP^* , de onda verde y de onda verde + DCI junto a la referencia real para la instancia Centro/tarde.

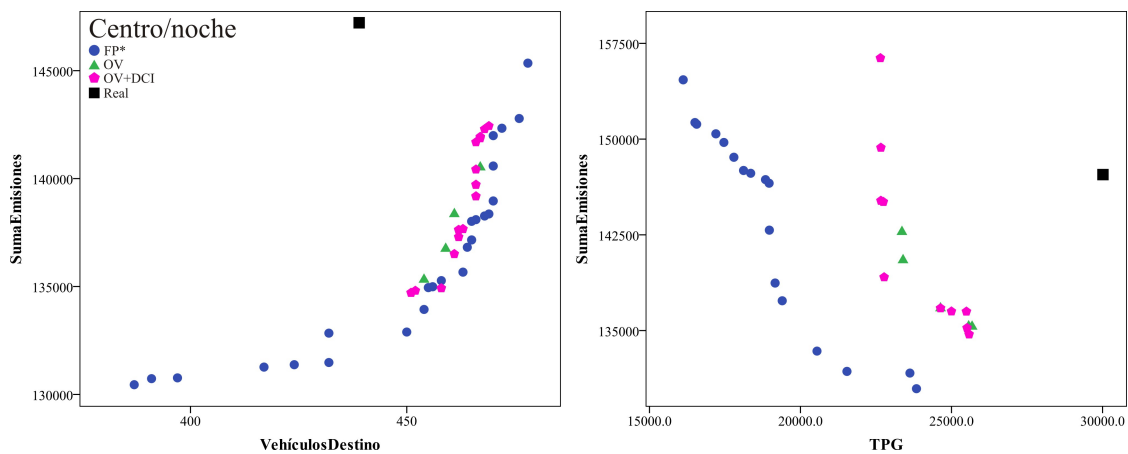


Figura 5.24: Frente de Pareto FP^* , de onda verde y de onda verde + DCI junto a la referencia real para la instancia Centro/noche.

onda verde + DCI - Centro									
obj.	horario mañana			horario tarde			horario noche		
	real	<i>max</i>	$\bar{x}$	real	<i>max</i>	$\bar{x}$	real	<i>max</i>	$\bar{x}$
CO	7396.3g	12.1 %	1.4 %	6369.5g	9.1 %	-9.4 %	3282.1g	8.5 %	2.8 %
CO ₂	304699.0g	6.0 %	-1.0 %	268274.7g	7.1 %	-3.7 %	144205.6g	5.3 %	0.3 %
HC	631.6g	18.7 %	1.9 %	532.1g	15.3 %	-13.8 %	268.2g	14.3 %	6.7 %
PM _x	53.9g	6.0 %	-2.3 %	46.6g	8.6 %	-4.8 %	24.8g	6.9 %	1.0 %
NO _x	1060.0g	5.0 %	-5.8 %	890.7g	13.0 %	-3.1 %	483.7g	10.3 %	0.7 %
VDG	633	16.0 %	-4.2 %	652	12.1 %	-12.0 %	439	6.8 %	2.5 %
TPG	57307.6s	34.8 %	11.8 %	50626.9s	31.2 %	14.8 %	30012.0s	24.5 %	15.4 %

Cuadro 5.10: Porcentaje de mejora de la técnica onda verde + DCI sobre la configuración real para la instancia Centro.

obj.	real	% mejora elegida
CO	7396.3g	8.6 %
CO ₂	304699.0g	-6.1 %
HC	631.6g	15.4 %
PM _x	53.9g	-5.3 %
NO _x	1060.0g	-17.9 %
VDG	633	19.1 %
TPG	57307.6s	1.9 %

Cuadro 5.11: Porcentaje de mejora de la configuración elegida sobre la real para la instancia Centro/mañana.

5.7.3. Análisis de una solución en el frente de Pareto

A continuación se presenta una solución elegida para el horario de la mañana. Esta elección se realiza en forma manual teniendo en cuenta los porcentajes analizados anteriormente, y en general se busca una mejora positiva para todos los objetivos. En el Cuadro 5.11 se reportan los porcentajes de mejora de la solución elegida sobre la configuración actual. En este se optó por priorizar los objetivos CO, HC, PM_x y vehículos en destino (es **19.1 %** mejor a la configuración real). En la Figura 5.25 se observa una zona del escenario al finalizar la simulación. Se debe notar que para la configuración elegida, no se tienen tantas colas de vehículos y las calles están menos congestionadas.

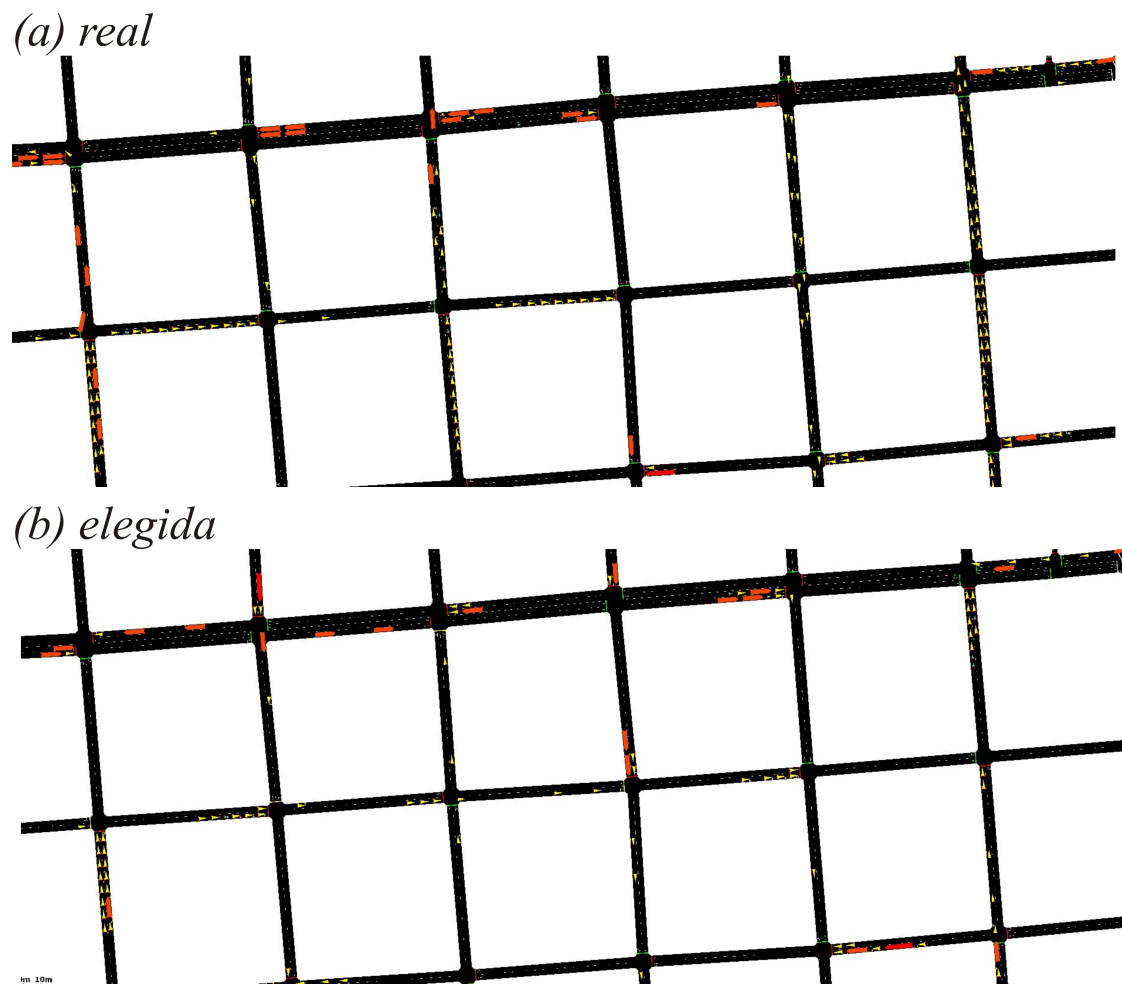


Figura 5.25: Captura del final de la simulación con (a) configuración real y (b) configuración elegida para la instancia Centro/tarde.

Capítulo 6

Conclusiones y trabajo futuro

6.1. Conclusiones

Este trabajo presenta una aproximación al problema de programación de ciclos de semáforos para reducción de emisiones vehiculares y mejora del flujo vehicular con la utilización de algoritmos evolutivos multiobjetivo.

Se propusieron cuatro algoritmos evolutivos, entre ellos los clásicos NSGA-II y GDE3, y los más modernos, MOEA/D y ϵ -NSGA-II. Estos fueron diseñados para proveer soluciones precisas al problema siguiendo casos de estudio realistas de la ciudad de Montevideo, Uruguay. Se propuso la reducción de las emisiones vehiculares (CO , CO_2 , HC , PM_x , NO_x) y, al mismo tiempo, mejorar las condiciones del tráfico buscando aumentar la cantidad de vehículos que alcanzan su destino al final de la simulación y disminuir la cantidad de tiempo global perdido por los vehículos como consecuencia de ir por debajo de la velocidad ideal.

Como casos de estudio se decidió analizar tres grandes zonas de la ciudad de Montevideo con distintivas características de tramado y tráfico vehicular. Los escenarios elegidos comprenden parte de los barrios Aguada, Jacinto Vera y Centro. Para cada escenario se consideraron además, tres horarios característicos. Esto permitió observar una variación del nivel de tráfico influyente en el comportamiento de los algoritmos. Para el estudio del tráfico de cada zona se desarrolló una estrategia conveniente de estimación de una matriz de tráfico, fuente de entrada de muchos estudios de transporte, mediante el uso de trazas de GPS.

En comparación con la configuración de semáforos actual y otras técnicas de planificación utilizadas, todos los algoritmos alcanzaron mejoras significativas, tanto a nivel de emisiones como en la mejora del tráfico vehicular en todas las instancias de evaluación. Para el escenario Aguada en el horario de la tarde, se obtuvieron mejoras de hasta un **77 %** en el tiempo global perdido por los vehículos y un **11.6 %** en la cantidad de vehículos que llegan a destino. A pesar de ser el horario con mayor volumen vehicular, se alcanzó una reducción máxima de **36.4 %**, **46.7 %**, **34.1 %** y **43.2 %** en las emisiones de CO , CO_2 , HC y PM_x respectivamente. En comparación con la técnica de onda verde, se observó una clara superioridad de los AEMO para todos los objetivos en todos los horarios. Para el escenario Jacinto Vera en el

mismo horario, el porcentaje máximo de mejora respecto a los vehículos que llegan a destino fue de **38.4 %** y se obtuvo una reducción máxima superior al **29.9 %** para todas las emisiones. Finalmente, para el escenario Centro (escenario con un total de 41 intersecciones semaforizadas) en el horario de la mañana, se obtuvieron mejoras de hasta un **43.6 %** para los vehículos que llegan a destino y de **19.3 %** en la reducción de CO y superior para las demás emisiones. Una solución particular elegida para este caso consigue una reducción de **8.6 %** en las emisiones de CO, **15.4 %** en las emisiones de HC y aumenta en un **19.1 %** los vehículos que llegan a destino. Al aplicar las técnicas alternativas de onda verde y descongestión manual, se obtuvieron resultados muy por debajo de los devueltos por los AEMO.

6.2. Trabajo futuro

Durante el desarrollo del proyecto surgieron varias tareas interesantes para ser tratadas como trabajo futuro. Estas tareas procuran mejorar la usabilidad de la metodología, simplificar su aplicación práctica, incrementar la precisión de los resultados e incorporar la contaminación acústica dentro de la optimización. Estas tareas se describen a continuación.

- *Integración de herramientas:* se plantea realizar una integración robusta de las herramientas desarrolladas, eliminando pasos intermedios manuales y determinando un módulo principal de operación. En esta misma línea, otra extensión a este trabajo es incorporar un algoritmo capaz de colaborar con la tarea de elección de la mejor solución y que el mismo pueda ser utilizado para adaptar la configuración de los semáforos *on the fly*.
- *Simplificación de la aplicación práctica:* si bien, en los experimentos realizados se consideraron configuraciones de semáforos distintas para horarios distintos, desde un punto de vista práctico, esto puede generar situaciones de inestabilidad o ser inaplicable debido a limitaciones de infraestructura. Por este motivo, sería apropiado estudiar configuraciones que optimicen simultáneamente todos los horarios considerados o considerar franjas horarias más amplias.
- *Incorporación de un modelo de dispersión atmosférica:* otra de las posibles mejoras es aplicar un modelo de dispersión atmosférica de contaminantes a los resultados obtenidos durante las simulaciones. Estos modelos se utilizan para predecir la concentración de contaminantes en un lugar determinado lejos de su fuente. Si bien, el uso de este tipo de modelo requeriría de un estudio exhaustivo de la arquitectura de las zonas, determinando áreas de interés para su aplicación, se podrían obtener resultados aun más precisos y realistas.
- *Considerar contaminación acústica:* por último, se propone considerar, además de los contaminantes atmosféricos, la contaminación acústica o sonora producida por los vehículos.

Apéndices

Apéndice A

Generación de las instancias de evaluación

Este apéndice especifica en detalle los pasos necesarios seguidos para la generación de las instancias de evaluación. Las redes en formato SUMO y el tráfico generado para las simulaciones desde este punto en adelante se nombrarán como red SUMO y tráfico SUMO respectivamente.

A.1. Generación de la red SUMO

Para correr una simulación en SUMO se necesitan archivos XML (eXtensible Markup Language) que definen el escenario de tráfico que queremos simular. Los archivos básicos para generar una simulación son: *red.net.xml* que define la red de calles, *red.rou.xml* que define los vehículos y sus trayectorias sobre la red, y un archivo de configuración *red.sumo.cfg* que define determinados parámetros como el tiempo de simulación, datos de entrada, salida, entre otros. Los pasos seguidos para generar los escenarios fueron los siguientes:

1. Construir la red de calles. Generalmente se utiliza la herramienta NETCONVERT del paquete SUMO, la cuál permite importar datos de desde varios formatos, siendo el mas popular la base de datos de *OpenStreetMap* (OSM - www.openstreetmap.org).
2. Validar y corregir la red de calles: Al convertir un mapa OSM a red SUMO, es muy probable que queden aspectos (giros no permitidos, número de carriles, conexiones entre carriles, programas de ciclos, etc) que no se corresponden con la realidad por lo que se deben corregir manualmente.
3. Generar el tráfico: Se definen la cantidad de vehículos y la información de cada vehículo como su tipo y su ruta. Las opciones dependen básicamente del tipo de información que se tiene disponible sobre el flujo vehicular.

En primer lugar, accedemos al sitio web de OSM y seleccionamos un área que contenga la red de tráfico que buscamos modelar como red SUMO. Luego descar-

gamos los datos a un archivo *red.osm*. Antes de convertir la red a formato SUMO, es conveniente utilizar el editor *JSOM*. Este programa nos permite, mediante una interfaz gráfica, modificar fácilmente el número de carriles de las calles, agregar reglas de giro, indicar que cruces se encuentran semaforizados, quitar calles extra, entre otras modificaciones propias de los datos OSM. Una vez que tenemos todos estos aspectos corregidos, generamos la red SUMO *red.net.xml* ejecutando:

```
\> netconvert --osm-files red.osm --geometry.remove
→ --roundabouts.guess --ramps.guess --junctions.join-dist
→ --tls.guess-signals --tls.discard-simple --tls.join -o
→ red.net.xml --plain-output-prefix
```

Esto nos devuelve la red SUMO *red.net.xml*. Se obtienen además los archivos individuales de configuración *true.con.xml*, *true.edg.xml*, *true.nod.xml* y *true.tll.xml* donde se especifican los arcos, conexiones entre carriles, nodos y programas de ciclos respectivamente. Debido a que se necesita tráfico SUMO para poder testear la red generada, podemos generar tráfico aleatorio ejecutando:

```
\> Sumo/tools/trip/randomTrips.py -n red.net.xml --fringe-factor 100
→ -l -L -e 480 -p 1 --trip-attributes="departLane=\"best\"
→ departSpeed=\"max\" departPos=\"random\" -r red.rou.xml
```

A continuación creamos un archivo de configuración que tenga como entradas: *red.net.xml* y *red.rou.xml* y un determinado tiempo de simulación (para conocer la estructura de los archivos, véase los esquemas XML accesibles desde la página web de SUMO). La herramienta SUMO-GUI nos permite visualizar la simulación correspondiente a los archivos generados donde podemos detectar errores de conexión entre carriles, programas de ciclos, etc. Estos errores se pueden corregir manualmente en los archivos individuales de configuración. Para terminar, estos archivos se combinan para generar una nueva *red.net.xml* corregida ejecutando:

```
\> netconvert --no-turnarounds -n true.nod.xml -e true.edg.xml -i
→ true.tll.xml -x true.con.xml -o red.net.xml
→ --ignore-errors.edge-type
```

Ahora que se tiene la red SUMO corregida, continuamos con la generación del tráfico SUMO deseado. El tráfico SUMO para las instancias de configuración fue generado en base a las proporciones de tipos de vehículo observadas en las instancias de evaluación. Para ello, se implementó la herramienta SearchAndReplace que se ejecuta del siguiente modo:

```
\> SearchAndReplace -n red.net.xml <tiempo de simulacion> <periodo
→ de arribo de vehiculos> <prop_LDV> <prop_PC> <prop_HDV>
→ <prop_Bus>
```

Para generar el tráfico de una instancia de evaluación se utilizan las herramientas de la sección siguiente.

A.2. Uso de las herramientas para el análisis de trazas de GPS

A.2.1. PowerGen

El primer paso en la generación de tráfico SUMO para una instancia de evaluación es la ejecución de PowerGen el cuál determina el valor ranking (ver Sección 4.5) para cada arco de la red de tráfico. Sin embargo, antes de ejecutar PowerGen se debe contar con un mapa ruteable de Montevideo. La herramienta OSM2Routing nos permite convertir un mapa OSM a formato ruteable:

```
\> OSM2Routing --osm=montevideo.osm --config=sample.config  
↪ --output=montevideo-routeable.osm
```

El programa PowerGen requiere además la red de tráfico *red.net.xml* y la selección de una franja horaria (ver Figura A.1). PowerGen devuelve los resultados en el archivo *red ranking.xml* donde se asigna un determinado ranking a cada arco de la red.

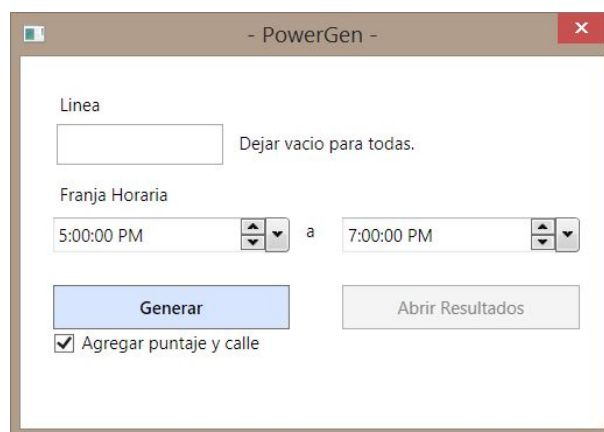


Figura A.1: Interfaz gráfica de PowerGen.

A.2.2. PriorWeight

Este programa, toma como entrada el resultado del PowerGen *red ranking.xml*, la red *red.net.xml* y devuelve el archivo *red.weight.xml* donde se especifica para cada arco un peso de prioridad y el conjunto mínimo de arcos que separan todos los pares OD:

```
\> PriorWeight -n red.net.xml -r red ranking.xml -o red.weight.xml
```

A partir de los recursos disponibles se realizan conteos en la mayor cantidad de arcos posible, comenzando por aquellos de mayor peso pertenecientes al conjunto mínimo y se ingresan en un archivo *red.conteos.xml* de la siguiente manera:

```
<mapa id="centro"/>
  <conteo id="1" edge="17906163">
    <light>8</light>
    <car>51</car>
    <heavy>1</heavy>
    <bus>1</bus>
  </conteo>
</mapa>
```

A.2.3. TrafficGenerator

El programa TrafficGenerator lee los conteos del archivo *red.conteos.xml* y genera el archivo de tráfico SUMO *red.rou.xml*. Además, devuelve el archivo *red.matrix.xml* con las matrices OD para cada tipo de vehículo:

```
\> TrafficGenerator -n red.net.xml -c red.conteos.xml -or
↪ red.rou.xml -om red.matrix.xml
```

Apéndice B

Ejecución de los AEMO

B.1. MOEA Framework

Para ejecutar un AEMO utilizando el framework se crea una instancia de la clase *Executor*:

```
Executor executor = new Executor()
    .withProblemClass(GenotipoEvaluator)
    .withAlgorithm(algorithm)
    .withProperties(props)
    .distributeOn(cores);
```

Donde, *GenotipoEvaluator* representa nuestro problema y define la evaluación de un individuo mediante la ejecución del simulador SUMO, *algorithm* es el nombre del algoritmo a ejecutar, *props* son los parámetros de configuración particulares del algoritmo y *cores* la cantidad de núcleos en los cuáles paralelizar el algoritmo (modelo maestro-esclavo).

Para realizar cada ejecución independiente ejecutamos:

```
NondominatedPopulation result = executor.run();
results.add(result);
```

El resultado *result* contiene la población del algoritmo al terminar la ejecución. Si el algoritmo elegido no soporta este modelo de paralelización (como, por ejemplo, MOEA/D), la ejecución será secuencial.

B.2. Ejecución en cluster FING

Para cada ejecución realizada en el Cluster se crea un *script.sh* donde configuramos todos los parámetros del trabajo y hacemos la llamada a nuestro programa. Se reserva el número de procesadores, la cola de ejecución apropiada, el directorio de trabajo y el correo electrónico para recibir el estado del trabajo. Después de definir la configuración del trabajo se ejecuta la tarea llamando al programa

moea.jar. A continuación se muestra un ejemplo de como ejecutar la tarea para una de las ejecuciones del NSGA-II:

```
$ time java -jar /home/pgtrafpol/NSGAII/moea.jar  
/home/pgtrafpol/0Escenarios/IC1/  
/home/pgtrafpol/0Resultados/Etapa1/IC1/  
E1_IC1_NSGAII_P1 NSGAII 10000 50 0.8 0.01
```

Como podemos ver, el ejecutable java *moea.jar* recibe varios parámetros. Algunos son necesarios para la ejecución correcta del programa, y otros dependen del algoritmo que se quiere utilizar. Primero le pasamos el directorio del escenario SUMO y el directorio raíz donde copiar los resultados de la ejecución. El tercer y cuarto argumento indica el nombre del archivo de configuración y el algoritmo a utilizar respectivamente. Luego se indica la cantidad máxima de evaluaciones y los parámetros de configuración del algoritmo. En este caso, la población elegida es de cincuenta individuos. Los últimos dos argumentos indican las tasas de mutación y cruzamiento.

Apéndice C

Resultados de la calibración paramétrica

Este apéndice reúne los resultados obtenidos para cada una de las 30 ejecuciones independientes de los AEMO NSGA-II, GDE3, MOEA/D y ϵ -NSGA-II, realizadas para cada instancia de configuración en las tres etapas de la calibración paramétrica.

C.1. Etapa 1

A continuación se muestran las medianas de las métricas utilizadas para la comparación entre configuraciones. Si una mediana se encuentra en **negrita** significa que los valores de la configuración obtenidos para esta métrica son los mejores. Si dos o más medianas se encuentran en **negrita**, entonces no existe diferencia estadísticamente significativa entre los valores de ambas configuraciones. Una mediana en *cursiva* indica que se trata del valor óptimo (mayor o menor, dependiendo de la métrica).

Para determinar cuál es la mejor configuración se utilizan las métricas HV, DG y DGI definidas en la Sección 2.6.1 y la contribución al frente de Pareto (CON). En primer lugar, se mira la métrica HV. En caso de que no haya diferencia estadísticamente significativa (empate) entre dos o más medianas, se utiliza la métrica DG. En caso de empate, se utiliza la métrica DGI y si hay empate nuevamente se utiliza la métrica CON. En caso de existir empate en todos los casos, se determina la configuración a partir de los valores óptimos (máximo o mínimo) siguiendo el mismo orden de decisión (HV, DG, DGI y CON). Para los AEMO NSGA-II, GDE3 y MOEA/D se eligió la configuración **C3**.

$\tilde{x}$	c.p. etapa 1 - NSGA-II								
	C1 - #P=50, mce=10000			C2 - #P=70, mce=15000			C3 - #P=100, mce=20000		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.130	0.218	0.090	0.157	0.270	0.100	0.173	0.268	0.123
DG	0.017	0.018	0.015	0.013	0.012	0.010	0.010	0.010	0.008
DGI	0.007	0.005	0.006	0.005	0.004	0.005	0.005	0.004	0.005
IEA	0.244	0.197	0.284	0.206	0.170	0.261	0.188	0.159	0.218
MEF	0.229	0.254	0.190	0.236	0.258	0.185	0.226	0.237	0.182
ESP	1.9E+5	2.1E+5	3.2E+5	1.5E+5	1.6E+5	2.6E+5	1.2E+5	1.3E+5	2.1E+5
CON	0.001	0.001	0.000	0.002	0.002	0.002	0.006	0.002	0.007
$t_e(s)$	204.0	253.0	270.0	254.5	329.5	306.5	325.0	418.5	445.0

Cuadro C.1: Resultados de la calibración paramétrica etapa 1, NSGA-II

$\tilde{x}$	c.p. etapa 1 - GDE3								
	C1 - #P=100, mce=10000			C2 - #P=150, mce=15000			C3 - #P=200, mce=20000		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.173	0.267	0.129	0.188	0.287	0.147	0.208	0.314	0.157
DG	0.011	0.009	0.010	0.008	0.007	0.007	0.006	0.005	0.006
DGI	0.005	0.004	0.004	0.004	0.004	0.004	0.004	0.003	0.003
IEA	0.178	0.154	0.186	0.167	0.140	0.179	0.147	0.117	0.158
MEF	0.232	0.216	0.195	0.211	0.204	0.184	0.196	0.178	0.181
ESP	1.2E+5	1.4E+5	2.2E+5	9.7E+4	1.1E+5	1.6E+5	8.7E+4	9.4E+4	1.3E+5
CON	0.003	0.003	0.002	0.004	0.006	0.005	0.012	0.013	0.009
$t_e(s)$	160.5	212.0	279.0	329.0	295.5	311.0	300.5	387.0	533.0

Cuadro C.2: Resultados de la calibración paramétrica etapa 1, GDE3

$\tilde{x}$	c.p. etapa 1 - MOEA/D								
	C1 - #P=100, mce=10000			C2 - #P=150, mce=15000			C3 - #P=200, mce=20000		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.061	0.156	0.048	0.065	0.175	0.065	0.100	0.189	0.075
DG	0.042	0.043	0.049	0.034	0.030	0.035	0.025	0.027	0.023
DGI	0.014	0.011	0.016	0.014	0.012	0.018	0.016	0.011	0.018
IEA	0.621	0.432	0.672	0.663	0.432	0.700	0.649	0.417	0.688
MEF	0.170	0.124	0.153	0.142	0.105	0.133	0.114	0.104	0.135
ESP	1.3E+6	1.2E+6	1.3E+6	8.7E+5	1.5E+6	1.4E+6	8.8E+5	1.5E+6	1.3E+6
CON	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_e(s)$	4347.0	4967.0	4858.5	6870.0	7337.0	7628.0	9791.0	9929.0	10338.0

Cuadro C.3: Resultados de la calibración paramétrica etapa 1, MOEA/D

C.2. Etapa 2

Continuando con el método de selección de la etapa anterior, para el NSGA-II se eligió la configuración **C9**, para el GDE3 la **C1** y para el MOEA/D la **C6**.

$\hat{x}$	c.p. etapa 2 - NSGA-II								
	C1 - $p_{SBX}=0.7, p_{PM}=0.005$			C2 - $p_{SBX}=0.7, p_{PM}=0.01$			C3 - $p_{SBX}=0.7, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.147	0.247	0.102	0.156	0.259	0.104	0.156	0.268	0.104
DG	0.011	0.011	0.009	0.011	0.010	0.009	0.012	0.010	0.010
DGI	0.004	0.004	0.005	0.004	0.004	0.005	0.004	0.004	0.005
IEA	0.213	0.191	0.249	0.215	0.191	0.252	0.208	0.185	0.257
MEF	0.231	0.217	0.168	0.248	0.207	0.175	0.250	<i>0.201</i>	0.186
ESP	1.2E+5	1.3E+5	2.1E+5	1.2E+5	1.3E+5	2.1E+5	1.3E+5	<i>1.3E+5</i>	2.1E+5
CON	0.001	0.001	0.000	0.001	0.001	0.000	0.001	0.001	0.000
$t_e(s)$	303.5	<i>381.0</i>	466.5	<i>278.0</i>	405.0	374.0	315.5	543.0	441.5
	C4 - $p_{SBX}=0.8, p_{PM}=0.005$			C5 - $p_{SBX}=0.8, p_{PM}=0.01$			C6 - $p_{SBX}=0.8, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.162	0.259	0.105	0.157	0.257	0.106	0.170	0.259	0.104
DG	0.011	0.010	0.009	0.011	0.010	0.009	0.011	0.010	0.010
DGI	0.004	0.004	0.005	0.004	0.004	0.004	0.004	0.004	0.005
IEA	0.203	0.191	0.250	0.214	0.192	0.234	<i>0.199</i>	0.194	0.258
MEF	0.239	0.209	0.161	0.249	0.201	0.169	0.242	0.220	0.171
ESP	1.2E+5	1.4E+5	<i>2.0E+5</i>	<i>1.2E+5</i>	1.3E+5	2.1E+5	1.3E+5	1.4E+5	2.1E+5
CON	0.001	0.001	0.001	<i>0.002</i>	0.001	<i>0.001</i>	0.001	0.001	0.001
$t_e(s)$	446.5	414.0	448.5	325.0	418.5	445.0	334.0	448.0	<i>372.0</i>
	C7 - $p_{SBX}=0.9, p_{PM}=0.005$			C8 - $p_{SBX}=0.9, p_{PM}=0.01$			C9 - $p_{SBX}=0.9, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.157	<i>0.269</i>	0.105	0.165	0.264	0.112	<i>0.175</i>	0.266	<i>0.112</i>
DG	0.011	0.010	0.009	0.010	0.010	<i>0.009</i>	<i>0.010</i>	<i>0.009</i>	0.009
DGI	0.004	0.004	0.005	<i>0.004</i>	0.004	0.004	0.004	<i>0.004</i>	<i>0.004</i>
IEA	0.213	<i>0.181</i>	0.257	0.200	0.192	0.236	0.206	0.185	<i>0.227</i>
MEF	0.242	0.226	<i>0.161</i>	<i>0.228</i>	0.214	0.163	0.229	0.224	0.186
ESP	1.2E+5	1.3E+5	2.2E+5	1.2E+5	1.3E+5	2.2E+5	1.3E+5	1.3E+5	2.3E+5
CON	0.001	0.001	0.001	0.001	0.001	0.001	<i>0.002</i>	<i>0.001</i>	0.001
$t_e(s)$	342.0	422.0	441.0	310.0	458.5	442.5	302.0	428.0	458.5

Cuadro C.4: Resultados de la calibración paramétrica etapa 2, NSGA-II

$\tilde{x}$	c.p. etapa 2 - GDE3								
	C1 - $p_{CR}=0.1, F=0.2$			C2 - $p_{CR}=0.1, F=0.4$			C3 - $p_{CR}=0.1, F=0.5$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.202	0.302	0.141	0.189	0.294	0.139	0.189	0.297	0.136
DG	0.006	0.005	0.005	0.007	0.005	0.006	0.007	0.005	0.006
DGI	0.003	0.003	0.004	0.003	0.003	0.003	0.003	0.003	0.003
IEA	0.163	0.159	0.216	0.173	0.163	0.192	0.170	0.153	0.184
MEF	0.190	0.165	0.153	0.210	0.153	0.174	0.208	0.161	0.185
ESP	8.0E+4	9.1E+4	1.3E+5	8.5E+4	9.4E+4	1.4E+5	8.7E+4	9.4E+4	1.3E+5
CON	0.004	0.004	0.004	0.002	0.003	0.002	0.002	0.003	0.001
$t_e(s)$	419.5	411.0	476.0	294.5	378.5	407.0	300.5	387.0	533.0
	C4 - $p_{CR}=0.2, F=0.2$			C5 - $p_{CR}=0.2, F=0.4$			C6 - $p_{CR}=0.2, F=0.5$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.187	0.288	0.136	0.178	0.281	0.131	0.175	0.283	0.131
DG	0.006	0.005	0.006	0.007	0.006	0.007	0.007	0.005	0.007
DGI	0.003	0.003	0.004	0.003	0.003	0.004	0.003	0.003	0.003
IEA	0.165	0.168	0.222	0.182	0.164	0.205	0.183	0.167	0.194
MEF	0.198	0.154	0.158	0.215	0.165	0.181	0.234	0.171	0.197
ESP	8.7E+4	8.8E+4	1.4E+5	8.8E+4	1.0E+5	1.4E+5	9.1E+4	1.0E+5	1.5E+5
CON	0.002	0.002	0.002	0.001	0.002	0.000	0.001	0.002	0.000
$t_e(s)$	316.5	388.0	448.5	299.0	460.5	412.0	292.0	375.0	429.5
	C7 - $p_{CR}=0.9, F=0.2$			C8 - $p_{CR}=0.9, F=0.4$			C9 - $p_{CR}=0.9, F=0.5$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.154	0.264	0.106	0.131	0.247	0.099	0.125	0.240	0.100
DG	0.008	0.006	0.007	0.009	0.007	0.008	0.010	0.007	0.009
DGI	0.004	0.003	0.004	0.004	0.004	0.004	0.004	0.003	0.004
IEA	0.207	0.181	0.234	0.233	0.180	0.238	0.230	0.195	0.237
MEF	0.246	0.194	0.199	0.272	0.209	0.241	0.286	0.217	0.251
ESP	8.8E+4	1.1E+5	1.3E+5	9.3E+4	1.2E+5	1.4E+5	1.0E+5	1.1E+5	1.4E+5
CON	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_e(s)$	569.0	389.5	407.5	294.0	491.0	412.5	417.0	426.0	429.0

Cuadro C.5: Resultados de la calibración paramétrica etapa 2, GDE3

$\tilde{x}$	c.p. etapa 2 - MOEA/D								
	C1 - $p_{CR}=0.1, p_{PM}=0.005$			C2 - $p_{CR}=0.1, p_{PM}=0.01$			C3 - $p_{CR}=0.1, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.091	0.182	0.059	0.094	0.178	0.066	0.113	0.194	0.083
DG	0.032	0.035	0.045	0.030	0.033	0.035	0.024	0.028	0.014
DGI	0.011	0.009	0.013	0.011	0.009	0.012	0.011	0.009	0.014
IEA	0.649	0.427	0.715	0.625	0.420	0.682	0.630	0.427	0.744
MEF	0.122	0.115	0.142	0.130	0.115	0.147	0.106	0.096	0.126
ESP	8.3E+5	1.7E+6	1.7E+6	8.8E+5	1.5E+6	1.3E+6	6.0E+5	1.6E+6	4.2E+5
CON	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_e(s)$	10435.5	10314.0	9719.0	9791.0	9929.0	10338.0	8973.0	9803.0	10281.0
	C4 - $p_{CR}=0.2, p_{PM}=0.005$			C5 - $p_{CR}=0.2, p_{PM}=0.01$			C6 - $p_{CR}=0.2, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.116	0.204	0.070	0.118	0.204	0.074	0.137	0.239	0.097
DG	0.022	0.026	0.035	0.026	0.027	0.029	0.019	0.021	0.007
DGI	0.013	0.009	0.013	0.011	0.009	0.014	0.012	0.010	0.012
IEA	0.652	0.427	0.749	0.654	0.422	0.774	0.633	0.420	0.652
MEF	0.100	0.106	0.146	0.121	0.109	0.135	0.092	0.087	0.125
ESP	6.3E+5	1.7E+6	1.3E+6	1.5E+6	1.9E+6	1.6E+6	8.8E+5	1.6E+6	8.9E+5
CON	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.001
$t_e(s)$	9640.0	9342.0	9286.0	10076.0	9299.0	9292.0	9793.0	9409.0	9350.0
	C7 - $p_{CR}=0.9, p_{PM}=0.005$			C8 - $p_{CR}=0.9, p_{PM}=0.01$			C9 - $p_{CR}=0.9, p_{PM}=0.05$		
mét	IC1	IC2	IC3	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.145	0.225	0.079	0.145	0.219	0.070	0.142	0.245	0.077
DG	0.009	0.022	0.015	0.010	0.024	0.015	0.009	0.009	0.016
DGI	0.010	0.009	0.011	0.010	0.008	0.011	0.010	0.008	0.011
IEA	0.601	0.398	0.652	0.606	0.379	0.675	0.596	0.369	0.668
MEF	0.125	0.095	0.185	0.129	0.103	0.185	0.141	0.114	0.183
ESP	6.4E+5	1.0E+6	7.0E+5	4.5E+5	1.3E+6	6.2E+5	3.9E+5	4.2E+5	5.6E+5
CON	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
$t_e(s)$	8878.0	9307.5	9152.0	8893.0	9213.0	9557.5	10337.0	9313.0	9217.0

Cuadro C.6: Resultados de la calibración paramétrica etapa 2, MOEA/D

C.3. Etapa 3

Para el AEMO ϵ -NSGA-II se configuraron $\#P$ y ϵ tomando los parámetros de cruzamiento y mutación determinados para el NSGA-II en la etapa 2. Se eligió la configuración **C1**.

$\hat{x}$	c.p. etapa 3 - ϵ -NSGA-II					
	C1 - $\#P=10, \epsilon=1$			C2 - $\#P=10, \epsilon=0.1$		
mét	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.155	0.270	0.098	0.152	0.268	0.090
DG	0.004	0.004	0.004	0.004	0.004	0.004
DGI	0.003	0.003	0.004	0.003	0.003	0.005
IEA	0.211	0.185	0.279	0.217	0.177	0.287
MEF	0.240	0.189	0.189	0.242	0.199	0.198
ESP	5.5E+4	7.2E+4	7.2E+4	5.7E+4	6.3E+4	6.8E+4
CON	0.001	0.002	0.000	0.000	0.001	0.000
$t_e(s)$	358.0	575.5	723.5	547.5	652.5	580.0
	C3 - $\#P=20, \epsilon=1$			C4 - $\#P=20, \epsilon=0.1$		
mét	IC1	IC2	IC3	IC1	IC2	IC3
HV	0.139	0.262	0.085	0.131	0.261	0.081
DG	0.004	0.004	0.004	0.004	0.004	0.004
DGI	0.004	0.003	0.005	0.004	0.003	0.005
IEA	0.243	0.190	0.303	0.238	0.189	0.314
MEF	0.260	0.189	0.188	0.254	0.202	0.195
ESP	5.9E+4	6.6E+4	6.8E+4	6.5E+4	6.5E+4	6.8E+4
CON	0.000	0.001	0.000	0.000	0.000	0.000
$t_e(s)$	566.5	685.5	781.5	443.5	748.5	847.0

Cuadro C.7: Resultados de la calibración paramétrica etapa 3, ϵ -NSGA-II

Apéndice D

Resultados de comparación entre los AEMO ejecutados

Este apéndice reúne los resultados obtenidos para cada una de las 30 ejecuciones independientes de los AEMO NSGA-II, GDE3, MOEA/D y ϵ -NSGA-II realizadas para cada instancia de evaluación.

D.1. Escenario Aguada

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.128	0.006	0.003	0.127	0.157	6.7E+5	0.000	882.0
	$\hat{x}$	0.165	0.008	0.003	0.158	0.192	8.2E+5	0.003	913.5
	<i>max</i>	0.212	0.010	0.003	0.192	0.335	1.5E+6	0.017	939.0
	σ	0.017	0.001	0.000	0.018	0.039	1.5E+5	0.004	12.6
GDE3	<i>min</i>	0.181	0.004	0.002	0.104	0.125	4.1E+5	0.005	828.0
	$\hat{x}$	0.190	0.005	0.002	0.129	0.153	5.3E+5	0.011	895.5
	<i>max</i>	0.217	0.006	0.003	0.163	0.273	9.2E+5	0.024	930.0
	σ	0.010	0.000	0.000	0.017	0.030	1.3E+5	0.005	30.8
MOEA/D	<i>min</i>	0.069	0.000	0.011	0.548	0.000	9.8E+4	0.000	16903.0
	$\hat{x}$	0.093	0.004	0.014	0.804	0.067	1.2E+6	0.002	17677.0
	<i>max</i>	0.141	0.010	0.016	0.855	0.195	3.9E+6	0.025	17678.0
	σ	0.018	0.002	0.001	0.079	0.044	1.1E+6	0.008	154.2
ϵ -NSGA-II	<i>min</i>	0.128	0.002	0.002	0.136	0.211	2.0E+5	0.000	1213.0
	$\hat{x}$	0.147	0.003	0.003	0.165	0.280	3.0E+5	0.004	1411.0
	<i>max</i>	0.179	0.004	0.004	0.225	0.528	6.1E+5	0.092	1806.0
	σ	0.014	0.001	0.000	0.020	0.067	9.9E+4	0.020	163.8

Cuadro D.1: Comparación entre los AEMO para la instancia Aguada/*mañana*

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.211	0.006	0.002	0.122	0.137	6.1E+5	0.000	750.0
	$\hat{x}$	0.245	0.008	0.003	0.152	0.195	7.2E+5	0.003	768.0
	<i>max</i>	0.285	0.010	0.003	0.191	0.280	8.8E+5	0.011	922.0
	σ	0.015	0.001	0.000	0.018	0.032	7.1E+4	0.003	39.6
GDE3	<i>min</i>	0.256	0.003	0.002	0.098	0.085	2.8E+5	0.002	815.0
	$\hat{x}$	0.295	0.004	0.002	0.114	0.123	4.1E+5	0.015	837.5
	<i>max</i>	0.322	0.006	0.004	0.175	0.219	1.3E+6	0.028	1142.0
	σ	0.013	0.001	0.001	0.022	0.028	2.0E+5	0.007	79.7
MOEA/D	<i>min</i>	0.092	0.001	0.010	0.692	0.028	1.0E+5	0.000	15575.0
	$\hat{x}$	0.145	0.005	0.013	0.763	0.076	1.7E+6	0.001	16376.5
	<i>max</i>	0.177	0.013	0.015	0.838	0.168	6.0E+6	0.023	16440.0
	σ	0.024	0.003	0.001	0.038	0.029	1.4E+6	0.005	323.4
ϵ -NSGA-II	<i>min</i>	0.206	0.002	0.002	0.114	0.116	2.2E+5	0.000	1389.0
	$\hat{x}$	0.246	0.003	0.002	0.132	0.182	3.1E+5	0.006	1781.5
	<i>max</i>	0.282	0.004	0.003	0.173	0.259	1.1E+6	0.049	2277.0
	σ	0.020	0.000	0.000	0.015	0.030	1.7E+5	0.011	210.9

Cuadro D.2: Comparación entre los AEMO para la instancia Aguada/*tarde*

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.168	0.006	0.004	0.136	0.142	2.3E+5	0.002	507.0
	$\hat{x}$	0.233	0.008	0.005	0.175	0.211	3.3E+5	0.005	520.0
	<i>max</i>	0.281	0.016	0.007	0.231	0.375	4.4E+5	0.025	537.0
	σ	0.024	0.002	0.001	0.023	0.058	5.0E+4	0.005	9.4
GDE3	<i>min</i>	0.232	0.003	0.003	0.115	0.109	1.6E+5	0.002	572.0
	$\hat{x}$	0.255	0.005	0.005	0.145	0.182	2.1E+5	0.013	581.0
	<i>max</i>	0.305	0.006	0.006	0.170	0.289	4.6E+5	0.073	678.0
	σ	0.018	0.001	0.001	0.017	0.042	8.0E+4	0.014	30.1
MOEA/D	<i>min</i>	0.099	0.002	0.010	0.428	0.093	2.1E+5	0.000	11823.0
	$\hat{x}$	0.127	0.008	0.011	0.481	0.141	9.0E+5	0.002	11900.0
	<i>max</i>	0.193	0.014	0.013	0.656	0.207	2.8E+6	0.044	12003.0
	σ	0.022	0.003	0.001	0.071	0.029	6.7E+5	0.012	55.1
ϵ -NSGA-II	<i>min</i>	0.172	0.003	0.004	0.116	0.146	9.3E+4	0.000	811.0
	$\hat{x}$	0.217	0.004	0.005	0.160	0.271	1.8E+5	0.002	1241.0
	<i>max</i>	0.261	0.006	0.006	0.212	0.402	3.8E+5	0.019	1957.0
	σ	0.024	0.001	0.001	0.021	0.063	6.0E+4	0.005	228.1

Cuadro D.3: Comparación entre los AEMO para la instancia Aguada/*noche*

D.2. Escenario Jacinto Vera

AEMO	med.	HV	GD	DGI	I _{ε+}	MEF	ESP	CON	t _e
NSGA-II	min	0.130	0.008	0.003	0.150	0.159	4.8E+5	0.000	1243.0
	$\hat{x}$	0.155	0.010	0.004	0.191	0.234	6.4E+5	0.003	1392.5
	max	0.201	0.015	0.004	0.282	0.390	9.8E+5	0.015	1661.0
	σ	0.018	0.002	0.000	0.030	0.059	1.0E+5	0.003	105.9
GDE3	min	0.184	0.003	0.003	0.116	0.110	3.5E+5	0.007	1312.0
	$\hat{x}$	0.202	0.005	0.003	0.150	0.168	4.1E+5	0.020	1443.0
	max	0.236	0.006	0.003	0.196	0.344	7.6E+5	0.054	1846.0
	σ	0.011	0.001	0.000	0.018	0.048	8.0E+4	0.010	148.7
MOEA/D	min	0.055	0.003	0.010	0.372	0.050	5.2E+5	0.000	21732.0
	$\hat{x}$	0.129	0.014	0.012	0.452	0.126	3.6E+6	0.003	28427.5
	max	0.167	0.068	0.015	0.630	0.282	1.5E+7	0.016	28679.0
	σ	0.030	0.015	0.001	0.072	0.055	4.8E+6	0.004	2731.6
ε-NSGA-II	min	0.100	0.003	0.003	0.138	0.203	1.7E+5	0.000	2207.0
	$\hat{x}$	0.139	0.004	0.004	0.197	0.265	2.5E+5	0.002	2598.0
	max	0.194	0.005	0.005	0.242	0.381	3.7E+5	0.058	3093.0
	σ	0.021	0.001	0.001	0.023	0.046	4.3E+4	0.011	206.3

Cuadro D.4: Comparación entre los AEMO para la instancia Jacinto Vera/*mañana*

AEMO	med.	HV	GD	DGI	I _{ε+}	MEF	ESP	CON	t _e
NSGA-II	min	0.137	0.008	0.003	0.140	0.144	5.6E+5	0.000	1054.0
	$\hat{x}$	0.182	0.010	0.003	0.195	0.206	6.6E+5	0.002	1138.0
	max	0.239	0.014	0.004	0.245	0.401	8.3E+5	0.007	1794.0
	σ	0.027	0.002	0.000	0.026	0.053	7.7E+4	0.002	166.9
GDE3	min	0.216	0.003	0.002	0.120	0.117	3.8E+5	0.006	1217.0
	$\hat{x}$	0.251	0.004	0.002	0.140	0.165	4.4E+5	0.018	1409.5
	max	0.269	0.006	0.002	0.176	0.259	6.6E+5	0.035	1874.0
	σ	0.014	0.001	0.000	0.014	0.033	7.0E+4	0.007	154.0
MOEA/D	min	0.034	0.000	0.010	0.516	0.000	7.0E+5	0.000	16846.0
	$\hat{x}$	0.096	0.020	0.013	0.672	0.138	4.1E+6	0.001	23264.5
	max	0.199	0.031	0.016	0.844	0.205	1.4E+7	0.003	23314.0
	σ	0.045	0.009	0.002	0.074	0.051	3.2E+6	0.001	2606.9
ε-NSGA-II	min	0.123	0.002	0.002	0.127	0.164	1.5E+5	0.001	1798.0
	$\hat{x}$	0.152	0.003	0.002	0.201	0.229	1.9E+5	0.008	2059.0
	max	0.219	0.004	0.003	0.245	0.369	2.6E+5	0.052	2540.0
	σ	0.025	0.000	0.000	0.029	0.056	3.1E+4	0.013	172.7

Cuadro D.5: Comparación entre los AEMO para la instancia Jacinto Vera/*tarde*

AEMO	med.	HV	GD	DGI	I _{ε+}	MEF	ESP	CON	t _e
NSGA-II	min	0.064	0.009	0.004	0.196	0.181	3.3E+5	0.000	1020.0
	$\hat{x}$	0.102	0.012	0.004	0.235	0.256	4.2E+5	0.003	1114.5
	max	0.133	0.017	0.006	0.291	0.406	7.5E+5	0.009	1396.0
	σ	0.017	0.002	0.000	0.029	0.045	9.6E+4	0.002	90.7
GDE3	min	0.098	0.004	0.003	0.179	0.153	2.1E+5	0.007	1020.0
	$\hat{x}$	0.122	0.006	0.003	0.219	0.193	2.8E+5	0.018	1131.5
	max	0.159	0.007	0.004	0.313	0.246	4.6E+5	0.041	1416.0
	σ	0.014	0.001	0.000	0.029	0.023	6.5E+4	0.008	74.9
MOEA/D	min	0.043	0.007	0.010	0.407	0.047	0.0E+0	0.000	14635.0
	$\hat{x}$	0.099	0.027	0.015	0.512	0.123	3.9E+6	0.002	14971.0
	max	0.148	0.097	0.018	0.709	0.281	1.8E+7	0.010	17903.0
	σ	0.027	0.019	0.002	0.075	0.049	4.4E+6	0.002	1552.1
ε-NSGA-II	min	0.051	0.003	0.004	0.260	0.207	1.2E+5	0.000	1402.0
	$\hat{x}$	0.077	0.004	0.004	0.315	0.258	1.5E+5	0.006	1674.5
	max	0.106	0.006	0.005	0.386	0.440	2.7E+5	0.040	2107.0
	σ	0.013	0.001	0.000	0.035	0.052	4.5E+4	0.009	148.8

Cuadro D.6: Comparación entre los AEMO para la instancia Jacinto Vera/*noche*

D.3. Escenario Centro

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.109	0.008	0.005	0.159	0.157	5.1E+5	0.000	1563.0
	$\hat{x}$	0.170	0.014	0.006	0.208	0.298	6.8E+5	0.002	1636.5
	<i>max</i>	0.209	0.018	0.007	0.260	0.372	9.5E+5	0.015	2255.0
	σ	0.022	0.002	0.001	0.024	0.059	1.0E+5	0.003	151.5
GDE3	<i>min</i>	0.198	0.003	0.004	0.116	0.109	2.4E+5	0.002	1312.0
	$\hat{x}$	0.249	0.005	0.005	0.158	0.159	3.6E+5	0.017	1370.5
	<i>max</i>	0.301	0.008	0.006	0.231	0.251	4.9E+5	0.066	1869.0
	σ	0.025	0.001	0.001	0.026	0.035	6.1E+4	0.017	134.2
MOEA/D	<i>min</i>	0.046	0.001	0.010	0.407	0.067	7.9E+4	0.000	16228.0
	$\hat{x}$	0.109	0.010	0.014	0.652	0.180	1.5E+6	0.000	38238.0
	<i>max</i>	0.191	0.023	0.017	0.836	0.374	7.2E+6	0.050	38900.0
	σ	0.032	0.005	0.002	0.102	0.071	1.8E+6	0.013	11094.9
ϵ -NSGA-II	<i>min</i>	0.097	0.003	0.005	0.193	0.193	1.8E+5	0.000	2017.0
	$\hat{x}$	0.157	0.006	0.006	0.232	0.251	2.8E+5	0.000	2297.5
	<i>max</i>	0.209	0.009	0.009	0.330	0.368	7.9E+5	0.027	2986.0
	σ	0.026	0.001	0.001	0.035	0.052	1.2E+5	0.005	219.6

Cuadro D.7: Comparación entre los AEMO para la instancia Centro/*mañana*

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.080	0.013	0.005	0.205	0.238	4.6E+5	0.000	2610.0
	$\hat{x}$	0.116	0.017	0.007	0.258	0.325	6.3E+5	0.003	3249.5
	<i>max</i>	0.158	0.023	0.009	0.311	0.472	1.0E+6	0.009	4005.0
	σ	0.020	0.003	0.001	0.026	0.052	1.2E+5	0.002	397.5
GDE3	<i>min</i>	0.128	0.002	0.004	0.156	0.114	2.3E+5	0.001	1226.0
	$\hat{x}$	0.169	0.009	0.005	0.192	0.258	3.8E+5	0.013	1309.5
	<i>max</i>	0.274	0.011	0.009	0.257	0.324	6.5E+5	0.145	1532.0
	σ	0.035	0.002	0.001	0.022	0.050	9.0E+4	0.029	75.4
MOEA/D	<i>min</i>	0.058	0.000	0.012	0.362	0.000	1.1E+5	0.000	31827.0
	$\hat{x}$	0.084	0.009	0.016	0.656	0.196	1.2E+6	0.000	31982.0
	<i>max</i>	0.157	0.023	0.020	0.812	0.357	5.0E+6	0.062	32828.0
	σ	0.025	0.005	0.003	0.115	0.091	1.3E+6	0.015	445.9
ϵ -NSGA-II	<i>min</i>	0.087	0.005	0.006	0.194	0.227	1.6E+5	0.000	1498.0
	$\hat{x}$	0.119	0.006	0.007	0.240	0.324	2.2E+5	0.000	1589.5
	<i>max</i>	0.160	0.008	0.009	0.310	0.398	5.1E+5	0.014	2093.0
	σ	0.017	0.001	0.001	0.031	0.045	8.2E+4	0.003	112.7

Cuadro D.8: Comparación entre los AEMO para la instancia Centro/*tarde*

AEMO	med.	HV	GD	DGI	$I_{\epsilon+}$	MEF	ESP	CON	t_e
NSGA-II	<i>min</i>	0.023	0.007	0.006	0.197	0.204	9.7E+4	0.000	821.0
	$\hat{x}$	0.076	0.024	0.009	0.302	0.446	1.7E+5	0.002	849.0
	<i>max</i>	0.184	0.040	0.013	0.377	0.724	3.3E+5	0.072	907.0
	σ	0.038	0.007	0.002	0.046	0.121	4.7E+4	0.017	23.6
GDE3	<i>min</i>	0.078	0.007	0.006	0.181	0.248	7.2E+4	0.000	703.0
	$\hat{x}$	0.126	0.011	0.008	0.279	0.352	1.5E+5	0.006	744.0
	<i>max</i>	0.181	0.014	0.010	0.416	0.635	4.6E+5	0.058	976.0
	σ	0.024	0.002	0.001	0.051	0.088	8.6E+4	0.017	64.9
MOEA/D	<i>min</i>	0.004	0.002	0.012	0.379	0.072	8.3E+4	0.000	10447.0
	$\hat{x}$	0.036	0.026	0.015	0.600	0.430	3.7E+5	0.000	24216.0
	<i>max</i>	0.093	0.062	0.022	0.771	0.794	2.1E+6	0.075	28211.0
	σ	0.021	0.014	0.002	0.077	0.147	4.9E+5	0.014	6972.8
ϵ -NSGA-II	<i>min</i>	0.017	0.006	0.010	0.311	0.255	5.0E+4	0.000	855.0
	$\hat{x}$	0.033	0.021	0.014	0.411	0.573	1.3E+5	0.000	946.0
	<i>max</i>	0.137	0.030	0.017	0.530	0.936	2.1E+5	0.034	1359.0
	σ	0.021	0.004	0.002	0.052	0.158	4.0E+4	0.006	126.8

Cuadro D.9: Comparación entre los AEMO para la instancia Centro/*noche*

Bibliografía

- E. Alba. *Análisis y diseño de algoritmos genéticos paralelos distribuidos*. PhD thesis, Universidad de Málaga, 1999.
- M. Amann. *Health risks of ozone from long-range transboundary air pollution*. WHO Regional Office Europe, Copenhagen, 2008.
- B. Asadi and A. Vahidi. Predictive cruise control: Utilizing upcoming traffic signal information for improving fuel economy and reducing trip time. *IEEE Transactions on Control Systems Technology*, 19(3):707–714, 2011.
- E. M. Ataz and Y. D. de Mera Morales. *Contaminación Atmosférica*, volume 45 of *Ciencia y Técnica*. Universidad de Castilla-La Mancha, Ciudad Real, España, 2004.
- R. Banos, C. Gil, B. Paechter, and J. Ortega. Parallelization of population-based multi-objective meta-heuristics: An empirical study. *Applied Mathematical Modelling*, 30(7):578–592, 2006.
- R. Baragona, F. Battaglia, and I. Poli. *Evolutionary Statistical Procedures: An Evolutionary Computation Approach to Statistical Procedures Designs and Applications*. Springer, Berlin, Heidelberg, 2011.
- J. Barceló. Models, traffic models, simulation, and traffic simulation. In *Fundamentals of Traffic Simulation*, volume 145, pages 1–62. Springer, New York, 2010.
- M. G. Bell and Y. Iida. *Transportation Network Analysis*. John Wiley & Sons, Chichester, 1997.
- M. G. Bell, C. M. Shield, F. Busch, and G. Kruse. A stochastic user equilibrium path flow estimator. *Transportation Research Part C: Emerging Technologies*, 5(3):197–210, 1997.
- S. Bera and K. V. K. Rao. Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport\Trasporti Europei*, 49:3–23, 2011.

- J. Branke, H. Schmeck, K. Deb, and M. Reddy. Parallelizing multi-objective evolutionary algorithms: Cone separation. In *Proceedings of the Congress on Evolutionary Computation*, volume 2, pages 1952–1957, 2004.
- J. Bremner, A. Frost, C. Haub, M. Mather, K. Ringheim, and E. Zuehlke. *World Population Highlights: Key Findings from PRB's 2010 World Population Data Sheet*, volume 65(2) of *Population Bulletin*. Population Reference Bureau, Washington D.C., 2010.
- E. Brockfeld, R. Barlovic, A. Schadschneider, and M. Schreckenberg. Optimizing traffic lights in a cellular automaton model for city traffic. *Physical Review E*, 64(5):056132, 2001.
- R. D. Brook, S. Rajagopalan, C. A. Pope, J. R. Brook, A. Bhatnagar, A. V. Diez-Roux, F. Holguin, Y. Hong, R. V. Luepker, M. A. Mittleman, et al. Particulate matter air pollution and cardiovascular disease an update to the scientific statement from the American Heart Association. *Circulation*, 121(21):2331–2378, 2010.
- G. D. Cameron and G. I. Duncan. PARAMICS-Parallel microscopic simulation of road traffic. *The Journal of Supercomputing*, 10(1):25–53, 1996.
- E. Cascetta and M. N. Postorino. Fixed point approaches to the estimation of O/D matrices using traffic counts on congested networks. *Transportation Science*, 35(2):134–147, 2001.
- P. S. Castro, D. Zhang, and S. Li. Urban traffic modelling and prediction using large scale taxi GPS traces. In *Pervasive Computing*, pages 57–72. Springer, Berlin, Heidelberg, 2012.
- B. Chapman, G. Jost, and R. Van Der Pas. *Using OpenMP: Portable Shared Memory Parallel Programming*, volume 10. MIT Press, Cambridge, MA, 2008.
- J. Chen and L. Xu. Road-junction traffic signal timing optimization by an adaptive particle swarm algorithm. In *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision*, pages 1–7, 2006.
- P. Chootinan, A. Chen, and W. Recker. Improved path flow estimator for origin-destination trip tables. *Transportation Research Record: Journal of the Transportation Research Board*, 1923:9–17, 2005.
- M. C. Coelho, T. L. Farias, and N. M. Rouphail. Impact of speed control traffic signals on pollutant emissions. *Transportation Research Part D: Transport and Environment*, 10(4):323–340, 2005.
- C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, Norwell, MA, 2nd edition, 2007.

- F. Costabile and I. Allegrini. A new approach to link transport emissions and air quality: An intelligent transport system based on the control of traffic air pollution. *Environmental Modelling and Software*, 23:258–267, 2008.
- H. Curtis and N. S. Barnes. *Biology*. Worth Publishers, New York, 5th edition, 1989.
- C. Darwin. *On the Origin of Species*. John Murray, London, 1859.
- B. K. David and W. H. Wen-mei. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann Publishers, Burlington, MA, 2nd edition, 2010.
- L. D. Davis, editor. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.
- B. De Coensel, A. Can, B. Degraeuwe, I. De Vlieger, and D. Botteldooren. Effects of traffic signal coordination on noise and air pollutant emissions. *Environmental Modelling & Software*, 35:74–83, 2012.
- K. A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1995.
- K. Deb and R. B. Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):1–15, 1994.
- K. Deb and M. Goyal. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and Informatics*, 26:30–45, 1996.
- K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- J. A. Díaz and J. E. G. Pantoja. *Ingeniería Ambiental*. Alfaomega Grupo Editor S.A, México, 2011.
- A. Ehlert, M. G. Bell, and S. Grosso. The optimisation of traffic count locations in road networks. *Transportation Research Part B: Methodological*, 40(6):460–479, 2006.
- A. E. Eiben and J. E. Smith. *Introduction to Evolutionary Computing*. Springer, Berlin, Heidelberg, 2007.
- A. Essabri, M. Gzara, and T. Loukil. Parallel multi-objective evolutionary algorithm with multi-front equitable distribution. In *Proceedings of the 5th International Conference on Grid and Cooperative Computing*, pages 241–244, 2006.
- J. Fenger. Urban air quality. *Atmospheric Environment*, 33(29):4877–4900, 1999.

- M. Ferrante, C. Ledda, G. O. Conti, M. Fiore, R. Fallico, and S. Sciacca. Old and new air pollutants: An evaluation on thirty years experiences. In *Air Pollution – A Comprehensive Perspective*, pages 3–26. InTech Open Access Publisher, Rijeka, 2012.
- C. Fisk. On combining maximum entropy trip matrix estimation with user optimal assignment. *Transportation Research Part B: Methodological*, 22(1):69–73, 1988.
- L. J. Fogel, A. J. Owens, and M. J. Walsh. *Artificial Intelligence through Simulated Evolution*. John Wiley & Sons, New York, 1966.
- C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*, pages 416–423, 1993.
- L. Gan, H. Yang, and S. C. Wong. Traffic counting location and error bound in origin-destination matrix estimation problems. *Journal of Transportation Engineering*, 131(7):524–534, 2005.
- N. J. Garber and L. A. Hoel. *Traffic and Highway Engineering*. Brooks/Cole Publishing, Pacific Grove, CA, 1999.
- J. M. García-Nieto, A. C. Olivera, and E. Alba. Optimal cycle program of traffic lights with particle swarm optimization. *IEEE Transactions On Evolutionary Computation*, 17(6):823–839, 2013.
- D. E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- D. Hadka. Moea framework version 2.4, 2015. URL <http://www.moeaframework.org/>. [En línea: 30-Junio-2015].
- A. Halati, H. Lieu, and S. Walker. Corsim-corridor traffic simulation model. In *Proceedings of the Conference on Traffic Congestion and Traffic Safety in the 21st Century: Challenges, Innovations, and Opportunities*, pages 570–576, 1997.
- M. L. Hazelton. Estimation of origin–destination matrices from link flows on uncongested networks. *Transportation Research Part B: Methodological*, 34(7):549–566, 2000.
- J. J. Higgins. *Introduction to Modern Nonparametric Statistics*. Duxbury Press, Pacific Grove, CA, 2003.
- M. K. Hill. *Understanding Environmental Pollution*. Cambridge University Press, Cambridge, UK, 2010.
- J. H. Holland. Outline for a logical theory of adaptive systems. *Journal of the Association for Computing Machinery*, 9(3):297–314, 1962a.

- J. H. Holland. Concerning efficient adaptive systems. In *Self-Organizing Systems 1962*, pages 215–230. Spartan Books, Washington D.C., 1962b.
- J. H. Holland. *Adaption in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, Michigan, 1975.
- S. Hoogendoorn and V. Knoop. Traffic flow theory and modelling. In *The Transport System and Transport Policy: An Introduction*, pages 125–159. Edward Elgar Publishing Ltd., Cheltenham, 2013.
- J. Horn, N. Nafpliotis, and D. E. Goldberg. A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the 1st IEEE Conference on Evolutionary Computation*, volume 1, pages 82–87, 1994.
- L. H. Immers and S. Logghe. *Traffic Flow Theory*. Faculty of Engineering, Department of Civil Engineering, Section Traffic and Infrastructure, Katholieke Universiteit Leuven, Heverlee, 2002.
- H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Proceedings of the 8th International Conference on Evolutionary Multi-Criterion Optimization*, pages 110–125, 2015.
- M. Z. Jacobson. *Air Pollution and Global Warming: History, Science, and Solutions*. Cambridge University Press, Cambridge, UK, 2012.
- L. Kabrt. Travel time analysis, 2010. URL <https://code.google.com/p/traveltimeanalysis/>. [En línea: 27-Octubre-2015].
- A. D. Kappos, P. Bruckmann, T. Eikmann, N. Englert, U. Heinrich, P. Höppe, E. Koch, G. H. Krause, W. G. Kreyling, K. Rauchfuss, et al. Health effects of particles in ambient air. *International Journal of Hygiene and Environmental Health*, 207(4):399–407, 2004.
- M. Keller and P. Wüthrich. Handbook emission factors for road transport (HBE-FA) 3.1 / 3.2: Quick reference. Technical report, INFRAS, 2014.
- B. Kelly. A ‘green wave’ reprieve. *Traffic Engineering & Control*, 53(2):55–58, 2012.
- W. D. Kelton and A. M. Law. *Simulation Modeling and Analysis*. McGraw-Hill, Boston, 3rd edition, 2000.
- G. A. Keoleian, K. Kar, M. M. Manion, and J. W. Bulkley. *Industrial Ecology of the Automobile: A Life Cycle Perspective*. Society of Automotive Engineers, Warrendale, PA, 1997.
- W. Knospe, L. Santen, A. Schadschneider, and M. Schreckenberg. Towards a realistic microscopic description of highway traffic. *Journal of Physics A: Mathematical and General*, 33(48):L477–L485, 2000.

- J. D. Knowles and D. W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- J. D. Knowles and D. W. Corne. On metrics for comparing nondominated sets. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 711–716, 2002.
- J. B. Kollat and P. M. Reed. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Advances in Water Resources*, 29(6):792–807, 2006.
- G. Kotushevski and K. A. Hawick. A review of traffic simulation software. Technical Report CSTN-095, Computer Science, Massey University, 2009.
- J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, 1992.
- D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner. SUMO (Simulation of Urban MObility). In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, pages 183–187, 2002.
- D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5:128–138, 2012.
- S. Krauß. *Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics*. PhD thesis, Universität zu Köln, 1998.
- S. Kukkonen and K. Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Proceedings of the Congress on Evolutionary Computation*, pages 1179–1186, 2006.
- S. Kukkonen and J. Lampinen. GDE3: The third evolution step of generalized differential evolution. In *Proceedings of the Congress on Evolutionary Computation*, volume 1, pages 443–450, 2005.
- S. Kukkonen and J. Lampinen. Performance assessment of generalized differential evolution 3 (GDE3) with a given set of problems. In *Proceedings of the Congress on Evolutionary Computation*, pages 3593–3600, 2007.
- J. Lampinen. DE’s selection rule for multiobjective optimization. Technical report, Department of Information Technology, Lappeenranta University of Technology, 2001.
- M. Laumanns, L. Thiele, K. Deb, and E. Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, 2002.

- H. Li and Q. Zhang. Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.
- M. Li, K. Boriboonsomsin, W. Zhang, and M. Barth. Traffic energy and emission reductions at signalized intersections: a study of the benefits of advanced driver information. *International Journal of ITS Research*, 7:49–58, 2009.
- Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-matching for low-sampling-rate GPS trajectories. In *Proceedings of the 17th International Conference on Advances in Geographic Information Systems*, pages 352–361, 2009.
- J. Lv, Y. Zhang, and J. Zietsman. Investigating emission reduction benefit from intersection signal optimization. *Journal of Intelligent Transportation Systems*, 17(3):200–209, 2013.
- J. Martikainen and S. J. Ovaska. Optimizing multiplicative general parameter finite impulse response filters using evolutionary computation. In *Success in Evolutionary Computation*, pages 327–354. Springer, Berlin, Heidelberg, 2008.
- J. McCrea and S. Moutari. A hybrid macroscopic-based model for traffic flow in road networks. *European Journal of Operational Research*, 207(2):676–684, 2010.
- G. Meyer, C. Reboul, P. Obert, and S. Tanguy. Carbon monoxide urban air pollution: cardiac effects. In *Advanced topics in environmental health and air pollution case studies*, pages 171–178. InTech Open Access Publisher, New York, 2011.
- Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer, Berlin, Heidelberg, 1992.
- J. R. Mihelcic and J. B. Zimmerman. *Ingeniería Ambiental: Fundamentos, Sustentabilidad, Diseño*. Alfaomega Grupo Editor S.A, México, 2012.
- M. J. Molina and L. T. Molina. Megacities and atmospheric pollution. *Journal of the Air & Waste Management Association*, 54(6):644–680, 2004.
- L. Mussone, S. Grant-Muller, and C. Haibo. A neural network approach to motorway OD matrix estimation from loop counts. *Journal of Transportation Systems Engineering and Information Technology*, 10(1):88–98, 2010.
- Navigant Research. Transportation forecast: Light duty vehicles - light duty stop-start, hybrid electric, plug-in hybrid electric, battery electric, natural gas, fuel cell, propane autogas, and conventional vehicles: Global market forecasts, 2015–2035. Technical report, Navigant Research, 2015.
- S. Nesmachnow. Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República. *Revista de la Asociación de Ingenieros del Uruguay*, 61:12–15, 2010.

- S. Nesmachnow. An overview of metaheuristics: accurate and efficient methods for optimisation. *International Journal of Metaheuristics*, 3(4):320–347, 2014.
- H. Nishuichi and T. Yoshii. A study of the signal control for the minimization of CO₂ emission. In *Proceedings of the 12th World Congress on Intelligent Transport Systems*, volume 6, pages 3375–3384, 2005.
- A. C. Olivera, J. M. García-Nieto, and E. Alba. Reducing vehicle emissions and fuel consumption in the city by using particle swarm optimization. *Applied Intelligence*, 42(3):389–405, 2015.
- B. Onursal and S. Gautam. *Vehicular Air Pollution: Experiences from Seven Latin American Urban Centers*, volume 373 of *World Bank Technical Paper*. World Bank Publications, Washington D.C., 1997.
- OpenStreetMap, 2015. URL <https://www.openstreetmap.org>. [En línea: 10-Abril-2016].
- J. A. Pardos. *Los ecosistemas forestales y el secuestro de carbono ante el calentamiento global*. Instituto Nacional de Investigación y Tecnología Agraria y Alimentaria, Madrid, 2010.
- L. Peng, M. Wang, J. Du, and G. Luo. Isolation niches particle swarm optimization applied to traffic lights controlling. In *Proceedings of the 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3318–3322, 2009.
- K. Praditwong and X. Yao. How well do multi-objective evolutionary algorithms scale to large problems. In *Proceedings of the Congress on Evolutionary Computation*, pages 3959–3966, 2007.
- K. Price, R. M. Storn, and J. A. Lampinen. *Differential Evolution: A Practical Approach to Global Optimization*. Springer, Berlin, Heidelberg, 2006.
- S. S. Rao. *Optimization, Theory and Applications*. Wiley Eastern Ltd., New Delhi, 1991.
- I. Rechenberg. *Evolutionstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart, 1973.
- C. R. Reeves, editor. *Modern Heuristic Techniques for Combinatorial Problems*. Blackwell Scientific Publications, Oxford, UK, 1993.
- D. I. Robertson, C. F. Lucas, and R. T. Baker. Coordinating traffic signals to reduce fuel consumption. Technical Report LR934, Transport Research Laboratory, 1980.
- N. M. Roupail, B. B. Park, and J. Sacks. Direct signal timing optimization: Strategy development and results. In *Proceedings of the 11th Pan American Conference in Traffic and Transportation Engineering*, pages 195–206, 2000.

- J. Sánchez, M. Galán, and E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas” area in Santa Cruz de Tenerife. *IEEE Transactions on Evolutionary Computation*, 12(1):25–40, 2008.
- M. V. Saraswathy and K. P. Isaac. Prioritisation of traffic count locations for trip matrix estimation using information theory. *International Journal for Traffic and Transport Engineering*, 3(2):112–126, 2013.
- J. D. Schaffer. *Some experiments in machine learning using vector evaluated genetic algorithms*. PhD thesis, Vanderbilt University, 1984.
- J. R. Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Master’s thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, 1995.
- H.-P. Schwefel. *Evolutionsstrategie und numerische Optimierung*. PhD thesis, Technische Universität Berlin, 1975.
- H.-P. Schwefel. *Numerische Optimierung von Computer Modellen mittels der Evolutionsstrategie*, volume 26 of *Interdisciplinary systems research*. Birkhäuser Verlag, Basel, 1977.
- P. Sharma and M. Khare. Modelling of vehicular exhausts—a review. *Transportation Research Part D: Transport and Environment*, 6(3):179–198, 2001.
- V. Shvetsov and D. Helbing. Macroscopic dynamics of multilane traffic. *Physical review E*, 59(6):6328, 1999.
- J. Sousanis. World vehicle population tops 1 billion units, 2011. URL http://wardsauto.com/ar/world_vehicle_population_110815. [En línea: 01-Octubre-2015].
- N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1995.
- A. Stevanovic, J. Stevanovic, K. Zhang, and S. Batterman. Optimizing traffic control to reduce fuel consumption and vehicular emissions - integrated approach with VISSIM, CMEM, and VISGAOST. *Transportation Research Record: Journal of the Transportation Research Board*, 2128:105–113, 2009.
- R. Storn. On the usage of differential evolution for function optimization. In *Proceedings of the Biennial Conference of the North American Fuzzy Information Processing Society*, pages 519–523, 1996.
- R. Storn and K. Price. Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces. Technical Report TR-95-012, International Computer Science Institute, 1995.

- F. Streichert, H. Ulmer, and A. Zell. Parallelization of multi-objective evolutionary algorithms using clustering algorithms. In *Evolutionary multi-criterion optimization*, pages 92–107, 2005.
- E.-G. Talbi and H. Meunier. Hierarchical parallel approach for GSM mobile network design. *Journal of Parallel and Distributed Computing*, 66(2):274–290, 2006.
- E.-G. Talbi, S. Mostaghim, T. Okabe, H. Ishibuchi, G. Rudolph, and C. A. Coello Coello. Parallel approaches for multiobjective optimization. In *Multi-objective Optimization*, pages 349–372. Springer, Berlin, Heidelberg, 2008.
- K. Tan. CEC 2007 conference report. *IEEE Computational Intelligence Magazine*, 3(2):72–73, 2008.
- F. Teklu, A. Sumalee, and D. Watling. A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering*, 22(1):31–43, 2007.
- T. Tielert, M. Killat, H. Hartenstein, R. Luz, S. Hausberger, and T. Benz. The impact of traffic-light-to-vehicle communication on fuel consumption and emissions. In *Proceedings of the Conference on Internet of Things*, pages 1–8, 2010.
- Transportation Research Board and National Research Council. *The Congestion Mitigation and Air Quality Improvement Program: Assessing 10 Years of Experience*, volume 264 of *Special Report*. National Academy Press, Washington D.C., 2002.
- A. M. Turkey, M. S. Ahmad, M. Z. Yusoff, and B. T. Hammad. Using genetic algorithm for traffic light control system with a pedestrian crossing. In *Proceedings of the 4th International Conference on Rough Sets and Knowledge Technology*, pages 512–519, 2009.
- A. Unal, N. M. Rouphail, and H. C. Frey. Effect of arterial signalization and level of service on measured vehicle emissions. *Transportation Research Record: Journal of the Transportation Research Board*, 1842:47–56, 2003.
- United Nations Centre for Human Settlements (Habitat). *Cities in a Globalizing World: Global Report on Human Settlements 2001*. Earthscan Publications Ltd., London, 2001.
- US EPA. What are the six common air pollutants?, 2015a. URL <http://www3.epa.gov/airquality/urbanair/>. Consulta: 01 octubre 2015.
- US EPA. National ambient air quality standards (NAAQS), 2015b. URL <http://www3.epa.gov/ttn/naaqs/criteria.html>. Consulta: 01 octubre 2015.

- D. A. Van Veldhuizen. *Multiobjective evolutionary algorithms: classifications, analyses, and new innovations*. PhD thesis, Department of Electrical and Computer Engineering Graduate School of Engineering, Air Force Institute of Technology, 1999.
- D. A. Van Veldhuizen, J. B. Zydallis, and G. B. Lamont. Considerations in engineering parallel multiobjective evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(2):144–173, 2003.
- D. Van Vliet. Saturn a modern assignment model. *Traffic Engineering and Control*, 23:578–581, 1982.
- H. J. Van Zuylen and L. G. Willumsen. The most likely trip matrix estimated from traffic counts. *Transportation Research Part B: Methodological*, 14(3):281–293, 1980.
- R. M. Velasco and P. Saavedra. Macroscopic models in traffic flow. *Qualitative Theory of Dynamical Systems*, 7(1):237–252, 2008.
- A. Warberg, J. Larsen, and R. M. Jørgensen. Green wave traffic optimization - a survey. Technical report, Department of Informatics and Mathematical Modelling, Technical University of Denmark, 2008.
- K. Wark and C. F. Warner. *Air Pollution: Its Origin and Control*. Harper and Row Publishers Inc., New York, 1981.
- S. Weart. *El Calentamiento Global - Historia de un Descubrimiento Científico*. Editorial Laetoli, Pamplona, España, 2006.
- D. Whitley. The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In *Proceedings of the 3rd International Conference on Genetic Algorithms*, pages 116–123, 1989.
- World Health Organization. 7 million premature deaths annually linked to air pollution, 2014. URL <http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/>. [En línea: 01-Octubre-2015].
- X. Wu, S. Deng, X. Du, and J. Ma. Green-wave traffic theory optimization and analysis. *World Journal of Engineering and Technology*, 2(3):14, 2014.
- S. Xiong and F. Li. Parallel strength Pareto multi-objective evolutionary algorithm for optimization problems. In *Proceedings of the Congress on Evolutionary Computation*, volume 4, pages 2712–2718, 2003.
- K. Xu, S. J. Louis, and R. C. Mancini. A scalable parallel genetic algorithm for X-ray spectroscopic analysis. In *Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation*, pages 811–816, 2005.

- H. Yang and J. Zhou. Optimal traffic counting locations for origin–destination matrix estimation. *Transportation Research Part B: Methodological*, 32(2):109–126, 1998.
- H. Yang, Y. Iida, and T. Sasaki. An analysis of the reliability of an origin–destination trip matrix estimated from traffic counts. *Transportation Research Part B: Methodological*, 25(5):351–363, 1991.
- H. Yang, T. Sasaki, Y. Iida, and Y. Asakura. Estimation of origin–destination matrices from link traffic counts on congested networks. *Transportation Research Part B: Methodological*, 26(6):417–434, 1992.
- H. Yang, L. Gan, and W. Tang. Determining cordons and screen lines for origin–destination trip studies. In *Proceedings of the 3rd Eastern Asia Society for Transportation Studies*, volume 3(2), pages 85–99, 2001.
- C. Zaror. *Introducción a la Ingeniería Ambiental para la Industria de Procesos*. Editorial Universidad de Concepción, Chile, 2002.
- Q. Zhang and H. Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.
- Q. Zhang, W. Liu, and H. Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *Proceedings of the Congress on Evolutionary Computation*, pages 203–208, 2009.
- Z. Zhou and M. Cai. Intersection signal control multi-objective optimization based on genetic algorithm. *Journal of Traffic and Transportation Engineering*, 1(2):153–158, 2014.
- E. Zitzler and S. Künzli. Indicator-based selection in multiobjective search. In *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature*, pages 832–842, 2004.
- E. Zitzler and L. Thiele. An evolutionary algorithm for multiobjective optimization: The strength Pareto approach. Technical Report 43, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology (ETH), 1998.
- E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.
- E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.
- E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2002.

- E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.