

UNIVERSIDAD DE LA REPUBLICA

PROYECTO DE GRADO
INGENIERIA EN COMPUTACIÓN

**Algoritmos evolutivos en
sincronización de semáforos en el
Corredor de Garzón**

Alvaro Acuña
Efrain Arreche
2014

Supervisor: Sergio Nesmachnow

CENTRO DE CALCULO - INSTITUTO DE COMPUTACIÓN
MONTEVIDEO, URUGUAY

Algoritmos evolutivos en sincronización de semáforos en el Corredor de Garzón

Acuña, Alvaro - Arreche, Efrain

Proyecto de Grado

Instituto de Computación - Facultad de Ingeniería

Universidad de la República

Montevideo, Uruguay, Marzo de 2015

ALGORITMOS EVOLUTIVOS EN SINCRONIZACIÓN DE SEMÁFOROS EN EL CORREDOR DE GARZÓN

RESUMEN

El proyecto propone el estudio de la sincronización de semáforos como problema de optimización multiobjetivo, y el diseño e implementación de un algoritmo evolutivo para resolverlo con alta eficacia numérica y desempeño computacional.

Se toma como aplicación la sincronización de semáforos en el Corredor de Garzón (Montevideo, Uruguay). La cantidad de cruces, calles, tráfico y semáforos lo hace un problema interesante desde el punto de vista de su complejidad. Además es admitido por las autoridades responsables los problemas relacionados con la sincronización de los semáforos por lo que todavía hay espacio para la mejora de los tiempos promedio de los viajes.

El problema de sincronización de semáforos es NP-difícil y no existe (hasta el momento) un método determinístico que lo resuelva, se buscará mediante un algoritmo evolutivo llegar a una configuración aceptable de los semáforos maximizando la velocidad media tanto de ómnibuses como de otros vehículos. El enfoque seguido es la obtención de datos reales relacionados a la red vial, tráfico y configuración de semáforos y utilizar el simulador de tráfico SUMO para generar los datos requeridos por el algoritmo.

El análisis experimental consiste en comparar los resultados del algoritmo con los valores obtenidos en la simulación de la realidad actual y además crear un escenario alternativo con modificaciones de la situación actual con el objetivo de obtener mejores métricas. Los resultados muestran que el algoritmo logra una mejora de hasta 24.2 % (21.40 % en promedio) el valor de fitness comparando con la realidad actual, mientras el escenario alternativo obtiene una mejora de hasta 37.1 % (34.7 % en promedio) en el valor de fitness.

Palabras clave: Corredor Garzón, simulación de tráfico, Algoritmo Evolutivo, Sincronización semáforos, escenario real, Cluster

Índice general

1. Introducción	1
1.1. Motivación y contexto	1
1.2. Objetivos	2
1.3. Enfoque	2
1.4. Limitaciones y alcance	2
1.5. Aportes	3
1.6. Estructura del documento	3
2. Marco Teórico	5
2.1. Problema del tránsito vehicular	5
2.2. Corredores Urbanos de tráfico	6
2.2.1. Corredor de Garzón	7
2.3. Sincronización de semáforos	8
2.4. Algoritmos Evolutivos	9
2.4.1. Algoritmos Genéticos	9
2.4.2. Funcionamiento	12
2.4.3. Algoritmos genético multiobjetivo	12
2.4.4. Algoritmo Evolutivo Paralelo	13
2.4.5. Resumen	14
2.5. Simuladores de tráfico	14
2.6. Trabajos relacionados	14
2.6.1. Resumen	18
3. Estrategia de resolución	19
3.1. Modelado del problema	19
3.1.1. Simulador SUMO (Simulation of Urban MObility)	19
3.1.2. Proceso del modelado	20
3.2. Trabajo de campo	21
3.3. Arquitectura de la solución	24
3.4. Biblioteca Malva	25
3.5. Especificación del Algoritmo Genético utilizado	26
3.5.1. Representación del cromosoma	26
3.5.2. Población inicial	27
3.5.3. Función fitness	28
3.5.4. Operadores	28

4. Análisis Experimental	31
4.1. Desarrollo y Plataforma de ejecución	31
4.2. Ajuste paramétrico	32
4.2.1. Pesos de la función fitness	32
4.2.2. Tiempo de simulación	33
4.2.3. Criterio de Parada	33
4.2.4. Tamaño de la población	34
4.2.5. Probabilidad de mutación y cruzamiento	34
4.3. Descripción de escenarios	36
4.3.1. Caso base o Realidad actual del corredor	36
4.3.2. Escenario Alternativo	36
4.4. Resultados	37
4.4.1. Valores numéricos del caso base	37
4.4.2. Resultados numéricos de la evaluación	37
4.4.3. Detalles del escenario alternativo	38
4.4.4. Resultados numéricos de la evaluación sobre el escenario alternativo	41
4.4.5. Variación de la función de fitness	41
4.4.6. Eficiencia computacional	43
5. Conclusiones y trabajo futuro	45
5.1. Conclusiones	45
5.2. Trabajo futuro	45
Bibliografía	46

Índice de figuras

2.1. Gráfica de costos de transporte en función de la gente transportada, BRT (Bus Rapid Transit), LRT(Light Rail Train)- Imagen original extraída de (Institute for Transportation & Development Policy (ITDP), 2007) . . .	7
2.2. La evolución se produce al mejorar el fitness de la población a lo largo de las generaciones.	9
2.3. Representación binaria de un cromosoma.	10
2.4. Cruzamiento de un punto	11
2.5. Mutacion por inversión binaria	11
2.6. Modelo Maestro-Esclavo	14
3.1. A la izquierda el mapa extraído de OSM, a la derecha el mapa procesado listo para ser usado en SUMO.	20
3.2. Ejemplo de planilla electrónica para el conteo manual en Camino Ariel . .	21
3.3. Mapa del TrafficModeler con las áreas de tráfico. Círculos del mismo color indican tráfico especificado entre esas áreas	23
3.4. Simulacion de tráfico en SUMO en el cruce entre Bulevar Battle y Ordoñez y el Corredor Garzon.	24
3.5. Arquitectura de la función de fitness	25
3.6. Cromosoma de 2 cruces	27
3.7. Representación de Sumo	27
3.8. Visualización del cruzamiento entre individuos.	29
4.1. Resumen representativo de ejecuciones del algoritmo para establecer el criterio de parada.	33
4.2. Gráfica con combinaciones de probabilidad de cruzamiento(pc) y de mutación (pm)	35
4.3. Comparación de la duración en minutos de los viajes para ómnibus y otros vehículos en el recorrido completo del corredor Garzón para los diferentes tipos de tráfico.	38
4.4. Gráfico de paradas alternativas. Gris: Parada Eliminada. Azul: linea G de Coectc y de Cutcsa. Rojo: linea G de Coectc. Violeta: G de Cutcsa. - Imagen original extraída de montevideo.gub.uy	39
4.5. Comparación de los speedup promedios para cada tipo de tráfico.	44

Índice de cuadros

3.1. Resumen del revelamiento del tráfico en la zona del corredor Garzón. Se muestra la cantidad de vehículos y la orientación hacia donde circulan. . .	22
4.1. Comparación de fitness para distintas poblaciones	34
4.2. Combinaciones de probabilidad de cruzamiento(pc) y de mutación (pm) .	35
4.3. Resultados del caso base mostrando la velocidad promedio ómnibus (vpb) y velocidad promedio vehículos(vpv) para los distintos tipos de tráfico . .	37
4.4. Resultados luego de ejecutado el algoritmo mostrando velocidad promedio ómnibus (vpb) y de otros vehículos(vpv) para los distintos tipos de tráfico	37
4.5. Valores del escenario alternativo con su velocidad promedio ómnibus (vpb) y velocidad promedio vehículos(vpv) comparando el fitness para el tráfico medio	40
4.6. Mejoras obtenidas para las velocidades promedio de los omnibus(vpb) y de otros vehiculos (vpv) en el escenario alternativo para distintos tipos de tráficos	41
4.7. Mejoras obtenidas al aplicar el algoritmo sobre el escenario alternativo. Comparando las velocidades de ómnibus(vpb), otros vehículos(vpv) y el fitness con cada tipo de tráfico contra el caso base o realidad actual. . . .	41
4.8. Modificación de los pesos para ómnibus (pb) y para otros vehículos (pv) en la función fitness. Analizando las variaciones en la velocidad promedio de ómnibus (vpb), otros vehículos (vpv) y fitness.	42
4.9. Análisis de eficiencia computacional comparando los tiempos de ejecución en serial y paralelo en minutos.	43

Índice de Algoritmos

1.	Algoritmo Genético	12
2.	Algoritmo Evolutivo MultiObjetivo. En rojo se indican las diferencias con el algoritmo evolutivo genérico.	13
3.	Algoritmo Genético de Malva.	26

Capítulo 1

Introducción

He llamado a este principio, por el cual cada pequeña variación, si útil, es preservada, con el término de Selección Natural

— Charles Darwin, El origen de las especies

En esta sección se pretende introducir al lector en el contexto general donde se desarrolla este trabajo así como los objetivos buscados.

1.1. Motivación y contexto

Los algoritmos evolutivos han demostrado su utilidad en problemas complejos y particularmente uno de ellos es la sincronización de semáforos. Se busca desarrollar un algoritmo que logre resolver este problema con buenas métricas. Por la flexibilidad inherente de este algoritmo es que no está destinado a resolver el problema en una zona en particular sino que se podría aplicar en forma general.

En este sentido se eligió la zona del corredor Garzón que presenta particularidades que la destacan y la hacen interesante desde el punto de vista de la investigación. Su complejidad viene dado por el largo del tramo, la cantidad de cruces, la complejidad y cantidad de semáforos en cada uno de ellos, las distintas reglas de tráfico aplicadas a cada tramo como por ejemplo exclusividad del ómnibus, o distinción para doblar a la izquierda, tráfico vehicular y transporte público, calles no paralelas, entre otros.

Por tanto al comprobar que el algoritmo obtiene un buen rendimiento en esta situación se puede tener confianza de que se comportaría adecuadamente en otras zonas que no presentan tanta complejidad.

1.2. Objetivos

Estos son los objetivos básicos que se plantearon al inicio del proyecto.

- Estudio del problema del tráfico y la sincronización de semáforos.
- Revelamiento de información sobre trabajos relacionados en este ámbito.
- Creación de un algoritmo evolutivo paralelo que resuelva el problema en la zona del corredor Garzón.
- Creación de un mapa y configuración relativa a semáforos, tráfico y reglas de tránsito que sea precisa y obtenida de la realidad actual.
- Aplicar técnicas de computación de alto desempeño para aumentar el rendimiento de la solución.

1.3. Enfoque

Desde un primero momento se intento dotar al proyecto de una buena aproximación de la realidad, en tal sentido se realizaron reuniones con el Ing. Juan Pablo Berta del Servicio de Ingeniería de Tránsito de la Intendencia de Montevideo en Agosto del 2014 y con el Ing. Daniel Muniz del departamento de Informática de la Intendencia en Setiembre de 2014 para conocer la situación del tráfico capitalino, aprender de su experiencia y obtener datos que nos fueran útiles para el proyecto.

Buscando una aproximación aun mas precisa se realizaron trabajos de campo para determinar la configuración de los semáforos, la densidad de tráfico y el tiempo de recorrida. El mapa y la frecuencia de ómnibus son de acceso publico así como el simulador utilizado.

Se creará un programa que implemente un algoritmo evolutivo multiobjetivo que utiliza un simulador de tráfico para obtener las métricas a optimizar. Se busca obtener una nueva configuración de semáforos que en las simulaciones se comporte mejor que la situación actual basándonos en la velocidad promedio de ómnibus y del resto de los vehículos.

Dada la complejidad del problema el algoritmo será paralelo y se utilizara la plataforma Cluster fing para poder acelerar el tiempo real de procesamiento. Además se realizará un escenario alternativo con modificaciones de la realidad actual con el objetivo de mejorar las métricas.

1.4. Limitaciones y alcance

La zona modelada comprende todo el tramo de el Corredor Garzón y dos caminos paralelos a ambos lados, que dada la configuración de las calles las cuales corren en diagonal fue un proceso complejo. El revelamiento de tráfico hecho in-situ fue realizado para un número determinado de calles que contiene a Garzón y cinco cruces representativos. Se busca una aproximación útil y no un estudio detallado sobre el tráfico en la zona.

Como lo que se pretende modelar es el tráfico vehicular y transporte publico no se realiza una simulación de peatones.

1.5. Aportes

- Se desarrollo un sitio web en la siguiente direccion: <http://www.fing.edu.uy/inco/grupos/cecal/hpc/AECG> donde los interesados podrán acceder para encontrar información sobre el proyecto y los resultados.
- Se realizó un *paper* en idioma ingles de 10 paginas con el objetivo de presentarlo en conferencias internacionales.
- El proyecto fue presentado en Ingeniera demuestra 2014, siendo bien recibido por el publico. Constatando de primera mano que la problemática es real y llegando a la conclusión que los Ingenieros tienen las herramientas necesarias para solucionar problemas que afectan directamente a la sociedad.

1.6. Estructura del documento

En el siguiente capitulo se hace un repaso sobre fundamentos teóricos necesarios para comprender el resto del trabajo. Se da información sobre corredores y en concreto el Corredor Garzón, así como un repaso breve sobre algoritmos evolutivos y simuladores de tráfico. Ademas se muestran los trabajo relacionados enfocando en algoritmos genéticos para la sincronización de semáforos. En el capítulo 3 se explica como se modela el problema y la arquitectura diseñada para resolverlo. Luego se comenta el trabajo de campo realizado para obtener datos de la realidad utilizados en la solución. Ademas se detalla el algoritmo genético utilizado así como la biblioteca y herramientas usadas. El capítulo 4 cuenta con la descripción de los escenarios, los resultados de la evaluación del algoritmo y las comparaciones realizadas. En este caso tenemos el escenario que representa la situación actual y también un escenario alternativo con modificaciones los cuales se evalúan con el algoritmo. Ademas se realizan pruebas para comprender como varia el algoritmo al modificar su función de fitness y se realiza un breve análisis de la eficiencia computacional del algoritmo. El capítulo 5 da las conclusiones finales y el trabajo a futuro que se puede realizar.

Capítulo 2

Marco Teórico

En este capítulo se aborda el marco teórico necesario para comprender más fácilmente el desarrollo de los capítulos posteriores. Se analiza el problema del tráfico en general y como se buscan soluciones como por ejemplo con la construcción de Corredores exclusivos para el transporte público. En este contexto se detalla el Corredor de Garzón tanto su descripción como sus problemas. Luego se da un breve repaso sobre los simuladores de tráfico y la teoría detrás de los algoritmos evolutivos. Para finalizar se encuentra una lista con trabajos relacionados para mostrar otras variantes y opciones.

2.1. Problema del tránsito vehicular

En gran parte del mundo se está produciendo un crecimiento sostenido del parque automotor lo que ocasiona una serie de problemas que afectan la calidad de vida de las personas relacionados con el agravamiento de las congestiones vehiculares (CEPAL, 2003).

Este problema tiene un gran impacto en el desarrollo de las ciudades por lo que es un componente principal en los planes estratégicos para su crecimiento.

La congestión ocasiona una progresiva merma en la velocidad promedio de circulación, lo que incrementa la duración de los viajes, aumenta el consumo de combustible y la contaminación atmosférica y sonora, lo que repercute directamente en la salud de las personas. además se genera una exigencia en las vías de tránsito que ocasiona un deterioro mayor de calles y rutas.

Uruguay no escapa a este fenómeno en particular Montevideo, donde el aumento del parque automotor está en ascenso constante desde el 2005 (INE, 2014) Y según proyecciones el crecimiento seguiría en un promedio de 4.5 % anual hasta el 2020. (BBVAResearch, 2013)

Esto viene de la mano con el sostenido aumento de las ventas de vehículos desde el 2003 (Autoanuario, 2014)

Los expertos indican que la congestión ya está instalada y la infraestructura vial no acompasó este crecimiento. además se indica que Montevideo es la ciudad con más semáforos por automóvil en Latinoamérica. Con más de 620 cruces semaforizados, alguno de los cuales no están coordinados.(Subrayado, 2013)

El crecimiento en la circulación de automóviles provoca que baje el nivel de aceptación del transporte público cuyo servicio en general es ineficiente. Para resolver este problema las autoridades suelen optar por sistemas de transporte costosos como los Metros pero

se ha demostrado que existen otras opciones viables como el BRT (Bus Rapid Transit / ómnibus de tránsito rápido). (Eugenio Bañobre, 2009)

En el caso de Montevideo para solucionar este problema se esta implementando el plan de movilidad urbana (IMM, 2010) con el objetivo de mejorar la eficiencia del transporte publico y democratizar el acceso al mismo. El sistema de transporte esta inspirado en un BRT con la construcción de varios corredores exclusivos en la ciudad. En la siguiente sección se dará mas información sobre los BRT, corredores urbanos y en particular el Corredor de Garzón.

2.2. Corredores Urbanos de tráfico

El corredor urbano de tráfico o también llamado corredor segregado se caracteriza por una separación física entre el carril de circulación de los autobuses y los carriles para el resto del tráfico y esta es la principal diferencia con un concepto similar llamado "carril de solo bus". Los carriles de solo bus se separan por líneas horizontales que son pintadas en la calle indicando que solo pueden circular autobuses por ese medio y los corredores segregados una separación en cemento que imposibilita el cambio de carriles. Los carriles de solo bus suelen fracasar por falta de control que evite que el tráfico los invada. En el caso de Montevideo mas allá de ser muy difícil de controlar que no se invada el carril, al hacer los carriles de solo bus sobre la derecha, es necesario que el transporte privado invada el carril al doblar a la derecha y que taxis lo invadan para levantar pasajeros por estar al lado de la vereda.

Los corredores segregados a nivel (sin hacer viaductos) cruzan las intersecciones mediante semáforos, lo que puede reducir mucho la capacidad del sistema. Los corredores segregados a distinto nivel evitan esos conflictos al construirse de forma que los carriles del autobús quedan completamente separados del resto del tráfico.

Las ciudades de países en vías de desarrollo deben estudiar si pueden instalar corredores pues son fundamentales para conseguir mayores velocidades en el sistema. Pero hay muchas posibilidades y no todas pueden ser aplicables por el contexto zonal, cultural o de inversión necesaria. Para intentar catalogar todos los corredores existentes, para poder saber que tan bueno puede llegar a ser un corredor en la fase de estudio del problema, la CUTR (Center for Urban Transportation Research) crea el National BRT Institute con la consigna de facilitar el intercambio de conocimientos y la innovación para aumentar la velocidad, la eficiencia y confiabilidad del servicio de autobuses. Se generan buenas practicas, se presentan estudios realizados en distintas zonas antes y despues de implementados los corredores, y se crea un standard que califica que tan bueno es un corredor BRT. <http://www.cutr.usf.edu/programs-1/national-bus-rapid-transit-institute/>

Bus Rapid Transit (BRT) es una solución innovadora, de alta capacidad y de menor costo para el transporte público que puede alcanzar el rendimiento y los beneficios de modos ferroviarios con un costo significativamente menor. Se trata de un sistema integrado de transporte basado en autobuses para el transporte de los pasajeros a sus destinos de manera rápida y eficiente. Al mismo tiempo ofrece la flexibilidad necesaria para satisfacer una variedad de condiciones locales. Los elementos del sistema de BRT pueden ser fácilmente personalizados a las necesidades de la comunidad e incorporan tecnologías de última generación de bajo costo que atraen a más pasajeros y en última instancia ayudan a reducir la congestión de tráfico en general.

Un BRT contiene características similares a un tren ligero o el sistema de metro por

ese motivo es mucho más confiable, conveniente y más rápido que los servicios regulares de autobús. Con las características adecuadas, el BRT es capaz de evitar las causas de los retrasos que suelen tener los servicios regulares de autobús, como estar atrapado en el tráfico y hacer cola para pagar a bordo.

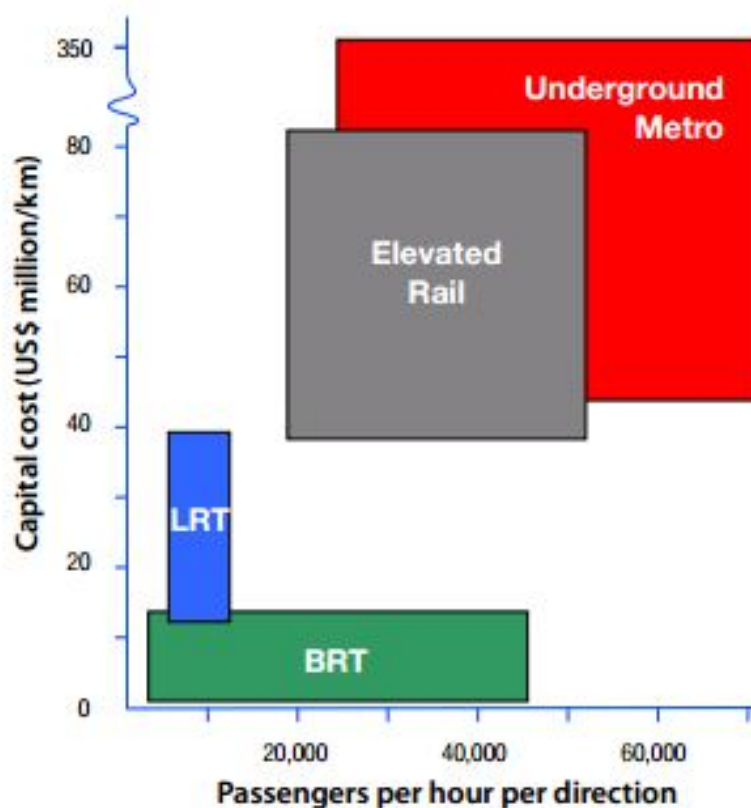


Figura 2.1: Gráfica de costos de transporte en función de la gente transportada, BRT (Bus Rapid Transit), LRT(Light Rail Train)- Imagen original extraída de (Institute for Transportation & Development Policy (ITDP), 2007)

2.2.1. Corredor de Garzón

COMPLETAR Estas referencias son las que usamos en el paper: (ElPais, 2013) (El-Pais, 2015)

Esta localizado al Nordeste de Montevideo Uruguay, fue construido como parte de un plan de movilidad que incluye otros 4 corredores de la ciudad. Tomando en cuenta de un extremo al otro conecta Colon con Paso Molino y teniendo 6.5km de largo, atraviesa muchas calles importantes como Millan p.e. que lleva a una autopista (Ruta 5). Es importante aclarar que no es solo una conexión de extremo a extremo ya que barrios como Sayago y otros constan de una gran cantidad de habitantes. Consiste básicamente en 3 calles paralelas e independientes; donde 2 de ellas son de dos carriles de una sola mano y entre medio de estas esta una calle doble vía con un carril para cada vía que es exclusivamente usado por ómnibus.

Se considera un BRT por motivos destacar puntos acumulados:

Recomendaciones del standard no aplicadas totalmente al corredor:

- **SEMÁFOROS.** Un corredor debe de funcionar al igual que una autopista en el sentido de que una vez que se entra debería ser posible mantener la velocidad máxima sin tener que parar seguido por lo que no deberían de haber semáforos cerca uno del otro, de haberlos la sincronización será la clave para minimizar los tiempos de espera.
- **CALLES PARALELAS.** Tener calles paralelas es de vital importancia para un corredor ya que una de las formas de minimizar los tiempos de espera es prohibir los giros a la izquierda, y una calle paralela provee la facilidad de poder realizarlo sin estorbar en el corredor.
- **LINEAS DE ÓMNIBUS.** Los corredores son realizados para mejorar tramos largos donde viaja unicamente una línea sola de transporte urbano.
- **GIRO A LA DERECHA CON LUZ ROJA.** Actualmente en muchos países se encuentra reglamentada una ley que permite a los conductores doblar a la derecha con luz roja ya que la misma (a menos que se especifique) es tomada como un cartel de Pare SOLAMENTE para doblar a la derecha. Esta ley acorta los tiempos de luz verde de las transversales mejorando así la velocidad promedio en el corredor.

Presenta un carril exclusivo para ómnibus y preferenciales <http://www.montevideo.gub.uy/ciudadania/stm-transporte-metropolitano/plan-de-movilidad/corredores>

Tiene 6km de largo , extendiéndose desde ... hasta ..

Agregar mapa y poner referencia de donde se saco

Los problemas de sincronización de semáforos fueron admitidos en varias publicaciones.

2.3. Sincronización de semáforos

—Esto va en Garzon, cuando decimos que tiene problemas: Por todo esto es relevante el tema de la sincronización de semáforos para agilizar el tránsito y no generar congestiones, aumentando la velocidad promedio de los viajes y mejorando las perspectivas de desarrollo de la ciudad así como la calidad de vida de sus habitantes.

Un buen funcionamiento de los semáforos es fundamental para asegurar que el tráfico se mueva con eficiencia y a la vez aporte seguridad a los peatones. Existen diversos métodos para lograr la coordinación necesaria que van desde simples mecanismos de reloj a sistemas computarizados que se ajustan en tiempo real con ayuda de sensores en la calle.

Cabe destacar que cuanto mayor cantidad de semáforos para sincronizar mayor es la dificultad a la hora de encontrar una solución eficiente. Por este motivo se considera a este prolijea como NP-Difícil, es decir que no existe hasta el momento un método determinístico que lo resuelva.

Existen diversas métodos para solucionar este problema, uno de los mas desarrollados y efectivos son los algoritmos evolutivos los cuales son usados en este trabajo y serán explicados a continuación.

2.4. Algoritmos Evolutivos

Uno de los puntos fundamentales del presente trabajo son los algoritmos evolutivos en particular los algoritmos genéticos por lo que se dedica esta sección para brindar un repaso por los conceptos y definiciones necesarias para comprender el desarrollo posterior de la solución.

Se dará un repaso general sin entrar en detalles, en el caso que el lector quiera profundizar se recomienda los trabajos de Goldberg (1989) y Mitchell (1996)

Los algoritmos evolutivos son métodos no determinísticos que se inspiran en la evolución natural de las especies utilizando conceptos como población, cruzamiento, mutación, selección, etc. Estos se utilizan para resolver problemas de optimización y búsqueda, entre otros (Nesmachnow, 2002).

Es una técnica iterativa que busca en cada paso mejorar las soluciones por medio de operadores basado en un criterio predefinido para maximizar o minimizar.

Este tipo de solución ha demostrado su utilidad en una amplia variedad de problemas complejos.

2.4.1. Algoritmos Genéticos

El algoritmo genético es uno de los más populares dentro de los algoritmos evolutivos.

La idea base es que partiendo de una población inicial de individuos se seleccionan los mejores en base a su aptitud respecto a solucionar el problema y estos se utilizan para generar nuevos individuos ya sea por combinación o modificación. Por tanto en cada paso obtenemos mejores soluciones hasta detenernos usando un criterio de parada ya sea el número de iteraciones o cuando ya no se puede mejorar más la solución.

Un individuo es una codificación de la solución que resuelve el problema. La población inicial puede generarse aleatoriamente o basándose en algún conocimiento previo. La función de evaluación o fitness indica que tan buena o apta es una solución en comparación con las demás. En cada iteración la cual se llama generación se aplican operadores de cruzamiento, estos son formas de combinar a los individuos para obtener otros que potencialmente sean una mejor solución y también cambios aleatorios sobre los individuos llamado mutación.

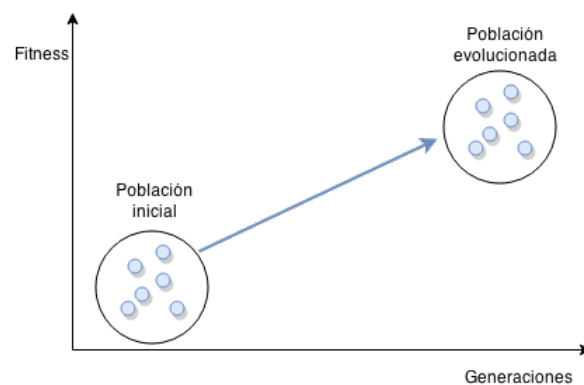


Figura 2.2: La evolución se produce al mejorar el fitness de la población a lo largo de las generaciones.

Por tanto se van seleccionando, combinando y cambiando las mejores soluciones en un proceso que va obteniendo mejores soluciones. El criterio de parada nos indica cuando termina este proceso, ya sea por que se alcanzó un número de generaciones predefinidos o por que la mejora no es evidente. Al final se devuelve la mejor solución encontrada en todo el proceso.

Hay que indicar que no es una técnica exacta pero logra muy buenas aproximaciones, ademas es muy buena en problemas complejos por su flexibilidad y robustez.

Representación de soluciones

No podemos trabajar directamente sobre las soluciones, por lo que tenemos que codificarlas en un modelo que nos sirva para poder aplicar el algoritmo. La inspiración biológica se ve en los nombres que adopta esta representación, llamada Cromosoma que es un vector de genes y cada valor de un gen se llama alelo. En general se codifica un vector de números binarios o reales de largo fijo, lo que facilita la aplicación de los operadores.

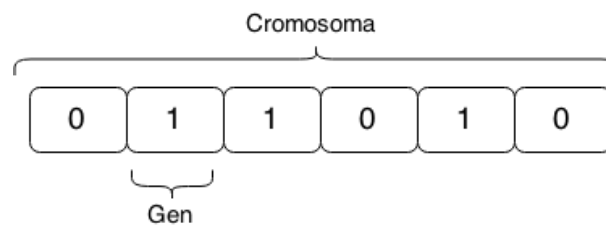


Figura 2.3: Representación binaria de un cromosoma.

Función de Evaluación

Indica que tan bueno es un individuo para resolver el problema en cuestión con un valor conocido como Fitness. Este se utiliza para seleccionar a los mejores y de esta forma guiar la exploración hacia la mejor solución. Se deben tener en cuenta las restricciones del problema para que las soluciones no factibles no sobrevivan. En general es donde se consume el mayor tiempo del algoritmo en comparación con los demás operadores.

Operador de Selección

Existen diversos operadores de selección, su función es que las mejores características de los individuos se mantengan en las siguientes generaciones. Los tipos más populares son:

- Ruleta: También conocida como selección proporcional elige aleatoriamente individuos en la cual la probabilidad de selección es proporcional al valor de fitness.
- Torneo: Se elige aleatoriamente un determinado número de individuos los cuales compiten entre si usando su valor de fitness.
- Rango: Se ordenan los individuos por fitness y se seleccionan los mejores.

Cruzamiento

Su función es combinar individuos con el objetivo de preservar las mejores características y así lograr mejores soluciones. Existe una tasa que se puede modificar para indicar la probabilidad de que se realice el cruzamiento.

- Cruzamiento de un punto: A partir de dos padres se selecciona un punto al azar de los cromosomas obteniendo dos trozos que se combinan para obtener dos hijos. Se explica en la figura 2.4
- Cruzamiento multipunto: El método anterior se puede generalizar para obtener más puntos de corte y más recombinaciones.

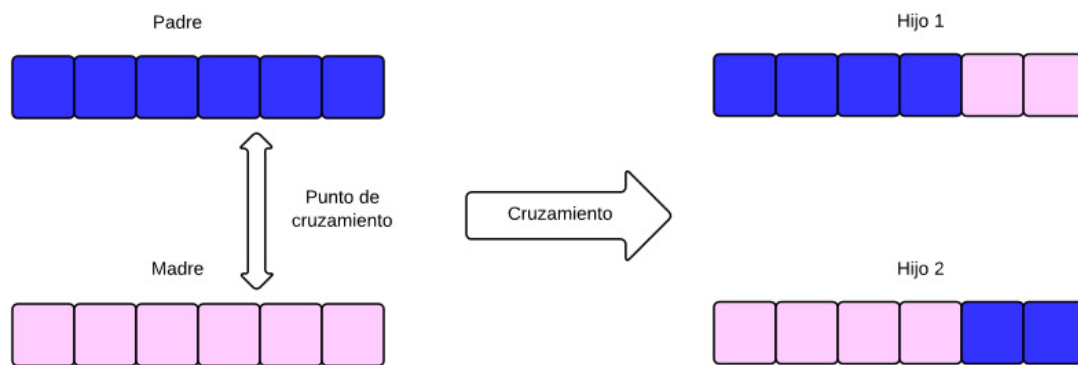


Figura 2.4: Cruzamiento de un punto

Mutación

Indica el método utilizado para modificar un individuo, esto se realiza para lograr más diversidad y no caer en máximos locales. En general aplica una modificación aleatoria en el cromosoma. Hay una tasa de probabilidad para aplicar este operador que en general es muy baja.



Figura 2.5: Mutacion por inversión binaria

Reemplazo

Luego de que se aplica el cruzamiento se insertan nuevos individuos que aumentan la población original por tanto este operador indica cual es el criterio que debemos tomar

para crear una nueva población con una cantidad de individuos igual a la original. Se podría reemplazar todos los padres por los hijos, o seleccionar cuales reemplazar, entre otros criterios.

Criterio de parada

Indica cuando debe terminar el algoritmo, puede ser definiendo un número fijo de generaciones o analizando si el mejor valor de fitness se mantiene relativamente constante durante un número determinado de generaciones entre otros criterios. El objetivo sera encontrar un compromiso entre un buen resultado y un tiempo acorde de ejecución, ya que el algoritmo no arrojará un valor exacto sino una buena aproximación.

2.4.2. Funcionamiento

El esquema básico de funcionamiento es el siguiente:

Algoritmo 1 Algoritmo Genético

```

1: Inicializo( Pob(0))
2: generacion = 0
3: while No llegue al criterio de parada do
4:   Evaluar Pob(generacion)
5:   Padres = Seleccionar(Pob(generacion))
6:   Hijos = Cruzamiento(Padres) y Mutacion(Padres)
7:   NuevaPob = Reemplazar Pob(generacion) con Hijos
8:   generacion++
9: end while
10: return Mejor solución

```

2.4.3. Algoritmos genético multiobjetivo

Los problemas de optimización multiobjetivo trabajan sobre un espacio multidimensional de funciones y no tienen una única solución por esto el significado de optimo cambia. Una solución es un optimo de Pareto si ninguna de las funciones objetivos puede mejorar su valor sin degradar otro de los valores objetivos. Todas las soluciones de Pareto son consideradas igualmente buenas ya que los vectores no se pueden ordenar completamente. Al conjunto de los valores funcionales de los optimos de Pareto se les llama frente de Pareto.

Existen algoritmos evolutivos para resolver el problema de la optimización multiobjetivo estos son los llamados MOEA por sus siglas en ingles *MultiObjective Evolutionary Algorithm*. Para un análisis mas detallado se recomienda el trabajo de Deb (2001).

Lo que buscan es aproximarse al frente de Pareto y lograr mostrar una gama de diferentes compromisos entre las funciones a optimizar para luego poder tomar la decisión de cual elegir.

Hay dos operadores propios de los MOEA estos son el operador de diversidad y el operador de asignación de fitness. El primero se aplica para evitar la convergencia a un sector en particular del frente de Pareto, el segundo intenta brindar mayor chance de perpetuar a los individuos con mejores características.

Se pueden clasificar por el método de asignación de fitness en:

Algoritmo 2 Algoritmo Evolutivo MultiObjetivo. En rojo se indican las diferencias con el algoritmo evolutivo genérico.

```

1: Inicializo( Pob(0))
2: generacion = 0
3: while No llegue al criterio de parada do
4:   Evaluar Pob(generacion)
5:   Operador Diversidad (Pob(generacion))
6:   Asignar Fitness (Pob(generacion))
7:   Padres = Seleccionar(Pob(generacion))
8:   Hijos = Cruzamiento(Padres) y Mutacion(Padres)
9:   NuevaPob = Reemplazar Pob(generacion) con Hijos
10:  generacion++
11: end while
12: return Frente de Pareto

```

- No basado en Pareto: Utilizan métodos sencillos de asignación de fitness. Son adecuados cuando el problema tiene no mas de tres funciones objetivo. Un mecanismo popular es la combinación lineal de los objetivos.

- Basado en Pareto: Utiliza explícitamente la dominancia de Pareto al asignar el fitness.

2.4.4. Algoritmo Evolutivo Paralelo

Los problemas complejos suelen requerir una alta demanda computacional por lo que paralelizar el algoritmo evolutivo es útil para lograr tiempos de ejecución menores, pero no es el único objetivo que se puede conseguir, entre otros tenemos: encontrar soluciones alternativas al mismo problema, búsqueda mas eficiente aun sin hardware paralelo, facilidad en la cooperación con otros métodos de búsqueda y búsqueda paralela de múltiples puntos en el espacio. (E. Alba, 2002).

Existen varios niveles de paralelización ya sea a nivel global enfocándonos en paralelizar la función fitness, a nivel de la población, o a nivel del individuo. (Nesmachnow, 2002)

En el caso de los algoritmos genéticos gran parte del tiempo se ocupa en la etapa de evaluación, por esta razón es un buen método para distribuir la carga en varios procesadores para que las evaluaciones se realicen en paralelo.

Un modelo muy utilizado es el Maestro-Esclavo. El proceso maestro es el encargado de realizar los operadores básicos del algoritmo y distribuir a procesos esclavos la evaluación de la función fitness para un conjunto de individuos, el esclavo devuelven el resultado y luego el maestro es el encargado de continuar ejecutando los operadores.

De este modo aumenta la eficiencia computacional del algoritmo ya que una de las funciones más costosas es distribuida entre varios nodos.

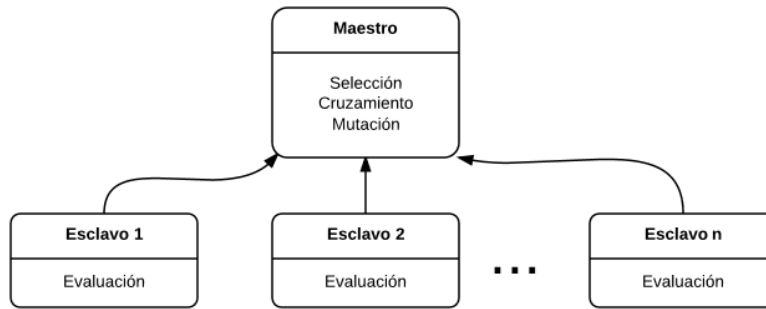


Figura 2.6: Modelo Maestro-Esclavo

2.4.5. Resumen

Los algoritmos evolutivos han demostrado su eficiencia en la resolución de gran cantidad de problemas de optimización. Si nos enfocamos en el problema de la sincronización de semáforos existen herramientas utilizadas para su resolución, entre las que se encuentran los simuladores de tráfico los cuales serán explicados en la siguiente sección.

2.5. Simuladores de tráfico

Los simuladores de tráfico son programas que simulan el movimiento de vehículos sobre una red de calles. Son usados en la investigación de tráfico vehicular, estudio de congestiones y análisis de impacto de obras. Existen varias razones para optar por esta herramienta, como es la rapidez en la obtención de resultados ya que la simulación se puede realizar en tiempos mucho más rápidos que en la realidad, el costo pues no es necesario cambiar la infraestructura para probar nuevos escenarios y sirve para prever situaciones que podrían darse bajo determinadas circunstancias.

Los simuladores se pueden dividir en microscópicos o macroscópicos según el nivel de detalle de la simulación. Un simulador macroscópico modela el tráfico vehicular como un fluido. En cambio un simulador microscópico simula el movimiento de cada vehículo según sus características particulares.

Cuanto más crece el número de vehículos y la complejidad de la red de mapas más difícil se hace crear la entrada que necesita el simulador. Aunque existen diversas herramientas que ayudan a este proceso aún se requiere un trabajo manual para el acondicionamiento de estos archivos.

SUMO es uno de los simuladores abiertos más populares. Es microscópico y utiliza una serie de archivos de configuración para representar las rutas, los vehículos y el tráfico.

2.6. Trabajos relacionados

La investigación del estado del arte se realizó con dos objetivos en mente, el primero analizar las distintas soluciones que existen actualmente para el problema y segundo encontrar nuevas prácticas, algoritmos o utilidades que pudieran fortalecer la solución.

El problema del tráfico optimizando las luces de los semáforos se puede resolver por diferentes métodos como redes neuronales (López et al., 1999), lógica difusa (G. Y. Lim y Hong, 2001), redes de petri (DiFebbraro et al., 2002), etc; por lo tanto la cantidad de

soluciones encontradas fue abundante y variada, por esto se decidió enfocarse en soluciones cercanas a la propuesta y en otras que tuvieran alguna particularidad interesante para destacar.

- J. Sánchez, M. Galán, y E. Rubio. Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. 2004

Este trabajo se basa en tres puntos: El uso de algoritmos genéticos para la optimización, simulación de autómatas celulares para la función de evaluación del tráfico y un cluster para realizar ejecuciones en paralelo. El modelo es pequeño con 5 calles de 2 vías que se intersectan. La codificación del cromosoma es un vector de números enteros, donde se codifica para cada intersección cual calle está habilitada en cada ciclo. Usa una estrategia de selección elitista donde los dos mejores se clonan a la siguiente generación y el resto es generado por cruzamiento de dos puntos.

Para la evaluación se utiliza el tiempo que transcurre desde el momento que un vehículo entra en la red hasta que sale. Se utilizó un cluster y programación paralela con una estrategia maestro-esclavo, el maestro envía los cromosomas a los esclavos que evalúan y devuelven el resultado, luego el maestro se encarga de generar la siguiente población.

Se compararon los resultados con una simulación aleatoria y con una fija, obteniendo la solución propuesta mejores resultados en todos los casos evaluados.

Este mismo grupo realizó trabajos similares expandiendo esta investigación, que se presentan a continuación.

- J. Sánchez, M. Galán, y E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas” Area in Santa Cruz de Tenerife. 2008 Lo interesante de este estudio es que se aplica lo expuesto en el trabajo anterior a un lugar real en Santa Cruz de Tenerife para validar los resultados. Algunas mejoras que se introdujeron fueron que el cromosoma se codifica utilizando código Gray lo que dicen mejora el rendimiento en mutación y cruzamiento. La población inicial esta compuesta por nueve “soluciones” provistas por la alcaldía de la ciudad. Tanto la estrategia de selección como de cruzamiento y mutación es similar al anterior trabajo.

El modelo se discretizó en 42 semáforos, 26 entradas y 20 salidas. Las soluciones provistas por la alcaldía se simularon y se utilizó para comparar con los resultados obtenidos por el algoritmo que en términos generales logra una mejora de hasta 26 %.

- J. Sánchez, M. Galán, y E. Rubio. Traffic signal optimization in “La Almozara” District in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. 2010 Este trabajo es similar al anterior pero se destacan algunos cambios, por ejemplo se probaron cuatro funciones de fitness diferentes, ellas fueron: cantidad de vehículos que llegaron a destino, tiempo de viaje promedio, tiempo de ocupación promedio y velocidad promedio global. También agrega medidas correspondientes al gas total emitido por los vehículos que tiene relación con la velocidad a la que van. El modelo discretizado de la zona de “La Almozara” cuenta con 17 semáforos, 7 intersecciones, 16 entradas y 18 salidas. Se simuló tanto un caso estándar como casos de alta congestión de tráfico, las

comparaciones se hacen respecto a las distintas funciones de fitness y los distintos escenarios planteados logrando buenos resultados.

- J. Penner, R. Hoar, y C. Jacob. Swarm-based traffic simulation with evolutionary traffic light adaptation. 2002 Este trabajo se centra en un modelo de simulación basado en enjambres que luego se optimiza utilizando un algoritmo genético cuya función de fitness es el tiempo promedio de los vehículos dentro de la red. El cromosoma cuenta con la secuencia y duración de los semáforos, así como la relación con los semáforos complementarios, la mutación tiene en cuenta esto para que no ocurra en una misma intersección dos luces verdes. El cruzamiento se hace entre los distintos semáforos con una probabilidad más alta si esta en la misma intersección.

El primer escenario es pequeño, cuenta con una ruta de 3 carriles y 3 intersecciones. Con diferentes variaciones logra mejoras significativas.

Luego se realiza otro escenario más complejo de 28 semáforos y 9 intersecciones logrando mejoras de hasta 26 %.

- D. H. Stolfi. Optimización del tráfico rodado en ciudades inteligentes. 2012 Este trabajo se basa en el concepto de una ciudad inteligente enfocado en la movilidad, indica que los atascos del tráfico provocan tanto pérdidas económicas como también contaminación ambiental.

Utiliza un algoritmo inteligente que tomando en cuenta el estado de congestión de las rutas sugiere al usuario cual es la ruta más rápida a su destino, utilizando un dispositivo en el automóvil que se enlaza por wifi con los semáforos que cuentan con sensores. Por lo tanto el trabajo no se basa en la optimización de las señales de los semáforos existentes sino agrega encima de esto un sistema de búsqueda de mejor ruta.

Para el modelo utiliza una zona de la ciudad de Málaga, cuenta con 8 entradas y 8 salidas, para la simulación utiliza SUMO. Los vehículos modelados son: turismo, monovolumen, furgoneta y camión donde se varia la longitud, velocidad y probabilidad que entre en la red de tráfico.

Se intenta minimizar los tiempo de viaje de los vehículos que circulan por la red. Para ellos se utiliza un algoritmo genético cuya estrategia de selección consiste en tomar los dos peores individuos reemplazándolos por los dos mejores hijos encontrados. En el cromosoma se representa cada sensor, con los destinos y rutas posibles. La función de fitness tiene en cuenta la cantidad de viajes completados durante el tiempo de ejecución, el tiempo medio utilizado y el retraso medio. Se prueban varias estrategias de cruzamiento y mutación. Las ejecuciones tienen un tiempo fijo de duración.

Compara el resultado con una simulación donde se generaron 64 itinerarios diferentes, esto se prueba en 3 escenarios. Las simulaciones se realiza hasta con 800 vehículos, concluyendo que al aumentar la cantidad de vehículos (más de 400) en el sistema la solución mejora sustancialmente el resultado base.

- K. T. K. Teo, W. Y. Kow, y Y. K. Chin. Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. 2010 Este trabajo presenta un modelo simple con una sola intersección en donde se intenta optimizar los tiempos de los semáforos para lograr mejorar rendimiento del tráfico. El cromosoma

representa los tiempos de la luces verdes mientras que la función de fitness es el largo de las colas generadas. Un aspecto interesante es que la simulación tiene un tiempo fijo de 600 segundos por generación pero no se detalla el tipo de simulación utilizada. Las conclusiones indican que la optimización usando algoritmos genéticos es buena para el problema del flujo de tráfico.

- D. J. Montana y S. Czerwinski. Evolving control laws for a network of traffic signals. 1996 Esta propuesta utiliza un enfoque adaptativo con sensores que analizan el tráfico en tiempo real. Un sensor para saber cuantos autos pasan y otro para saber que tan larga es la cola. Considera los cambios que se producen con respecto al caso promedio y cambia los tiempos de las señales en forma acorde. La premisa se basa en la inteligencia colectiva en donde agentes individuales realizan tareas simples que al interactuar producen resultados globales.

Se aplica programación genética más específicamente STGP (strongly typed genetic programming) (Montana, 1995) que aprende el árbol de decisión que sera ejecutado por todas las intersecciones cuando decida el cambio de fase. Además un algoritmo genético híbrido busca diferentes constantes que serán usadas en los arboles de decisión mejorando el flujo de tráfico.

La medida básica de efectividad en la función de evaluación es el *Delay*, esto es el total de tiempo perdido por causa de las señales de tráfico. Se probaron tres modelos distintos de cuatro intersecciones con una versión especial del simulador TRAF-NETSIM (Rathi, 1990)

El experimento arroja buenos resultados en cuando a la performance de la red y destaca la buena adaptabilidad en diferentes circunstancias. Aunque se marca el hecho de que el modelo es simple y de tamaño pequeño, siendo una incógnita como funcionara con problemas más complejos.

- A. Vogel, C. Goerick, y W. von Seelen. Evolutionary algorithms for optimizing traffic signal operation. 2000

La solución utiliza un enfoque auto-adaptable para mejorar el tráfico tanto en el corto como en el largo plazo a través de la optimización de las señales de tráfico en las intersecciones de una red de rutas. Al darle dinamismo a cada intersección se mejora el rendimiento de la red.

Destaca el hecho que dada una configuración de semáforos aún siendo optimizada usando simulaciones es difícil que sea la mejor en todas las situaciones o en casos extremos (horas picos). Para solucionar esto proponen un sistema auto-adaptable que toma la información del tráfico actual usando detectores de vehículos y espacios.

Utiliza el concepto de fases para representar las distintas posibilidades en la señalización de la intersección y cuanto tiempo debe permanecer en esa fase. Propone el desarrollo de un algoritmo evolutivo donde cada individuo representa un sistema de fases mientras el fitness se obtiene simulando ese sistema en un modelo de tráfico. Este es relativamente pequeño con una intersección con cuatro brazos, cada uno con tres líneas donde la ruta principal tiene el doble de densidad vehicular.

Los resultados indican que la ventaja de usar conocimiento experto para configurar los parámetros iniciales es mínimo ya que llega muy rápido a resultados similares. Tanto la búsqueda de los mejores parámetros como en estructuras más simples el algoritmo se comporta con buenos resultados.

- N. M. Rouphail, B. B. Park, y J. Sacks. Direct signal timing optimization: Strategy development and results. 2000

Se estudia una pequeña red de tráfico de nueve intersecciones con semáforos en la ciudad de Chicago, contando con tráfico de vehículos, parking, rutas de ómnibus y paradas. Se toman valores reales en horas pico, comprobando que las colas que se generan en la simulación coinciden con la realidad. Usa el programa (TRANSYT-7F) que permite visualizar mapas y contiene optimización de varios algoritmos genéticos y (CORSIM) un simulador de tráfico comercial. Se probaron 12 estrategias distintas de resolución midiendo el tiempo de demora en la red y el largo de las colas producidas. Los resultados indican que la performance de la red aumentó considerablemente usando este método.

2.6.1. Resumen

Aquí un breve repaso sobre los trabajos evaluados y su comparación con la propuesta presentada.

El trabajo de Sánchez et al. (2004) posee algunos puntos de contacto como es la ejecución paralela en un cluster y la arquitectura maestro-esclavo. La principal diferencia es que el escenario que evalúan es pequeño y no se compara con un escenario real.

El siguiente trabajo de Sánchez et al. (2008) expande lo anterior y lo utiliza en un caso real en Santa Cruz de Tenerife siendo de una complejidad similar a Garzón en términos de cantidad de cruces y semáforos. Los resultados obtenidos son muy positivos obteniendo mejoras de hasta 26 %.

Se destaca de Sánchez et al. (2010) las pruebas de diferentes funciones de fitness teniendo en cuenta diversos factores como tiempo de viaje o velocidad promedio. Este trabajo inspiró la realización de una función multiobjetivo que tuviera en cuenta la velocidad promedio en el proyecto actual.

Aunque Stolfi (2012) no optimiza la configuración de los semáforos, si plantea una posibilidad interesante para mejorar el tráfico en una ciudad indicando a los vehículos la mejor ruta por lo que se podría tomar como un elemento en trabajos futuros.

Tanto los trabajos de Teo et al. (2010) como Stolfi (2012) plantean la simulación con un tiempo fijo lo que se utilizó en el proyecto.

Montana y Czerwinski (1996) y Vogel et al. (2000) proponen algoritmos que se adapten en tiempo real por lo que se destacan como posibles trabajos a futuro. Es digno de mención que todos los trabajos consultados lograron mejoras significativas en sus resultados.

En conclusión el estudio de los trabajos relacionados permitió conocer en profundidad distintas soluciones y métodos que fueron tenidos en cuenta en menor o mayor medida en la solución propuesta. El hecho de que se obtuvieran buenos resultados motivo aun mas el desarrollo del trabajo presentado.

Capítulo 3

Estrategia de resolución

En este capítulo se explica el proceso llevado a cabo para desarrollar la solución. Se presenta el modelado básico del problema que consiste en un mapa de la zona con todos los datos necesarios para realizar la simulación y el algoritmo genético utilizado con su estructura, parámetros y configuración.

Se detalla la biblioteca utilizada para el desarrollo del algoritmo y la arquitectura desarrollada.

3.1. Modelado del problema

Para solucionar el problema se usaran simulaciones de la realidad, por tanto es importante contar con un modelo detallado y fidedigno. A continuación se da información sobre el simulador a utilizar, las herramientas necesarias y el proceso seguido para generar el modelo.

3.1.1. Simulador SUMO (Simulation of Urban MObility)

Es un simulador de tráfico gratis y abierto que nos permite modelar redes de calles, vehículos, transporte público, ciudadanos y semáforos (SUMO). Sigue un modelo microscópico ya que realiza la simulación individual explícita de cada elemento. Incluye un conjunto de herramientas destinadas a facilitar la generación de tráfico y la construcción de mapas.

Entre las razones para elegir este simulador se encuentra que es de código abierto sin costo comercial, con una detallada documentación y buen ritmo de actualizaciones. Puede ser ejecutado tanto en linux como windows. Permite las opciones de ejecutarse sin interfaz gráfica utilizando la línea de comando lo que aumenta sensiblemente la velocidad de ejecución y con la interfaz que sirve para mostrar exactamente como se comporta el tráfico.

Tiene un funcionamiento sencillo basado en tomar como entradas archivos de configuración que representan la red vial, los vehículos, el tráfico y los semáforos. Y archivos de salida con información interesante como el tiempo de simulación, la cantidad de vehículos, sus velocidades, duración del viaje, emisiones de gases contaminantes, etc.

Algunas otras herramientas relacionadas a la simulación:

- NetConvert: Utilizado para la generación de la red a partir de un mapa. Por ejemplo podemos transformar un mapa de Open Street Map en archivo XML de red que

SUMO reconoce. Este programa viene integrado con SUMO.

- DUaRouter: Genera recorridos de vehículos basado en dinámicas definidas por el usuario. Esta utilidad viene integrada con SUMO.
- Traffic Modeler: Herramienta para la generación de tráfico de manera visual y luego exportarlo para que sea reconocido por SUMO. (Papaleontiou)

3.1.2. Proceso del modelado

Se describen los pasos realizados para tener los elementos necesarios para realizar la simulación.

Diseño del mapa

El primer paso consiste en desarrollar un mapa de la zona que sea compatible con el simulador. Para ello se utiliza el servicio de Open Street Map (OSM) que cuenta con mapas libres y actualizados por una comunidad muy activa. Luego para cotejar su exactitud se comparo con otros servicios como Google Maps y Bing Maps.

Se exporta el mapa al editor JOSM para poder adecuarlo a las necesidades del problema. El alcance que se pretende es la zona que corresponde al Corredor Garzon desde su inicio hasta el final y dos caminos paralelas a cada lado. Como se ve en el mapa no existen calles paralelas reales por lo que fue una tarea dificultosa en este aspecto. Ademas se eliminaron elementos del mapa que no eran necesarios para nuestro modelo.

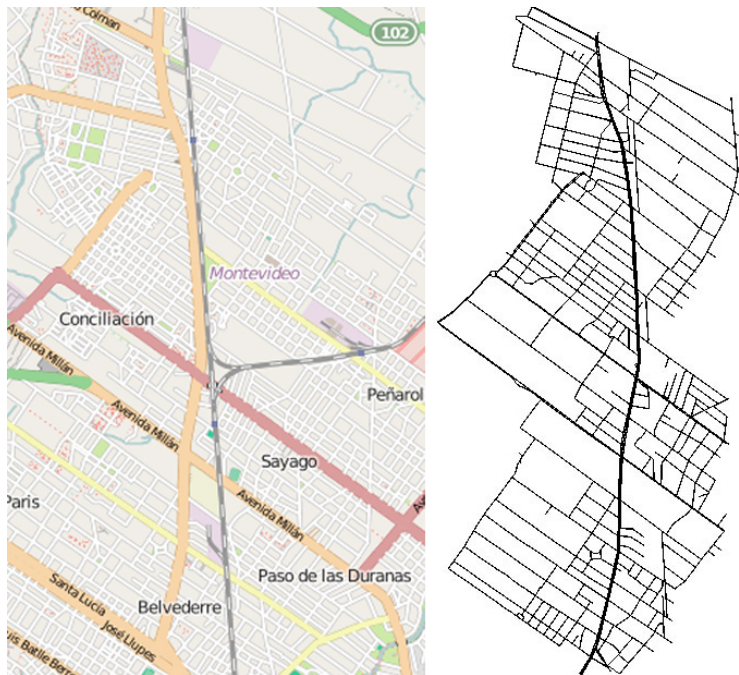


Figura 3.1: A la izquierda el mapa extraído de OSM, a la derecha el mapa procesado listo para ser usado en SUMO.

Luego se utilizó la herramienta Netconvert para transformarlo al formato que SUMO acepta y se realizaron varios ajustes editando los archivos generados para corregir errores

en las calles, cruces y conexiones. De esta manera se obtuvo un mapa fiel a la realidad y compatible con el simulador.

3.2. Trabajo de campo

Se cuentan con los datos públicos sobre la cantidad de ómnibus y sus frecuencias pero no se tienen datos sobre la densidad de tráfico vehicular en la zona, por esto se realiza un revelamiento in-situ con las siguientes características.

Se seleccionaron cinco cruces que presentan diferentes características para poder modelar variantes en la cantidad de vehiculos circulantes. Estos son: Camino Ariel, Battle y Ordoñez, Plaza Videla, Camino Colman y Aparicio Saravia.

Se siguen las recomendaciones de los textos consultados al respecto (CTRE, 2002). Se eligió el día miércoles, que estuviera soleado y entre las 15 y 17 horas para no tener los sesgos que se producen los fines de semana o días de lluvia. Se realizaron filmaciones de entre 15 y 30 minutos en los cruces y luego se analizaron los vídeos para realizar el conteo manual con la posibilidad de enlentecer el vídeo para mayor facilidad. Luego se completa una planilla electrónica donde se tiene la información de vehículos que recorren Garzón, la calle que cruza y los que doblan.

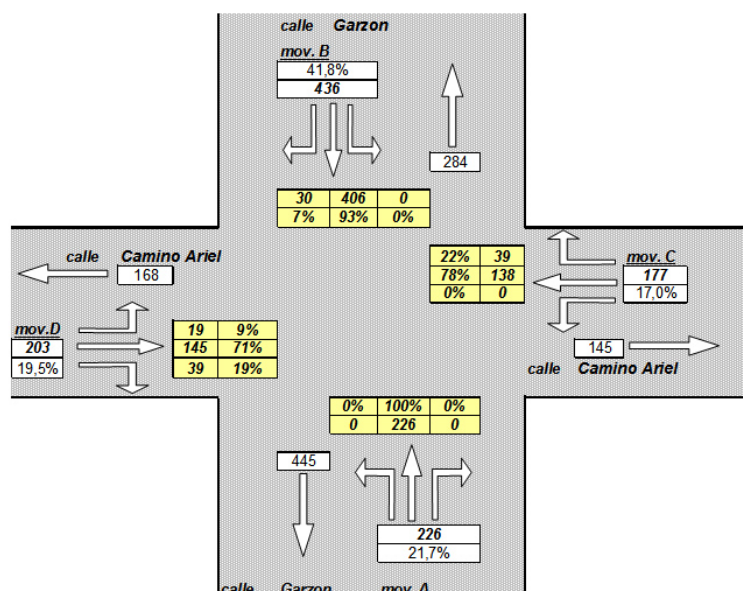


Figura 3.2: Ejemplo de planilla electrónica para el conteo manual en Camino Ariel

Además se realizaron recorridas de punta a punta del corredor a una velocidad constante en auto para verificar los tiempos obtenidos en la simulación.

Para obtener la configuración de los semáforos se realizó un recorrido en bicicleta por el corredor cronometrando la duración del tiempo en cada esquina de cada semáforo. Tanto de ida como de vuelta para corroborar que fueran correctos. Estos tiempos fueron verificados por los vídeos obtenidos.

Cuadro 3.1: Resumen del revelamiento del tráfico en la zona del corredor Garzón. Se muestra la cantidad de vehículos y la orientación hacia donde circulan.

Intersección	Sur-Garzon	Norte-Garzon	Oeste	Este
Camino Colman	410	390	0	230
Plaza Vidiella	400	444	292	0
Aparicio Saravia	390	450	40	90
Battle y Ordoñez	510	390	470	300
Camino Ariel	436	226	177	203

Modelo del tráfico

Una vez que tenemos el mapa compatible con el simulador y los datos relevados de la realidad necesitamos agregar el tráfico en la simulación.

Existen varios modelos disponibles para la representación de la circulación de vehículos, aquí exponemos algunos de los más populares.

- Aleatorios: Genera diferentes recorridos que seguirán los vehículos aleatoriamente
- Basado en áreas: Se especifican áreas como un conjunto de calles y se realizan recorridos entre ellas.
- Basado en Actividad: Se especifica la cantidad de casas, vehículos y población en un determinado lugar y el modelo genera la densidad de tráfico que se producirá basado en los tipos de actividades que se realizan comúnmente como ir al trabajo, hacer las compras, ir a la escuela, etc
- Definido por el usuario: que permiten determinar la ruta de los vehículos con mayor detalle.

Se utilizó el programa Traffic Modeler (Papaleontiou) que sirve para generar modelos de tráfico complejos de manera visual. Se optó por un modelo de movilidad entre áreas lo que permite una buena granularidad al especificar la densidad de tráfico.



Figura 3.3: Mapa del TrafficModeler con las áreas de tráfico. Círculos del mismo color indican tráfico especificado entre esas áreas

Se manejaron dos tipos de vehículos autos y ómnibus cada uno con características distintas tanto de tamaño, aceleración y velocidad máxima.

Se agregaron las frecuencias y los distintos recorridos de los ómnibus obtenidos de datos públicos de la Intendencia Municipal de Montevideo (IMM) Estos incluyen las líneas urbanas 'G', '409', '2', 'd5' y '148'.

Las líneas de ómnibus suburbanas realizan un mismo trayecto y las generalizamos con el nombre 'A' las cuales no van a ser tomadas en cuenta en la optimización del algoritmo pero si aparecerán en la simulación.

La ubicación y líneas de cada parada se obtuvo de (SIG IMM), existen 14 paradas para líneas urbanas por el corredor para el recorrido de ida y las mismas para la vuelta. Para los recorridos se hicieron variantes en los viajes dentro de una misma línea para simular el hecho de que no siempre se para en las mismas paradas.

También se tuvo en cuenta para la simulación el tiempo que demora un ómnibus al detenerse en una parada, y que hay algunas donde por la cantidad de gente se demora mas. Estos datos fueron obtenidos en el lugar considerando tiempos de entre 15 a 35 segundos.

Para establecer la velocidad media de los ómnibus se realizó un estudio basado en datos de GPS proporcionados por la IMM luego de reuniones con Ingeniería de Tránsito.

Estos datos cuentan con la posición GPS, la velocidad instantánea y la referencia al ómnibus para un conjunto de líneas seleccionadas tomadas en un periodo de una semana. Luego de procesar los datos se obtuvo una velocidad media de 14.5 km/h lo que permite ajustar mejor el modelo.

Para introducir estos datos en el simulador se utilizan tres archivos de configuración en formato xml que son tomados como entrada, estos son:

- Semáforos: Cuenta con la información detallada de la duración de los semáforos, el comienzo de su fase y la ubicación del mismo respecto al mapa base.
- Rutas de vehículos: Se indica la ruta de cada vehículo que recorrerá el mapa.
- Recorrido de ómnibus: Indica las rutas de los ómnibus, la frecuencia, la ubicación de las paradas, en cuales se detiene y cuanto demora.

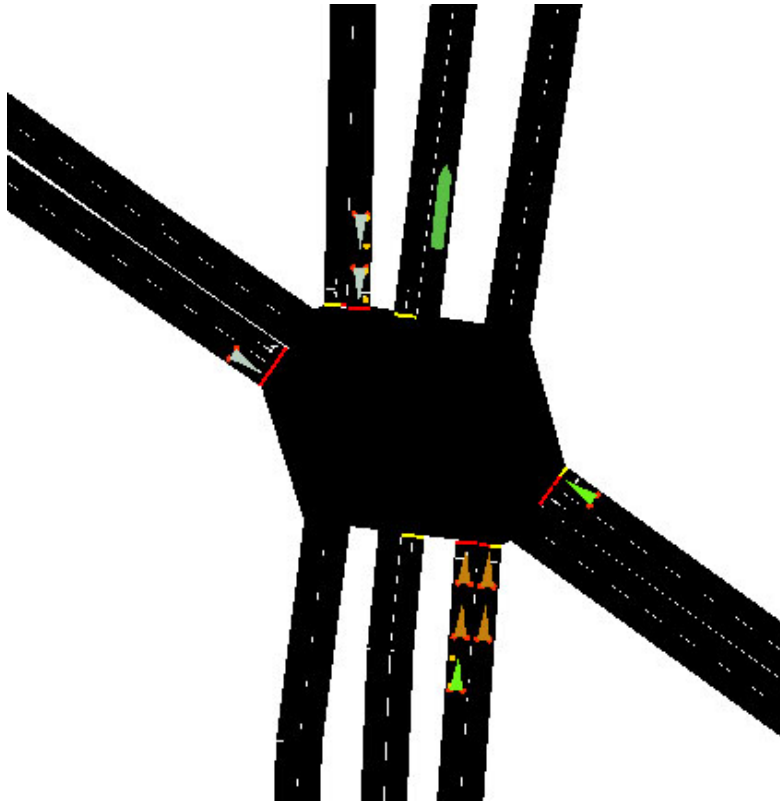


Figura 3.4: Simulación de tráfico en SUMO en el cruce entre Bulevar Battle y Ordoñez y el Corredor Garzon.

3.3. Arquitectura de la solución

En el siguiente diagrama se muestra la arquitectura propuesta para el problema.

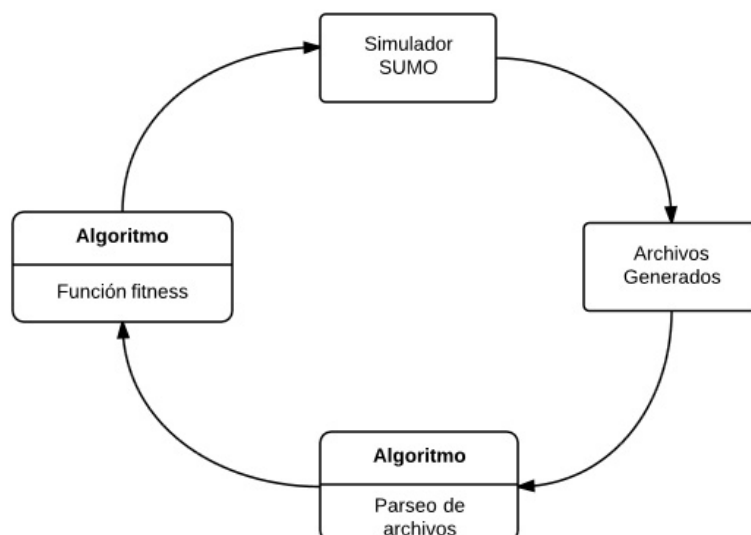


Figura 3.5: Arquitectura de la función de fitness

La biblioteca Malva se utiliza para la implementación del algoritmo, en cada evaluación de la función de fitness se realiza un llamado al simulador SUMO.

En el primer paso el algoritmo genera la población inicial, estos individuos representan una configuración de semáforos para el Corredor con el tiempo de duración de las luces y las fases. Esta población se genera realizando algunas variaciones de los valores reales obtenidos.

Luego por cada individuo se genera un archivo compatible con el simulador que representa la configuración de semáforos. Este archivo se utiliza como entrada para ejecutar el simulador para cada individuo. Al finalizar la ejecución se generan archivos de salida que son procesados por el algoritmo, estos tienen la información necesaria para calcular el fitness de cada individuo como la velocidad media de ómnibus y otros vehículos.

3.4. Biblioteca Malva

(Malva) surge como una variante del proyecto (Mallba). Propone su actualización, mejora y desarrollo como un proyecto de código abierto colaborativo. Su objetivo es proveer de varios esqueletos de heurísticas de optimización que puedan ser utilizados y extendidos de manera fácil y eficiente.

Los esqueletos se basan en separar dos conceptos: El problema concreto que se quiere resolver y por otro lado el método utilizado para resolverlo. Por tanto un esqueleto se puede ver como una plantilla genérica que se instancia para resolver un problema particular, manteniendo todas las funcionalidades genéricas.

Utiliza el lenguaje c++ dado su alto nivel, modularidad y flexibilidad. Los esqueletos se ofrecen como un conjunto de clases requeridas que son las que el usuario deberá modificar para adaptarlo a su problema y las provistas que incluyen todos los aspectos internos del esqueleto y son independientes del problema particular. Entre los algoritmos provistos se encuentra el de Algoritmos genéticos y (CHC).

3.5. Especificación del Algoritmo Genético utilizado

Se utiliza el algoritmo genético provisto por la biblioteca Malva llamado NewGA al cual se le realizan algunas modificaciones para lograr su ejecución en paralelo.

A continuación se realiza un resumen de sus características:

- Algoritmo paralelo: Utiliza el método maestro-esclavo para que en cada iteración el maestro genere un hilo para cada ejecución de la función fitness y luego espere a la terminación de todos los hilos para consolidar los datos.
- Función Multiobjetivo: Se intenta optimizar tanto la velocidad promedio de vehículos como de ómnibus teniendo cada uno un peso específico.
- Representación del cromosoma: Es un vector de números reales que representan los tiempos de los semáforos y sus fases.
- Cruzamiento y mutación: Se utiliza una variante del cruzamiento de un punto específico para el problema al igual que para la mutación.
- Selección y reemplazo: Reemplaza padres por hijos, la selección de los padres se realiza por el método de torneo de tres individuos y la selección de hijos por el método de ruleta.

El siguiente esquema muestra el algoritmo utilizado.

Algoritmo 3 Algoritmo Genético de Malva.

```

1:  $t = 0$ 
2: Inicializo(  $P(t)$ )
3: Evaluar estructuras en (  $P(t)$ )
4: while No termine do
5:    $t++$ 
6:   Seleccionar  $C(t)$  de  $P(t-1)$ 
7:   Recombinar estructuras en  $C(t)$  formando  $C'(t)$ 
8:   Mutar estructuras en  $C'(t)$  formando  $C''(t)$ 
9:   Evaluar estructuras en  $C''(t)$  generando un hilo de ejecución por cada una
10:  Consolidar valores de la evaluación
11:  Reemplazar  $P(t)$  de  $C''(t)$  y  $P(t-1)$ 
12: end while

```

3.5.1. Representación del cromosoma

Definimos los siguientes conceptos:

- Cruce: Es el lugar de intersección de dos o más vías de circulación.
- Fase: Es la configuración de las luces de los semáforos en un determinado cruce.

El cromosoma se va a agrupar lógicamente en cruces siendo el valor de cada gen el tiempo que demora una fase de un cruce, además se agrega para cada cruce un número que representa en que fase comenzará el semáforo. Por tanto el tamaño del cromosoma depende tanto de la cantidad de cruces como de la cantidad de fases de cada uno.

En la representación se omiten las luces amarillas ya que no modifican los tiempos reales del paso de vehículos.

Es importante que el algoritmo no genere soluciones inviables por lo que no debe modificar la combinación de luces de cada fase para evitar combinaciones de luces erróneas. Por tanto la modificación se realizará solo en el valor que indica el comienzo de fase y en los tiempos relacionados.

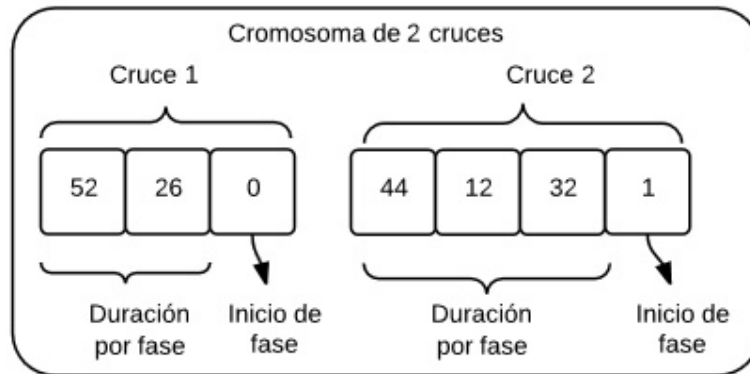


Figura 3.6: Cromosoma de 2 cruces

En la siguiente figura vemos el archivo con la configuración de semáforos que utiliza el simulador para el cromosoma anterior, donde se representan las fases. Por ejemplo el texto "GGGGrrGGGGrr" indica la secuencia de luces y su duración en este caso 52 segundos. "G" indica la luz verde, "r" la roja, "z." el Amarillo. El offset indica el inicio de la fase.

```
<add>
  <tlLogic id="cluster_-46_-48_-50" type="static" offset="0">
    <phase duration="52" state="GGGGrrGGGGrr" />
    <phase duration="3" state="yyyyrryyyyrr" />
    <phase duration="26" state="rrrrGGrrrrGG" />
    <phase duration="3" state="rrrryyrrrryy" />
  </tlLogic>
  <tlLogic id="cluster_1858033111_917381626_917457905" type="static" offset="1">
    <phase duration="44" state="rrrrrrrrGGGGrrrr" />
    <phase duration="3" state="rrrrrrrryyyyrrrr" />
    <phase duration="12" state="rrrrrGrrrrrrrGrr" />
    <phase duration="3" state="rrrryyrrrrrryyrr" />
    <phase duration="32" state="GGGGrrrrrrrrrrrr" />
    <phase duration="3" state="yyyyrrrrrrrrrrrr" />
  </tlLogic>
</add>
```

Figura 3.7: Representación de Sumo

3.5.2. Población inicial

Para la inicialización de la población se toma como referencia la configuración obtenida con los datos in-situ, luego para cada cruce se hacen variar las duraciones de las

fases de manera aleatoria entre un rango de valores configurable. Se elige la fase inicial aleatoriamente entre la cantidad de fases del cruce.

3.5.3. Función fitness

La evaluación de un individuo se realiza generando un archivo con la configuración de los semáforos en base a su cromosoma y ejecutando el simulador SUMO utilizando esta configuración para luego obtener los tiempos necesarios para calcular el fitness.

Se empleara una función multiobjetivo usando combinación lineal de la velocidad de los ómnibus y del resto de los vehículos, ya que es un método sencillo y adecuado cuando son menos de tres objetivos. El fitness se calcula como una suma ponderada, con los pesos fijados a priori.

$$F(x) = \sum_{i=1}^n w_i f_i(x) \quad (3.1)$$

En nuestro caso tenemos como objetivo la velocidad promedio de los ómnibus (vpb) y la velocidad promedio del resto de los vehículos (vpv).

Esta es la formula de fitness donde x e y indica el peso que le vamos a especificar a la función.

$$f = x.vpb + y.vpv \quad (3.2)$$

En una primera instancia se establece $x = y = 1$ mas adelante se experimentara con otros pesos.

3.5.4. Operadores

Operador de Cruzamiento

Se utilizará cruzamiento de un punto, seleccionando del cromosoma una posición aleatoria entre dos cruces como punto de corte, por tanto si un tramo del corredor tiene un buen comportamiento se intentara mantener esta propiedad.

En la siguiente figura los padres cuentan con tres cruces, se elige un punto de corte aleatoriamente entre dos cruces y se generan los hijos que son una combinación de los padres.

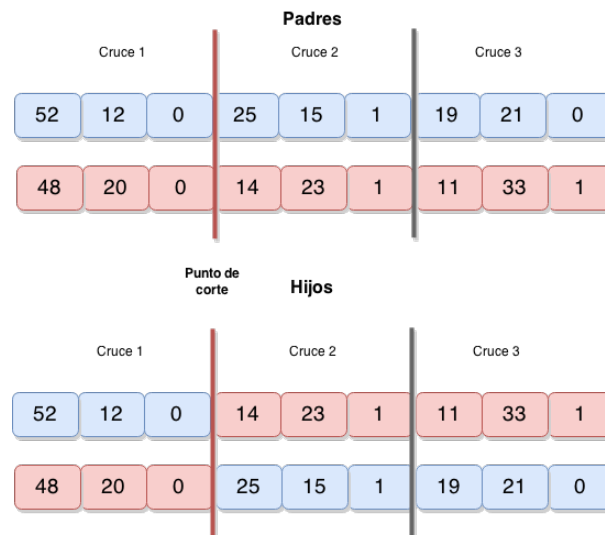


Figura 3.8: Visualización del cruzamiento entre individuos.

Operador de Mutación

Se implementaron dos tipos de mutación:

- Mutación de duración de fase: para cada fase de cada cruce se hace variar su duración sumando o restando una cantidad dada de segundos entre un rango determinado con una probabilidad dada.
- Mutación de inicio de fase: se elige aleatoriamente una fase con la cual va a comenzar inicialmente el cruce con una probabilidad dada.

Operador de selección y reemplazo

Se utilizan los provistos por el algoritmo newGA de Malva. La selección de padres se realiza por el método de torneo de tres individuos y la selección de hijos por el método de ruleta. La política de reemplazo indica que tanto padres como hijos pueden ser parte de la siguiente generación, por lo que existe reemplazo de padres por hijos.

Parámetros del algoritmo

Los parámetros específicos del algoritmo se establecen en la siguiente sección, donde se realiza un análisis experimental para encontrar los mas adecuados. Estos parámetros son: numero de generaciones, tamaño de población, probabilidad de cruzamiento y de mutación.

Capítulo 4

Análisis Experimental

En esta sección se presenta la descripción de los escenarios, la plataforma de ejecución y el análisis experimental.

Este se divide en dos etapas, primero se realiza la configuración paramétrica para encontrar el mejor rendimiento del algoritmo y luego se realizan las pruebas donde se muestran los resultados y comparaciones.

4.1. Desarrollo y Plataforma de ejecución

Los algoritmos fueron desarrollados usando la biblioteca Malva que fue extendida en el código base para soportar la creación de nuevos hilos de ejecución para lograr el funcionamiento en paralelo.

Los escenarios fueron ejecutados en el cluster Fing.

Cluster: Es un conjunto de computadoras independientes conectadas para que trabajen integradas como un solo sistema. De esta forma se consigue un alto rendimiento en la ejecución de tareas.

Cluster Fing: Es una infraestructura de alto desempeño, que brinda soporte en la resolución de problemas complejos que demandan un gran poder de computo.

Descripcion del hardware:

- 9 servidores de cómputo

- Quad core Xeon E5430, 2x6 MB caché, 2.66GHz, 1.333 MHz FSB.

- 8 GB de memoria por nodo.

- Adaptador de red dual (2 puertos Gigabit Ethernet).

- Arquitectura de 64 bits.

- Servidor de archivos: 2 discos de 1 TB, capacidad ampliable a 10 TB.

- Nodos de cómputo: discos de 80 GB.

- Switch de comunicaciones

- Dell Power Connect, 24 puertos Gigabit Ethernet.

- Switch KVM (16 puertos) y consola.

- UPS APC Smart RT 8000VA.

4.2. Ajuste paramétrico

Se busca la mejor configuración inicial de los parámetros realizando pruebas experimentales con diferentes combinaciones. Los elementos que se ajustaran serán los siguientes:

- Tiempo de simulación
- Criterio de parada
- Tamaño de la población
- Probabilidad de mutación
- Probabilidad de cruzamiento

Para la realización de las pruebas se generan tres escenarios de tráfico diferentes para realizar las pruebas de configuración. En este caso no se utilizan datos recabados de la realidad como el tráfico específico de las calles y el recorrido de las líneas de ómnibus. Esto para no sesgar el algoritmo a un caso en particular. Los tres escenarios de tráfico son los siguientes:

- Tráfico Bajo: 30 ómnibus y 500 vehículos
- Tráfico Medio: 60 ómnibus y 1000 vehículos
- Tráfico Alto: 120 ómnibus y 2000 vehículos

En el ajuste paramétrico primero se define el tiempo de simulación y el criterio de parada, luego de establecidos se realizan las pruebas para todas las combinaciones de tasa de cruzamiento y mutación buscando los mejores valores para optimizar el algoritmo.

Al estar utilizando un cluster tenemos disponible tanto la métrica del tiempo real que llevo la ejecución, así como también el tiempo secuencial, es decir la suma del tiempo de procesamiento de todos los procesadores involucrados en la evaluación del algoritmo. Cuando se realicen comparaciones en los tiempos de ejecución se utilizara el tiempo secuencial que es independiente de la cantidad de procesadores utilizados en las pruebas. Las diferentes ejecuciones fueron realizadas sobre el Cluster utilizando entre 4 y 32 procesadores ya que por la naturaleza de recursos compartidos no siempre se tiene una cantidad igual de procesadores libres y para estas pruebas el numero de procesadores no es relevante. Si lo sera cuando realicemos el análisis de eficiencia computacional que sera estudiando mas adelante.

4.2.1. Pesos de la función fitness

Para las siguientes pruebas la función fitness del algoritmo (3.2) tendrá los pesos $x = y = 1$. Esto da pesos equitativos tanto a ómnibus como a otros vehículos por lo que no existe prioridad para uno u otro. Mas adelante se realizaran experimentos con otras variantes.

4.2.2. Tiempo de simulación

Esto se refiere al tiempo que representa una simulación usando SUMO, es un parámetro que se puede configurar y permite tener un mejor control sobre los tiempos totales de ejecución del algoritmo.

Se establece un tiempo de simulación de 4000 *steps* (medida interna de tiempo del simulador SUMO). Este numero representa 66 minutos en la simulación, mientras el tiempo real de ejecución depende de la plataforma y de la cantidad de vehículos pero se encuentra entre los 5 a 20 segundos. Se tuvo en cuenta y validó que en cada escenario mas del 80 % de los vehículos hayan completado la simulación, es decir llegado a sus destinos. Se realizaron 10 ejecuciones del algoritmo para cada uno de los tres tipos de tráfico comprobando que se cumplía el criterio.

4.2.3. Criterio de Parada

Se elige como criterio de parada el número de generación, esto permite estandarizar las pruebas para una mejor comparación.

Para determinarlo se busca un compromiso entre un buen resultado y un tiempo de ejecución apropiado que no sea excesivo.

Para esto se decide que por un lado la ejecución del algoritmo deberá estar comprendida entre 1 y 24 horas, y además comprobar experimentalmente que el valor de fitness no tiene una gran variación en las ultimas 100 generaciones.

Se ejecutaron 10 ejecuciones del algoritmo por cada tipo de tráfico.

Luego de la realización de las pruebas se elige el número de 500 generaciones como criterio de parada. En la siguiente gráfica se aprecia como el valor de fitness no presenta grandes variaciones luego de la generación 400 ademas el tiempo de ejecución real esta dentro del margen pautado. La gráfica no muestra todos los valores, solo algunos representativos para una mejor visualización.

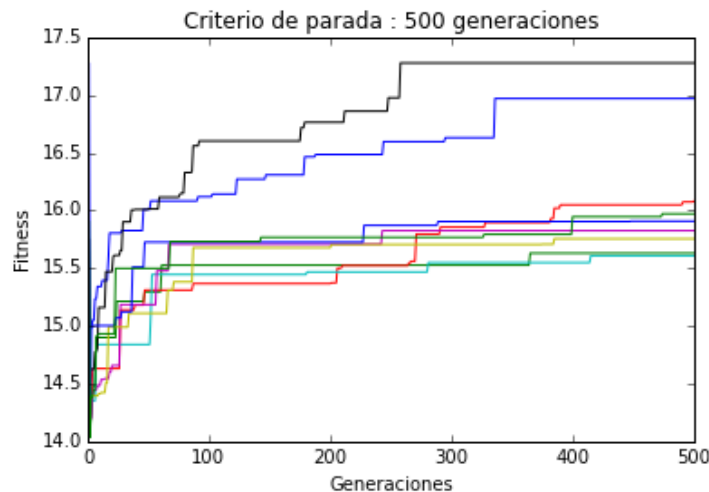


Figura 4.1: Resumen representativo de ejecuciones del algoritmo para establecer el criterio de parada.

4.2.4. Tamaño de la población

Para la elección de la población se tendrán en cuenta 3 elementos. El valor de fitness encontrado, el tiempo de ejecución total y la plataforma de ejecución.

Dado que estamos ejecutando en el cluster y la máxima cantidad de procesadores que se pueden utilizar son 64 en un mismo nodo y teniendo en cuenta que la mejor distribución del trabajo es un elemento de población por procesador, se tiene que la máxima cantidad de población que estudiaremos sera 64.

Luego se eligen los valores 32 y 48 para completar el análisis teniendo en cuenta que no son lo suficientemente bajos y son valores con los que se obtiene una distribución más adecuada.

Se realizan 10 ejecuciones del algoritmo por cada tipo de tráfico y luego se obtiene el promedio de esos valores.

La siguiente tabla muestra los resultados obtenidos, como se aprecia no existen grandes diferencias en la elección de un número poblacional sobre otro. Por tanto se elige como número de población 32 teniendo en cuenta el tiempo de ejecución secuencial del algoritmo que como se aprecia es el menor. Se elige esta métrica por que aunque al ejecutar en paralelo el tiempo real que se puede obtener utilizando la máxima cantidad de procesadores para cada población son similares también se tiene en cuenta la disponibilidad y utilización de recursos que insume en el cluster fing. Por ejemplo obtener 64 procesadores para utilizar por un proceso en el cluster es algo que puede demorar varios días por la cantidad de otros procesos que también están funcionando en la plataforma y como se ve la ganancia que tenemos es mínima.

Cuadro 4.1: Comparación de fitness para distintas poblaciones

Población	Fitness		Tiempo ejecución serial (m)
	mejor	promedio	
32	17.28	16.37±0.5	10184±526
48	16.19	15.84±0.3	6772±256
64	17.27	16.46±0.6	4853±155

4.2.5. Probabilidad de mutación y cruzamiento

Los valores elegidos fueron:

- Probabilidad de cruzamiento (pc): 0.5, 0.8, 1
- Probabilidad de mutación (pm): 0.01, 0.05, 0.1

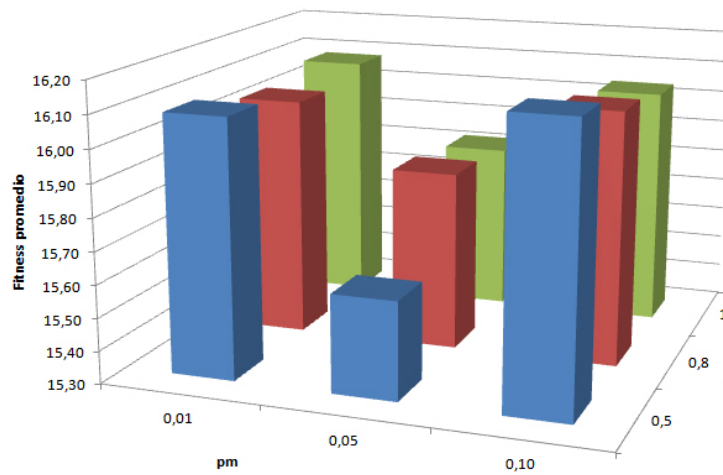
Se prueban las nueve combinaciones posibles de cruzamiento y mutación, para cada una se realizan 3 ejecuciones para el escenario con tráfico bajo, 3 con tráfico medio y 3 con tráfico alto.

Analizando la tabla y la gráfica se puede apreciar claramente que para una probabilidad de mutación de 0.05 se obtienen los peores resultados. Otro dato interesante es que no existe gran diferencia en el resto de las combinaciones.

Se comprueba que todas las muestras siguen la distribución normal para poder aplicar el test de Student.

Cuadro 4.2: Combinaciones de probabilidad de cruzamiento(p_C) y de mutación (p_M)

p_C	p_M	Fitness promedio \pm desviación estándar
0.5	0.01	16.09 \pm 0.30
0.5	0.05	15.60 \pm 0.17
0.5	0.1	16.16 \pm 0.42
0.8	0.01	16.04 \pm 0.55
0.8	0.05	15.85 \pm 0.32
0.8	0.1	16.08 \pm 0.34
1	0.01	16.08 \pm 0.45
1	0.05	15.82 \pm 0.34
1	0.1	16.04 \pm 0.25

Figura 4.2: Gráfica con combinaciones de probabilidad de cruzamiento(p_C) y de mutación (p_M)

Siendo las hipótesis de la prueba (μ_1 el promedio del grupo 1 y μ_2 el del grupo 2)
 $H_0: \mu_1 = \mu_2$
 $H_1: \mu_1 \neq \mu_2$

Si hacemos una comparación entre la combinación del mejor promedio (0.5-0.1) y del peor (0.5-0.05) con el test de Student obtenemos $t(x) = 0.07$ que nos indica que para un nivel de significancia de 0,1 la hipótesis nula es rechazada por lo tanto existe evidencia estadística para elegir la combinación con el mejor promedio (0.5-0.1) sobre la combinación con el peor promedio (0.5-0.05) para la ejecución del algoritmo.

Para comprobar si es la mejor opción se toman las dos combinaciones con el mejor promedio (0.5-0.1) y (0.5-0.01) obteniendo en el test Student $t(x) = 0.71$ que nos indica que no existe una diferencia significativa entre ambas muestras por lo que elegir una sobre otra no implicaría grandes beneficios.

En tal sentido podríamos elegir cualquiera de las dos, en este caso se elige la combinación (0.5-0.01) por su buen promedio y baja desviación estándar.

4.3. Descripción de escenarios

En esta sección se presentan los escenarios que serán evaluados, el primero es el escenario base que representa la realidad actual y el segundo un escenario alternativo que contiene modificaciones con el objetivo de mejorar la realidad.

4.3.1. Caso base o Realidad actual del corredor

Esto representa la situación actual en términos de tráfico, red vial y sincronización de semáforos del corredor Garzón. El objetivo es obtener datos precisos para realizar una simulación de la realidad y poder obtener datos para comparar con los resultados del algoritmo.

Se valida su correctitud comparando los tiempos obtenidos en la simulación con tiempos obtenidos in-situ de los recorridos de ida y vuelta para los vehículos. Para el caso de los ómnibus se utilizan las frecuencias, cantidad y recorridos que son de acceso público.

Se realizó un estudio sobre datos proporcionados por la IMM que contenían el posicionamiento de los ómnibus, velocidad instantánea y datos de la línea durante todo el día para una semana en particular. De esta forma se constató que para de las líneas de ómnibus que pasan por Garzón la velocidad promedio de los ómnibus es de 14.5km/h.

Esto permitió calibrar el escenario modificando aspectos de la simulación relacionados con los ómnibus para mayor precisión.

Sobre este escenario geográfico se realizaron tres escenarios de tráfico : bajo, medio y alto. El caso medio representa los datos obtenidos, el bajo es disminuyendo el 50 % de vehículos y el tiempo de espera en las paradas de ómnibus teniendo en cuenta que en este caso existirá menos personas utilizando el transporte público. Las frecuencias de ómnibus se mantienen iguales ya que no son alteradas en la realidad. El caso de tráfico alto se aumenta 50 % los vehículos y el tiempo de espera en la parada de los ómnibus.

El aumento y disminución del 50 % se obtuvo al analizar datos proporcionados por la IMM de la zona de Garzón de años anteriores.

En resumen tenemos los siguientes escenarios de tráfico:

- Tráfico Alto: 2800 vehículos en la simulación y 70 ómnibus.
- Tráfico Medio: 2000 vehículos y 70 ómnibus.
- Tráfico Bajo: 1000 vehículos y 70 ómnibus.

4.3.2. Escenario Alternativo

Para mostrar la utilidad que tienen las simulaciones sobre un escenario real, se realiza solo a modo de ejemplo un escenario alternativo. Una de las ventajas principales es que no requiere gran inversión monetaria, de tiempo y que no afecta la situación actual de la realidad, por lo que se pueden generar distintas pruebas para encontrar aquellas que logren un beneficio.

Analizando aquellos puntos que se entienden podrían atentar contra el buen funcionamiento del Corredor, se agregan algunas modificaciones al escenario base para intentar mejorarlo.

El objetivo no es demostrar que esta sera la mejor alternativa sino dar una de las muchas alternativas que se pueden generar y probar con la simulación si se logran mejoras. Ya que pueden existir limitaciones o reglas que no estamos tomando en cuenta y que deben cumplirse en la realidad.

Entre los cambios estudiados se encuentran: eliminación de paradas y pasajes peatonales, alternar paradas y modificación de reglas de semáforos.

4.4. Resultados

Se muestran los resultados obtenidos tanto de la simulación de la realidad, como de la evolución del algoritmo sobre el mismo. Se presenta la simulación del escenario alternativo y la posterior evaluación. Además se realizan estudios sobre cambios en la función de fitness del algoritmo y un breve análisis de la eficiencia computacional.

4.4.1. Valores numéricos del caso base

En la tabla se pueden ver las métricas obtenidas para los diferentes instancias de tráfico simulado.

Cuadro 4.3: Resultados del caso base mostrando la velocidad promedio ómnibus (vpb) y velocidad promedio vehículos(vpv) para los distintos tipos de tráfico

	$vbp(km/h)$	$vpv(km/h)$	Fitness
Tráfico Bajo	15.89	32.45	13.42
Tráfico Medio	14.59	28.81	12.05
Tráfico Alto	14.31	26.36	11.30

4.4.2. Resultados numéricos de la evaluación

Cuadro 4.4: Resultados luego de ejecutado el algoritmo mostrando velocidad promedio ómnibus (vpb) y de otros vehículos(vpv) para los distintos tipos de tráfico

Tráfico	$vbp(km/h)$	$vpv(km/h)$	Fitness		Mejora fitness (%)	
			Promedio	Mejor	Promedio	Mejor
Bajo	17.92±0.18	34.30±0.40	14.50±0.14	14.88	8.04	10.8
Medio	16.95±0.32	33.29±0.29	13.95±0.15	14.19	15.70	17.7
Alto	16.51±0.61	32.90±0.25	13.72±0.17	14.04	21.40	24.2

Se realizaron 20 ejecuciones independientes para cada tipo de tráfico comprobando que siguieran una distribución normal.

Por tanto se puede aplicar el criterio de significancia estadística para validar los resultados, que dice que el algoritmo A es mejor que el algoritmo B si los resultados de A y B cumplen:

$$|f_{avg}(A) - f_{avg}(B)| > \max(std(f_A), std(f_B)) \quad (4.1)$$

En este caso A representa el algoritmo y B el caso base. Esto indica que la diferencia del resultado promedio del caso base restado al resultado del caso base debe ser mayor a la máxima desviación. Esto se cumple para todos los casos, por lo que podemos afirmar que existe evidencia estadística para decir que los resultados del algoritmo son mejores al del caso base.

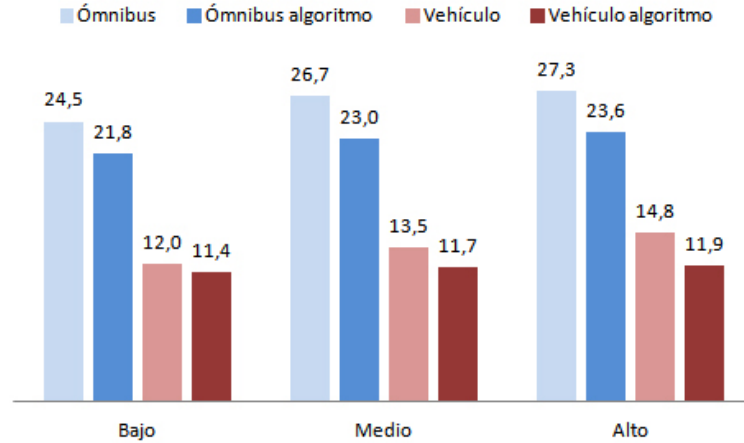


Figura 4.3: Comparación de la duración en minutos de los viajes para ómnibus y otros vehículos en el recorrido completo del corredor Garzón para los diferentes tipos de tráfico.

Como se aprecia en el gráfico el algoritmo mejora la duración de los viajes tanto de ómnibus como de otros vehículos en los tres tipos de tráfico estudiados. Como cabía esperar aumenta la duración del viaje al aumentar el tráfico, aunque un resultado interesante es que la duración original de los viajes cuando el tráfico es bajo es casi igual a la duración de los viajes con tráfico alto luego de ejecutar el algoritmo. Para ómnibus tenemos 24.5m y 23.6m y para otros vehículos 12.0m y 11.9m.

4.4.3. Detalles del escenario alternativo

Los cambios propuestos incluyen eliminación de paradas, semáforos, pasajes peatonales y alternar paradas. Se estudiaron otras propuestas pero fueron descartadas por la poca viabilidad real de las mismas. Como por ejemplo construir calles paralelas a Garzón o nuevas reglas en los cruces como existen en otros países.

Eliminación de paradas

Se consideraron dos paradas a eliminar que cumplieran con algunas características, no fueran cercana a una calle principal y que existiera otra parada cercana. Por lo que la eliminación de la parada no afecte en demasía a la gente en un traslado mayor. En este caso se selecciono la parada en la calle Ariel y Casavalle.

Eliminación de pasajes peatonales

Hay tres pasajes peatonales en el corredor con semáforos que detienen el tráfico, dos de ellos solo manejan una esquina (sin pulsador en funcionamiento) donde en el escenario alternativo se implemento solamente mediante un "Pare.^{en} la calle transversal al corredor y el otro es netamente peatonal frente a la Facultad de Agronomía que fue totalmente eliminado. Una opción que mantiene los pasajes peatonales así como también los resultados obtenidos en el escenario alternativo sería implementar el pasaje peatonal por encima del corredor. Al eliminar los pasajes peatonales se aumenta la velocidad media de todo el transporte.

Alternar paradas

Uno de los problemas del ómnibus es su baja aceleración por lo que cada vez que este frena en un semáforo o en una parada demora en retomar una velocidad aceptable. Por tanto al reducir la cantidad de paradas que un ómnibus tiene que hacer se mejora la velocidad promedio. La línea G es la que recorre a Garzón de punta a punta, esta línea es cubierta por las empresas Coectc y Cutcsa. Una posibilidad de alternancia de paradas consiste en dividir las paradas por empresa y compartir las ganancias del corredor u otro método para equiparar el pasaje transportado.

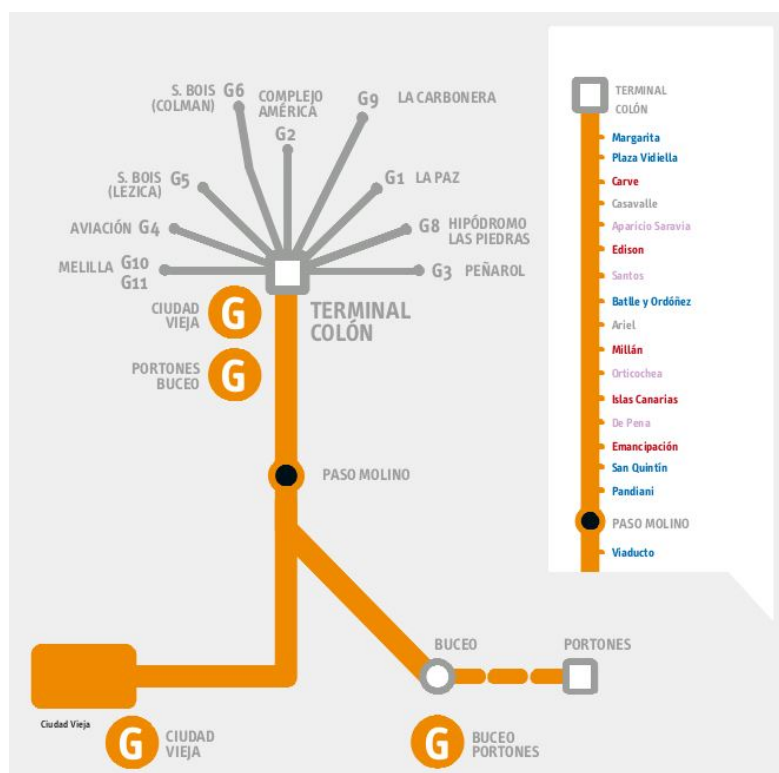


Figura 4.4: Gráfico de paradas alternativas. Gris: Parada Eliminada. Azul: línea G de Coectc y de Cutcsa. Rojo: línea G de Coectc. Violeta: G de Cutcsa. - Imagen original extraída de montevideo.gub.uy

Una empresa se detendrá en las paradas pares y la otra en las impares y algunas de mayor aglomeración de pasaje/tiempo serán realizadas por las dos. Cada empresa

viajará por el corredor a 4 minutos (como en la actualidad) Si se reduce el numero de paradas que hace un ómnibus, aumentará su velocidad promedio y al cubrir las paradas entre las dos empresas no se deberá resentir en demasía el servicio ya que la disminución de la frecuencia en una parada se contrarresta con el aumento promedio de velocidad.

Este cambio es el que mas aumenta la velocidad media y tal vez es uno de los mas sencillos de implementar en la realidad.

Cambio básico de semáforos

Al hacer el revelamiento de los datos se encontró que en todas las intersecciones en donde una línea de ómnibus que circula por el corredor tiene un viraje a la izquierda, se hace detener el tránsito de la derecha de la misma, cada vez que el corredor central tiene la luz verde. Esto no parece tener mucho sentido ya que podrían seguir circulando por el corredor sin ningún tipo de problema, el carril que hay que detener es el de la izquierda del ómnibus cuando una línea dobla a la izquierda pero no los dos carriles al mismo tiempo.

No se tiene conocimiento si esto corresponde a un error en la configuración, un tema de costos o facilidad para manejar los dos semáforos de los carriles paralelos juntos. Al suceder en varias intersecciones y para los dos lados se mejora la velocidad promedio de los autos que circulan por los dos carriles.

Este cambio se aplico en las siguientes intersecciones:

- Islas Canarias: dobla línea 409 hacia la izquierda, orientado a Colon.
- Camino Ariel: doblan líneas como la 2 y la 148 hacia la izquierda, orientado a Paso Molino
- Camino Casavalle: dobla línea 174 hacia la izquierda, orientado a Paso Molino

Resumen

Lo interesante del resultado obtenido de esta modificación es que aumenta la velocidad promedio de los vehículos y disminuye la de los ómnibus. Aunque igualmente aumenta el fitness.

En la siguiente tabla se detallan los cambios realizado con la mejora asociada, comparando los valores para el tráfico medio. Estos cambios son acumulativos, por lo que se hacen uno después del otro . Luego se detallan los resultados obtenidos para el tráfico bajo y alto.

Cuadro 4.5: Valores del escenario alternativo con su velocidad promedio ómnibus (vpb) y velocidad promedio vehículos(vpv) comparando el fitness para el tráfico medio

	$vbp(km/h)$	$vpv(km/h)$	Fitness	Mejora(%)
Base	14.59	28.81	12.05	-
Eliminar Paradas	15.44	29.03	12.35	2.4
Eliminar Peatonales	16.02	29.32	12.59	4.4
Paradas alternadas	19.17	28.88	13.34	10.7
Cambio reglas	18.50	29.70	13.39	11.1

Cuadro 4.6: Mejoras obtenidas para las velocidades promedio de los omnibus(vpb) y de otros vehiculos (vpv) en el escenario alternativo para distintos tipos de tráficos

	$vbp(km/h)$	$vpv(km/h)$	Fitness	Mejora fitness(%)
Tráfico Bajo	20.72	33.18	14.97	11.5
Tráfico Medio	18.50	29.70	13.39	11.1
Tráfico Alto	18.60	27.17	12.7	12.6

4.4.4. Resultados numéricos de la evaluación sobre el escenario alternativo

Al comparar los resultados obtenidos se aprecia claramente que cuanto mas densidad de tráfico mayor es el porcentaje de mejora. Ademas un resultado interesante es que las diferencias entre los valores de los distintos tipos de tráfico se redujo.

Esto significa que el recorrido de un ómnibus por Garzón pasa de demorar 26,7 minutos a 17,8 minutos y un auto de demorar 14,5 minutos a 11.5 minutos con una densidad de tráfico media.

Cuadro 4.7: Mejoras obtenidas al aplicar el algoritmo sobre el escenario alternativo. Comparando las velocidades de ómnibus(vpb), otros vehículos(vpv) y el fitness con cada tipo de tráfico contra el caso base o realidad actual.

Tráfico	$vbp(km/h)$	$vpv(km/h)$	Fitness		Mejora fitness (%)	
			Promedio	Mejor	Promedio	Mejor
Bajo	23.15±0.36	34.43±0.33	15.99±0.08	16.10	19.1	19.90
Medio	21.83±0.50	33.89±0.22	15.47±0.09	15.65	28.3	29.87
Alto	21.46±0.54	33.41±0.38	15.24±0.19	15.50	34.8	37.10

4.4.5. Variación de la función de fitness

La función fitness (3.2) utilizaba los pesos $x = y = 1$ lo que representa un balance equitativo para ómnibus y vehículos.

Estos pesos pueden ser variados en función de lo que se necesite, por lo que se realizaran pruebas con dos tipos de pesos para comparar como varían las velocidades cuando se da mas peso a un tipo de vehículo sobre el otro.

Prioridad ómnibus

En este caso se le dará mas prioridad a los ómnibus, esto se sostiene en el hecho que uno de los objetivos buscados por la IMM es que se utilice mas el transporte colectivo como parte de su plan de movilidad urbana (IMM, 2010). Con la premisa que al mejorar la duración del viaje en ómnibus relativo al del auto por el corredor las personas que utilizan auto para sus viajes opten por el transporte colectivo.

Por tanto vamos a experimentar cambiando los pesos de la función fitness con un peso de 70 % para los ómnibus y 30 % al resto de los vehículos.

Prioridad a otros vehículos

En este caso, le damos 70 % del peso a los vehículos y 30 % a los ómnibus. Es el caso opuesto al anterior y servirá para poder comparar como varían los valores de las velocidades.

Resultados

La siguiente tabla compara las velocidades promedio de ómnibus y vehículos para los tres tipos de pesos que se probaron y por cada tipo de tráfico. El caso 50-50 es el caso base donde los pesos son iguales, 70-30 es el caso con mas prioridad para los ómnibus y el 30-70 mas prioridad a los otros vehículos. Se analiza cuanto varían las velocidades de omnibues (var. vpb) y otros vehículos(var. vpv) comparando contra el caso 50-50 de cada tipo de tráfico.

Cuadro 4.8: Modificación de los pesos para ómnibus (pb) y para otros vehículos (pv) en la función fitness. Analizando las variaciones en la velocidad promedio de ómnibus (vpb), otros vehículos (vpv) y fitness.

Trafico	pb(%) pv (%)	vpb	vpv	fitness	var. vpb(%)	var. vpv(%)	var. fitnesss(%)
Bajo	50-50	17.92±0.18	34.30±0.40	14.50±0.14	-	-	-
	70-30	17.93±0.23	34.06±0.17	12.65±0.11	+0.07	-0.70	-12.79
	30-70	17.55±0.23	34.71±0.21	16.42±0.10	-2.06	+1.18	+13.21
Medio	50-50	16.95±0.32	33.29±0.29	13.95±0.15	-	-	-
	70-30	17.29±0.27	33.08±0.14	12.24±0.12	+2.0	-0.62	-12.30
	30-70	16.71±0.42	33.79±0.31	15.92±0.11	-1.41	+1.49	+14.11
Alto	50-50	16.51±0.60	32.90±0.25	13.72±0.17	-	-	-
	70-30	16.72±0.14	32.79±0.26	13.75±0.07	+1.24	-0.33	+0.19
	30-70	15.48±0.42	33.20±0.25	15.49±0.16	-6.23	+0.92	+12.87

Los resultados indican que al variar los pesos de la función fitness las velocidades promedio de los vehículos se ve afectada. En el caso de dar mas prioridad a los ómnibus se produce como cabía esperar un aumento en su velocidad promedio y una leve baja en la velocidad promedio del resto de los vehículos. Cuando el tráfico es bajo este cambio casi no aumenta la velocidad de los ómnibus, una explicación posible de este comportamiento es que ya se llevo a un limite máximo y no se puede mejorar mas.

Al dar mas prioridad a los otros vehículos se produce un aumento en su velocidad y una disminución en la velocidad de los ómnibus la cual es muy evidente en el caso de tráfico alto. Este resultado permite apreciar como estos valores son fuertemente afectados por la densidad de tráfico que se estudie.

En general las variaciones en las velocidades no son grandes pero suficientemente apreciable para tener cierta libertad al plantear distintos objetivos que tiendan a favorecer un tipo u otro de vehículos.

4.4.6. Eficiencia computacional

Se realiza un estudio de la eficiencia computacional del algoritmo para analizar los tiempos de ejecución cuando se usan varios procesadores y como se comporta su capacidad de paralelismo.

Se evalúan nueve ejecuciones del algoritmo; tres con cada tipo de tráfico: alto, medio y bajo. Para estudiarlo en diferentes contextos. El algoritmo utiliza 32 hilos de ejecución por lo que utilizamos esa cantidad de procesadores.

Las pruebas fueron realizadas sobre el node40 del Cluster Fing que tiene las siguientes características:

- Modelo: HP Proliant DL585
- Procesador: AMD Opteron 6272 2.09GHz
- Memoria: 48GB
- Cores utilizados: 32

El speedup(S) mide la mejora de rendimiento de una aplicación al aumentar la cantidad de procesadores comparando con el rendimiento al usar un solo procesador.

$$S = \frac{T_1}{T_N} \quad (4.2)$$

Donde T_1 es el tiempo de ejecución del algoritmo serial o secuencial, y T_N el tiempo del algoritmo ejecutado sobre N procesadores.

La eficiencia computacional (E) corresponde al valor normalizado del speedup (entre 0 y 1) respecto a la cantidad de procesadores. Los valores cercanos a uno indican una alta eficiencia computacional.

$$E = \frac{T_1}{N * T_N} = \frac{S}{N} \quad (4.3)$$

Cuadro 4.9: Análisis de eficiencia computacional comparando los tiempos de ejecución en serial y paralelo en minutos.

Instancia	Serial(m)	Paralelo(m)	Speedup	Eficiencia
bajo1	1572	59	26.64	0.83
bajo2	1571	59	26.62	0.83
bajo3	1183	44	26.88	0.84
medio1	3002	119	25.22	0.78
medio2	2195	82	26.76	0.83
medio3	3007	120	25.05	0.78
alto1	2920	110	26.5	0.82
alto2	4365	183	23.85	0.74
alto3	4276	177	24.15	0.75
		Promedio	25.7±1.1	0.80±0.03

El algoritmo paralelo logra una mejora sustancial en los tiempos de ejecución con un valor promedio del speedup de 25.7 y eficiencia promedio de 0.8, lo que podemos considerar como buenas métricas.

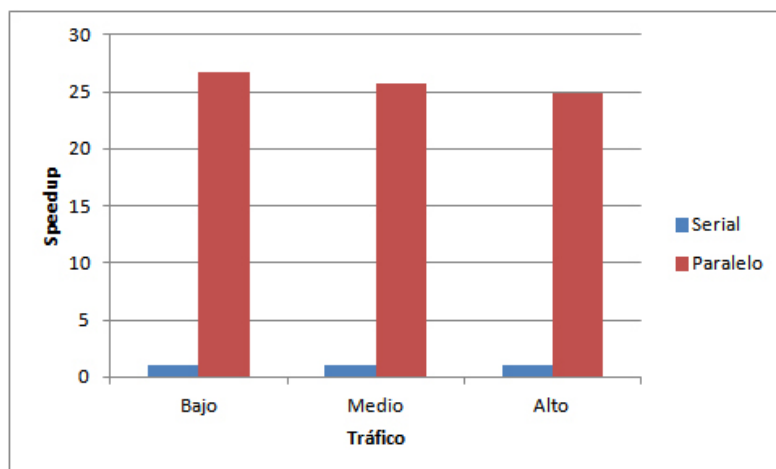


Figura 4.5: Comparación de los speedup promedios para cada tipo de tráfico.

En la gráfica se observa como a medida que aumenta el tráfico disminuye el speedup esto sucede por que esta influenciado por los accesos al disco duro, al tener mas autos circulando en la simulación se tiene que leer y escribir mas información en los archivos lo que aumenta el tiempo de ejecución del algoritmo, aunque como se ve esto no tiene un gran impacto.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

Todos los objetivos planteados al principio del trabajo fueron cumplidos; se encontró información útil al realizar la investigación sobre los trabajos relacionados que ayudo a mejorar la solución presentada. Al estudiar el problema del tráfico se constato que realmente afecta la vida de las personas y la búsqueda de nuevas soluciones pueden mejorar su calidad de vida. Uno de los trabajos mas tediosos es la confección del mapa y los datos relacionados con la simulación ya que se intento en todo momento acercarse a la realidad y para ellos se puso énfasis en la recolección de datos en el lugar, la validación de los resultados, preguntar a expertos, etc.

A pesar de que el problema de sincronización de semáforos es un problema difícil de abordar, los resultados obtenidos muestran la capacidad de los algoritmos genéticos para resolver problemas de este tipo, obteniendo resultados muy buenos.

El desarrollo de algoritmos con capacidad de paralelización son fundamentales sobre todo en problemas complejos que requieren mucho poder de computo como el que se abordo. El algoritmo presentado obtuvo casi un speedup lineal.

Algo a destacar es que el algoritmo puede ser aplicado a otros lugares con solo cambiar los datos de entrada: mapa, tráfico, configuración de los semáforos y recorrido de ómnibus.

Las simulaciones demostraron su valor al dar la flexibilidad de probar distintas variantes de forma sencilla y poder crear un escenario alternativo con mejoras agregadas con las que se logran mejoras del 11 %

Los resultados de aplicar el algoritmo fueron muy satisfactorios logrando disminuir el tiempo de viaje en ómnibus por el corredor de 26,7 minutos a 23.5m aplicando solo el algoritmo y hasta 17,6m al aplicarlo sobre el escenario alternativo.

5.2. Trabajo futuro

La elaboración de los mapas para la simulación donde se incluye cambios en las rutas para que sean reconocidas por el simulador, agregado de la configuración de semáforos, agregado de las líneas y paradas de ómnibus puede ser un proceso lento y tedioso, por tanto para un futuro se podría sugerir la realización de herramientas que automaticen o agilicen este trabajo.

Los trabajos de Montana y Czerwinski (1996) y Vogel et al. (2000) proponen la adaptabilidad del algoritmo en tiempo real, aunque esto requiere del agregado de sensores

a la red puede ser un método de mejora importante sobre todo en zonas de gran densidad de tráfico.

Bibliografía

- Autoanuario. Gráficas y cuadros de valores estadísticos del mercado nacional. http://www.autoanuario.com.uy/index_mercado_graficas.html, 2014. Disponible online, consultado noviembre 2014.
- BBVAResearch. Uruguay situación automotriz. https://www.bbvarresearch.com/wp-content/uploads/2014/05/1303_SitAutomotrizUruguay_2013.pdf, 2013. Disponible online, consultado noviembre 2014.
- CEPAL. Congestión de tránsito, el problema y cómo enfrentarlo. <http://www.cepal.org/publicaciones/xml/9/13059/cue-87.pdf>, 2003. Disponible online, consultado noviembre 2014.
- CHC. Chc method. <http://neo.lcc.uma.es/mallba/easy-mallba/html/algorithms.html#chc>. Disponible online, consultado noviembre 2014.
- CORSIM. Corsim. <http://www-mctrans.ce.ufl.edu/featured/TSIS/Version5/corsim.htm>. Disponible online, consultado agosto 2014.
- CTRE. Handbook of simplified practice for traffic studies : Traffic volume counts. <http://www.ctre.iastate.edu/pubs/traffichandbook/3trafficcounts.pdf>, November 2002. Disponible online, consultado noviembre 2014.
- K. Deb. Multiobjective optimization using evolutionary algorithms. 2001.
- DiFebbraro, Giglio, y Sacco. On applying petri nets to determine optimal offsets for coordinated traffic light timings. *Proc. IEEE 5th Int. Conf. Intell. Transportation Syst.*, pages 773–778, 2002.
- M. T. E. Alba. Parallelism and evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, *IEEE Press*, 6(5):443–462, 2002.
- ElPais. Garzón: Olivera admitió errores. <http://www.elpais.com.uy/informacion/garzon-olivera-admitio-errores.html>, 2013. Disponible online, consultado febrero 2015.
- ElPais. Ana olivera: "no hay justificación, nos equivocamos con garzón". <http://www.elpais.com.uy/informacion/ana-olivera-no-justificacion-corredor.html>, 2015. Disponible online, consultado febrero 2015.
- l. R. Eugenio Bañobre. Los brt en corredores segregados como sistema óptimo de transporte urbano. <http://dialnet.unirioja.es/servlet/articulo?codigo=3119987>, 2009. Disponible online, consultado febrero 2015.

- J. J. K. G. Y. Lim y Y. S. Hong. The optimization of traffic signal light using artificial intelligence. *10th IEEE Int. Conf. Fuzzy Syst*, page 1279–1282, 2001.
- D. Goldberg. Genetic algorithms in search, optimization and machine learning. 1989.
- IMM. Intendencia municipal de monteideo. <http://www.montevideo.gub.uy>. Disponible online, consultado febrero 2015.
- IMM. Plan de movilidad urbana. http://www.montevideo.gub.uy/sites/default/files/plan_de_movilidad.pdf, Febrero 2010. Disponible online, consultado febrero 2015.
- INE. Transporte y comunicaciones. http://www.ine.gub.uy/biblioteca/uruguayencifras2014/Uruguay_en_cifras_2014_Cap_11.pdf, 2014. Disponible online, consultado enero 2015.
- Institute for Transportation & Development Policy (ITDP). Bus rapid transit planning guide. <http://www.nbrti.org/docs/pdf/ITDPBRTPlanningGuide.pdf>, Junio 2007. Disponible online, consultado febrero 2015.
- S. López, P. Hernandez, A. Hernandez, y M. Garcia. Artificial neural networks as useful tools for the optimization of the relative offset between two consecutive sets of traffic lights. *Foundations and Tools for Neural Modeling*, page 795–804, 1999.
- Mallba. Mallba library. <http://neo.lcc.uma.es/mallba/easy-mallba/index.html>. Disponible online, consultado julio 2014.
- Malva. The malva project. <http://themalvaproject.github.io/>. Disponible online, consultado julio 2014.
- M. Mitchell. An introduction to genetic algorithms. 1996.
- D. J. Montana. Strongly typed genetic programming. *Evolutionary Computation*, 3(2): 199–230, 1995.
- D. J. Montana y S. Czerwinski. Evolving control laws for a network of traffic signals. 1996.
- S. Nesmachnow. Evolución en el diseño y clasificación de algoritmos genéticos paralelos,. 2002.
- OSM. Open street map. www.openstreetmap.org/. Disponible online, consultado agosto 2014.
- L. Papaleontiou. Sumo traffic modeler. <http://sourceforge.net/projects/trafficmodeler/>. Disponible online, consultado diciembre 2014.
- J. Penner, R. Hoar, y C. Jacob. Swarm-based traffic simulation with evolutionary traffic light adaptation. 2002.
- A. K. Rathi. Urban network traffic simulation:traf-netsim program. *Journal of Transportation Engineering*, 116(6):734–743, November-December 1990.

- N. M. Rouphail, B. B. Park, y J. Sacks. Direct signal timing optimization: Strategy development and results. 2000.
- SIG IMM. Sistema de información geográfica montevideo. <http://sig.montevideo.gub.uy>. Disponible online, consultado agosto 2014.
- J. Sánchez, M. Galán, y E. Rubio. Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. 2004.
- J. Sánchez, M. Galán, y E. Rubio. Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas” Area in Santa Cruz de Tenerife. 2008.
- J. Sánchez, M. Galán, y E. Rubio. Traffic signal optimization in “La Almozara” District in Saragossa under congestion conditions, using genetic algorithms, traffic microsimulation, and cluster computing. 2010.
- D. H. Stolfi. Optimizacion del trafico rodado en ciudades inteligentes. 2012.
- Subrayado. Expertos en tránsito: “el colapso está establecido” en montevideo. <http://www.subrayado.com.uy/Site/noticia/23835/expertos-en-transito-el-colapso-esta-establecido-en-montevideo>, 2013. Disponible online, consultado noviembre 2014.
- SUMO. Simulation of urban mobility. http://sumo.dlr.de/wiki/Main_Page. Disponible online, consultado agosto 2014.
- K. T. K. Teo, W. Y. Kow, y Y. K. Chin. Optimization of traffic flow within an urban traffic light intersection with genetic algorithm. 2010.
- TRANSYT-7F. Transyt-7f. http://mctrans.ce.ufl.edu/mct/?page_id=943. Disponible online, consultado agosto 2014.
- A. Vogel, C. Goerick, y W. von Seelen. Evolutionary algorithms for optimizing traffic signal operation. 2000.