
SISTEMA DE OPTIMIZACIÓN DE TRÁFICO VEHICULAR APLICADO A LA GLORIETA SANTA FE

Munguía Torres Iván Agustín

Trabajo de Tesis para obtener el grado de Maestría en Ciencias con especialidad en
Matemáticas Industriales y Computación

Asesores: Dr. Ignacio Barradas Bribiesca y Dr. Salvador Botello Rionda

Centro de Investigación en Matemáticas A.C.

Guanajuato Gto. México

16 de diciembre de 2016

ÍNDICE GENERAL

Agradecimientos	1
Introducción	2
1 Preliminares.....	3
1.1 Motivación	3
1.2 El problema del tráfico vehicular	3
1.3 Antecedentes	4
1.3.1 Estrategias aisladas y coordinadas	4
1.3.2 Estrategias de respuesta al tráfico	4
1.4 Propuesta	6
2 Modelación de tráfico vehicular.....	7
2.1 Simulación microscópica	7
2.2 Modelo de movimiento vehicular	8
2.2.1 Frecuencia de cálculo.....	8
2.2.2 Cálculo de posiciones	8
2.3 Modelos aleatorios	9
2.3.1 Variabilidad de vehículos	9
2.3.2 Comportamiento de vehículos	10
2.3.3 Toma de decisiones de rutas	11
2.3.4 Inserción y extracción de vehículos al escenario	11
2.4 Modelo de flujo	11
2.5 Modelo constructivo.....	12
2.5.1 Calles.....	12
2.5.2 Curvas	14
2.5.3 Cruceros	17
2.5.4 Elementos compuestos.....	18
2.5.5 Vehículos	23
2.5.6 Semáforos	24

2.6	Modelo de colisiones y ajuste de velocidades.....	25
2.6.1	Colisiones y velocidades en calles y curvas	25
2.6.2	Colisiones y velocidades en cruceos	26
3	Construcción de escenarios viales.....	28
3.1	Requerimientos.....	28
3.1.1	Coordenadas geográficas	28
3.1.2	Tipos y velocidades de vehículos	29
3.1.3	Rutas y frecuencia de inserción de vehículos	29
3.2	Traza de calles curvas y cruceos.....	30
3.3	Conexiones y flujo entre elementos	31
3.4	Datos de salida de la simulación	32
4	Optimización de escenarios viales	37
4.1	Definición matemática del problema de optimización.....	37
4.2	Algoritmos de descenso	38
4.3	Algoritmos genéticos.....	39
4.4	Problema de tiempos de procesamiento	40
4.5	Sincronización de semáforos.....	40
4.6	Descripción del algoritmo	41
4.6.1	Población inicial.....	41
4.6.2	Función objetivo	42
4.6.3	Función de cruceo.....	42
4.6.4	Función de selección.....	42
4.6.5	Función de mutación.....	43
4.7	Función real vs función estadística	44
4.7.1	Método 1: Escenario fijo.....	44
4.7.2	Método 2: Estimador de la media	45
4.7.3	Ventajas y desventajas entre los métodos	45
5	Aplicación: La Glorieta Santa Fe	47
5.1	La Glorieta Santa Fe.....	47
5.2	Obtención de la imagen satelital	48
5.3	División del escenario en sus elementos básicos	49

5.4	Definición de elementos básicos en el software.....	52
5.4.1	Definición de calles.....	53
5.4.2	Definición de curvas	54
5.4.3	Construcción de cruceros	56
5.4.4	Archivo escenario esc	57
5.5	Definición de flujos en el escenario	58
5.5.1	Elementos generadores	58
5.5.2	Definición de rutas.....	60
5.6	Definición de semáforos en el escenario.....	62
5.7	Generación de automóviles en el escenario	63
5.8	Simulación del escenario.....	63
5.8.1	Software Reportador De Tráfico.....	63
5.8.2	Software Analizador De Tráfico.....	68
5.9	Optimización del escenario	70
5.9.1	Software Optimizador De Tráfico	70
5.9.2	Reducción de dimensionalidad	70
5.9.3	Reducción del espacio de búsqueda.....	71
5.9.4	Definición de archivo opt.....	72
5.9.5	Operación de Optimizador De Tráfico	72
5.9.6	Búsqueda de parámetros	76
5.9.7	Hardware disponible y tiempos de procesamiento	79
5.9.8	Optimización mediante estimador de la media.....	82
5.9.9	Escenario fijo vs estimador de la media	82
5.10	Visualización de escenario	84
5.10.1	Archivos necesarios para la animación.....	84
5.10.2	Operación de software SimTraf	85
5.10.3	Zoom y desplazamiento	87
6	Resultados y conclusiones.....	89
6.1	Resultados analíticos	89
6.2	Resultados visuales	89
6.3	Validación de resultados	91

6.4	Conclusión.....	91
7	Apéndice.....	93
7.1	Software utilizado	93
7.1.1	GNU g++	93
7.1.2	OpenMP	93
7.1.3	CodeGear™ C++Builder® 2009	93
7.1.4	GIMP.....	93
7.1.5	Google maps	93
7.1.6	Git	93
7.2	Constantes utilizadas	94
7.3	Tabla de figuras	95
8	Bibliografía.....	96

AGRADECIMIENTOS

A Dios

Romanos 11:36 Porque de El, por El y para El son todas las cosas. A El sea la gloria para siempre. Amén.

A mi esposa, por su incondicional apoyo.

A mis hijos, por su paciencia todos los días de no poder jugar con ellos al elaborar este trabajo.

Al Dr. Barradas, por no quitar el dedo del renglón.

Al Dr. Botello, por creer en este proyecto.

A Chuche, por facilitarme la infraestructura necesaria para realizar este trabajo.

INTRODUCCIÓN

El presente trabajo consiste en una herramienta de análisis de tráfico vehicular que permite simular alteraciones en las condiciones viales para encontrar configuraciones óptimas en un escenario dado. El propósito de este laboratorio de análisis es replicar escenarios reales de una ciudad con el fin de probar diferentes configuraciones de sentido de calles, tiempos de semáforos, configuraciones de cruces y ampliaciones de carriles. Una vez escogidas diferentes configuraciones se procede a analizar todas las posibilidades para diferentes cargas de tráfico y así obtener la configuración que más favorece el flujo vehicular.

1 PRELIMINARES

1.1 MOTIVACIÓN

Una de las ventajas de tener una formación científica es que el mundo exterior siempre es observado bajo los lentes de la ciencia. Al encontrarse un problema en la vida cotidiana nuestra mente usualmente se hace preguntas: ¿Cuál es la causa de este problema? ¿Cómo lo puedo resolver? ¿Cómo lo puedo resolver más rápido y mejor que la solución actual?

Al manejar mi automóvil diariamente rumbo al Centro de Investigación y al estar estancado en los usuales embotellamientos viales, no pude evitar hacerme preguntas científicas. ¿Por qué estoy atorado en este tráfico? Porque los automóviles que queremos pasar somos demasiados. ¿Hay algo que se pueda hacer al respecto? Claro, ampliar las calles, poner semáforos, poner agentes de tránsito o construir rutas alternas. ¿Hay algo que yo pueda hacer al respecto, que resuelva el problema mejor que la solución actual? El presente trabajo es la respuesta a esta última pregunta.

1.2 EL PROBLEMA DEL TRÁFICO VEHICULAR

El tráfico vehicular es un problema que afecta a todos los sectores de la sociedad. Los embotellamientos ocasionan pérdidas de tiempo y dinero. Además, incrementan la contaminación atmosférica y el estrés en los individuos. Cada ciudad trata el problema de tráfico de una manera diferente, según el tamaño de la ciudad, el presupuesto con el que cuenta, y la disposición de las autoridades de resolver el problema.

Estos tres factores; tamaño, presupuesto y disposición hacen una gran diferencia en la solución de ciudad a ciudad. Grandes ciudades, como la ciudad de México, invierten en sistemas de control sofisticados donde se construyen centrales de control que regulan tiempos de semáforos de acuerdo al flujo de calles en tiempo real. Otras ciudades de menor tamaño optan por sistemas de control más sencillos, donde se realiza un análisis con software especializado, pero el sistema no se retroalimenta con datos en tiempo real.

Ciudades con menor presupuesto, al no tener acceso a los sistemas especializados optan por regular el tráfico de manera manual. Los agentes de tránsito regulan el tráfico presencialmente. Los tiempos de semáforo son establecidos de acuerdo a la experiencia del agente. Se experimenta con cambios en el sentido de las calles, ampliación de carriles y construcción de glorietas con el fin de aligerar el problema de tráfico.

Este último es el caso de la ciudad de Guanajuato Gto. y muchas otras, donde el presupuesto no es suficiente para implementar sistemas de control sofisticados y por lo tanto las

decisiones de control de tráfico son el resultado de la experiencia de individuos y no de un software de optimización. El alcance de este trabajo es proporcionar una herramienta computacional para este tipo de ciudades. La solución que se propone no es construir nuevas rutas, ni adquirir equipo de control sofisticado. La propuesta es ofrecer una herramienta que, a partir de las condiciones y el equipo existente en las rutas viales, encuentre configuraciones que mejoren el flujo vial actual.

1.3 ANTECEDENTES

Mucho trabajo científico y comercial se ha realizado para resolver el problema de tráfico. Como se mencionó anteriormente, existen en el mercado soluciones sofisticadas que miden en tiempo real las condiciones viales. A continuación, se presenta una breve descripción de varias estrategias y software existentes.

1.3.1 ESTRATEGIAS AISLADAS Y COORDINADAS

Las **estrategias aisladas** ofrecen soluciones de tráfico para cruceros de manera individual. Esto es, consideran un cruce y encuentran la configuración y tiempos de semáforos adecuados a las condiciones locales, sin considerar las repercusiones a los cruceros vecinos. Ejemplos de algoritmos en esta categoría son el algoritmo SIGSET que busca minimizar el tiempo total de espera de los automóviles, y el algoritmo SIGCAP que maximiza la capacidad vehicular de las intersecciones.

Por otro lado, se tienen las **estrategias coordinadas** las cuales buscan aligerar el tráfico globalmente. Estas estrategias optimizan el flujo de tráfico vehicular a través de múltiples cruceros y rutas viales generando por ejemplo las olas verdes donde vehículos en rutas principales viajan sin detenerse a través de varios cruceros gracias a la coordinación en los tiempos de semáforos. El Boulevard López Mateos en la Ciudad de León Gto. es un ejemplo de una ruta vial que presenta este fenómeno.

Ejemplos de algoritmos en esta categoría son el algoritmo MAXBAND que busca encontrar el desfase óptimo entre los semáforos de las rutas para maximizar el número de vehículos que cruzan sin parar a través de ellos. MULTIBAND es otro algoritmo derivado de MAXBAND que además considera diferentes anchos de calles y vueltas a la izquierda en la ruta. TRANSYT es un algoritmo iterativo de minimización con heurísticas de búsqueda local (hill-climb) que minimiza el número de paradas de los vehículos.

1.3.2 ESTRATEGIAS DE RESPUESTA AL TRÁFICO

Las estrategias presentadas en la sección anterior son consideradas **estrategias de tiempo fijo**. Esto es, las optimizaciones se hacen basadas en datos históricos de flujo vehicular considerando las condiciones de flujo en la hora pico. Una vez realizada la optimización las señales de tránsito son configuradas con un tiempo fijo permaneciendo así por varios meses. Las desventajas de este tipo de estrategias es que el tráfico no es óptimo en horas

con flujo vehicular ligero. Además de que no consideran cambios en el flujo vehicular dependiendo de la hora del día, el día de la semana (días hábiles o fin de semana) la época del año (vacaciones o días laborales) o eventos especiales en la ciudad que alteren el flujo vehicular.

Debido a esto se han desarrollado **estrategias de respuesta al tráfico**, las cuales realizan optimizaciones en tiempo real considerando como datos de entrada las observaciones puntuales de cada una de las entradas de tráfico mediante sensores como cámaras de video, mangueras neumáticas, lazos inductivos y sensores piezoeléctricos.

Las estrategias de respuesta al tráfico también se subdividen en aisladas y coordinadas. En las estrategias aisladas, las mediciones realizadas en un cruce específico son enviadas a la señal de tráfico, la cual ajusta la duración de sus estados (verde-rojo) para maximizar el flujo vehicular. El algoritmo MOVA es un ejemplo de este tipo de estrategias. Este algoritmo comienza con un tiempo verde inicial, el cual se incrementa dependiendo del número de vehículos observados en la dirección controlada por el semáforo.

Las estrategias coordinadas operan con más complejidad. Todos los datos de tráfico recolectados por los diversos sensores en diferentes cruces son enviados a la central de control, la cual mediante software especializado realiza la optimización en tiempo real. Una vez realizada la optimización, se envía el resultado a las señales de tránsito para ajustar la duración de sus estados a las nuevas condiciones de tráfico. Los algoritmos que entran en esta categoría son SCOOT, SCATS, RHODES, TUC y MOTION, entre otros.

Una variante de esta última estrategia es hacer el control de tráfico descentralizado. Es decir, la optimización no se realiza en una central sino en estaciones distribuidas a lo largo de la red vial que se comunican entre sí. Ejemplos de algoritmos en esta variante son UTOPIA, PROLYN y OPAC.

Las soluciones obtenidas por las estrategias de respuesta al tráfico son claramente superiores ya que toman en cuenta el estado de tráfico en cada momento. El lado negativo de estas estrategias es el elevado costo de implementación. Estas estrategias requieren la compra, instalación, configuración y mantenimiento de los sensores y señales de tráfico, la construcción y adecuación de una central de control, la compra y mantenimiento de los sistemas de cómputo (hardware y software) especializados, la contratación del personal calificado para operar los sistemas, y la implementación de infraestructura de comunicación entre la central de control y los sensores y señales de tráfico.

Las ciudades, al enfrentarse con los elevados costos en los sistemas de control toman decisiones de acuerdo al presupuesto. Ciudades grandes pueden costear soluciones de última tecnología. Sin embargo, las ciudades medianas y pequeñas tienen que optar por soluciones más sencillas como lo son las estrategias aisladas de tiempo fijo.

1.4 PROPUESTA

El propósito de este trabajo es proporcionar una herramienta de control de tráfico que supla las necesidades de ciudades medianas y pequeñas. Este trabajo presenta una solución de diseño y optimización coordinada de tiempo fijo para el tráfico vehicular. Se optó por realizar un software de tiempo fijo debido a la naturaleza del problema. Las ciudades que se quieren atender no cuentan con infraestructura de respuesta a tráfico. El objetivo es que esta herramienta represente una opción lista para su utilización sin necesidad de requerir la compra de equipo adicional al que ya se cuenta.

Si bien la herramienta de software desarrollada ya es aplicable a cualquier escenario vial de cualquier ciudad, se seleccionó realizar el análisis de la ciudad de Guanajuato en su escenario de mayor tráfico vial, como el alcance de este proyecto.

2 MODELACIÓN DE TRÁFICO VEHICULAR

La simulación de tráfico vehicular involucra gran cantidad de modelos; modelos de aleatoriedad, modelos de construcción de calles, avenidas y cruces, modelos de aceleración y velocidad, modelos de flujo y rutas de tránsito entre otros. En este capítulo se describen los diferentes estrategias y procedimientos involucrados en la simulación vehicular de este trabajo.

2.1 SIMULACIÓN MICROSCÓPICA

Existen tres metodologías para la simulación de agentes, La simulación macroscópica, microscópica y mesoscópica. La simulación macroscópica simula el flujo de tráfico como un todo, es decir, los cálculos se llevan a cabo considerando el flujo colectivo. Las principales variables de interés son las relacionadas con las características de flujo. Mucho del software en el mercado emplea esta metodología, por ejemplo, TRANSYT-7F, VISUM, FREFLO, NETVACI, TransCAD, KRONOS, AUTOS, EMME/2, METANET y METACOR. Los cálculos y la complejidad de esta metodología son relativamente simples dado que consideran las vías de flujo como un todo sin tomar en cuenta el comportamiento, velocidades y dimensiones de cada uno de los vehículos en el sistema.

En el lado opuesto se encuentra la metodología microscópica que considera a cada uno de los vehículos como una unidad independiente calculando la interacción de cada agente con todos los elementos a su alrededor. En este modelo cada agente es diferente a los demás en velocidad, aceleración, y dimensiones. Debido a esto, la complejidad y el tiempo de cálculo es mucho mayor al de los modelos macroscópicos. La ventaja radica en que los resultados son más exactos y además abre la puerta a simulaciones visuales que permiten al usuario final tomar decisiones con más conocimiento. Los paquetes de software que recurren a este tipo de simulación son INTRAS, FRESIM, MITSIM, NETSIM, CORSIM, VISSIM, THOREAU, FLEXSYT-II y AIMSUM, entre otros.

La tercera metodología es llamada modelación mesoscópica. Esta busca establecer un compromiso entre la simplicidad de cálculos del modelo macroscópico y la exactitud del modelo microscópico. Esta técnica considera las vías de flujo como un todo simulando el flujo colectivo, pero introduce diferentes comportamientos de grupos de conductores. Los paquetes de software que se encuentran en esta categoría son METROPOLIS, DYNASMART, DYNAMIT e INTEGRATION, entre otros.

La metodología seleccionada para este trabajo es el análisis microscópico debido a la exactitud que se obtiene en los resultados. Este trabajo busca proporcionar una herramienta que refleje lo mejor posible las condiciones de tráfico para así poder proponer soluciones realistas y proporcionar una herramienta visual para el usuario final.

2.2 MODELO DE MOVIMIENTO VEHICULAR

2.2.1 FRECUENCIA DE CÁLCULO

Debido a la metodología de análisis microscópico seleccionada es necesario recalculer las posiciones y velocidades de todos los vehículos varias veces por segundo. Se seleccionó recalculer de manera sobrada, 30 veces por segundo, para apegarse a los estándares de cuadros por segundo de una pantalla de computadora y así presentar al usuario una animación suave del flujo de tráfico.

Dada entonces la frecuencia de 30 cuadros por segundo, el escenario es calculado para cada instante de tiempo con incrementos de $\Delta t = \frac{1}{30}$ s. Esto es, las posiciones, velocidades e interacciones de cada uno de los vehículos en el escenario serán calculadas para el tiempo $t_0 = 0$, $t_1 = \Delta t$, $t_2 = 2\Delta t$, $t_3 = 3\Delta t \dots$ y así sucesivamente hasta concluir el tiempo de simulación deseado.

2.2.2 CÁLCULO DE POSICIONES

El cálculo de la posición y orientación de los vehículos se realiza de la siguiente manera. La posición inicial y la orientación de cada vehículo están determinadas por el carril de la calle en la que es insertado. Esto es, las coordenadas y la dirección del vehículo corresponden a las coordenadas y dirección del inicio de la calle en el carril correspondiente. La velocidad inicial del vehículo se asigna considerando el límite de velocidad de la calle y la preferencia del conductor (las preferencias del conductor se describen en la sección 2.3.2)

Una vez establecidas la posición, orientación y velocidades iniciales del vehículo, el cálculo de la posición en el siguiente tiempo se realiza en tres pasos. Primero se ajusta la velocidad v dependiendo de las condiciones viales. Por ejemplo, si el vehículo se acerca a un semáforo o a un vehículo frente a él, la velocidad de viaje se disminuye. Por otro lado, si el frente del vehículo está libre de obstáculos, la velocidad se incrementa gradualmente, de acuerdo a la aceleración específica del vehículo y a la preferencia de aceleración del conductor, hasta llegar a la velocidad deseada máxima específica para cada conductor (el cálculo de esta velocidad explica en la sección 2.6)

El segundo paso es el ajuste de la posición. Se calcula la distancia de avance d considerando la velocidad recientemente calculada y el incremento de tiempo mediante la fórmula $d = v\Delta t$. Una vez calculada esta distancia, se desplaza el vehículo d unidades a lo largo del trazo de la calle.

El tercer y último paso consiste en calcular la nueva orientación del vehículo. En caso de calles rectas la orientación del vehículo no se modifica. Para calles con curvas la nueva orientación corresponde a la recta tangente al trazo de la calle en la posición deseada en el carril correspondiente.

La actualización de las posiciones de los vehículos en el nuevo tiempo se hace de forma asíncrona, esto es, la nueva posición de un vehículo se actualiza en el sistema conforme va siendo calculada sin necesidad de esperar el resultado de los demás vehículos. Esto aligera el uso de memoria RAM en el sistema y no afecta al realismo del simulador dado que las decisiones de movimiento en la realidad son tomadas de esta manera y no hay riesgo de choque dado el pequeño paso de tiempo.

2.3 MODELOS ALEATORIOS

Para realizar una simulación que refleje el comportamiento real de los vehículos es necesario introducir aleatoriedad. A continuación, se describen las diferentes áreas en las que el simulador recurre a cálculos probabilísticos.

2.3.1 VARIABILIDAD DE VEHÍCULOS

En todo escenario real de tráfico existe diversidad de vehículos. Automóviles particulares, camionetas, microbuses y camiones. Cada elemento contribuye al tráfico vehicular de manera diferente. Por ejemplo, los camiones reducen el flujo promedio de las calles debido a su baja velocidad y pobre aceleración. Por esta razón, el simulador considera cuatro diferentes tipos de autos: vehículo chico (automóvil sedán) vehículo mediano (miniván) vehículo grande (microbús) y vehículo extra grande (camión de pasajeros). Los datos particulares de cada vehículo (ancho, largo, peso y aceleración) fueron obtenidos de los manuales de los respectivos fabricantes. A continuación, se muestran los datos particulares de cada uno de ellos.

Tipo de vehículo	Ancho(m)	Largo(m)	Peso(kg)	Aceleración ($\frac{kg}{h^2}$)
Chico	1.64	4.33	1389	24000
Mediano	2	5.15	2041	18000
Grande	2.31	5.69	3311.22	14400
Extra grande	2.6	11.58	10205.82	10285

Tabla 1: Datos particulares de los vehículos considerados

El número de vehículos de cada tipo está dado por la función de probabilidad discreta descrita en la siguiente tabla.

Tipo de vehículo	Probabilidad
Chico	0.55
Mediano	0.3
Grande	0.1

Tipo de vehículo	Probabilidad
Extra grande	0.05

Tabla 2: Función de probabilidad para los diferentes tipos de vehículos

Al insertar cada nuevo vehículo al escenario se decide aleatoriamente el tipo de vehículo mediante la función de probabilidad. Los valores asignados a esta tabla son valores empíricos. Para la implementación del simulador en un escenario específico será necesario medir físicamente estos valores mediante un análisis estadístico en campo y modificarlos en el software antes de realizar las simulaciones.

2.3.2 COMPORTAMIENTO DE VEHÍCULOS

Además de las variaciones en el tipo de vehículo, también se consideran variaciones en el comportamiento del conductor, tanto en la velocidad deseada como en la aceleración. Si bien, cada calle tiene límite de velocidad v_l , la velocidad real de cada vehículo en una calle sin tráfico varía dependiendo de los hábitos de manejo del conductor. Por ejemplo, para una calle con límite de velocidad de $60 \frac{km}{h}$, una madre que viaja con un bebe en el automóvil podría tener una preferencia de $50 \frac{km}{h}$. Sin embargo, una persona que se dirige al trabajo con prisa podría tener una preferencia de $80 \frac{km}{h}$ para la misma vía. El simulador toma este hecho en consideración y decide la velocidad deseada de cada automóvil mediante la función de probabilidad uniforme $U(v_l - \Delta v, v_l + \Delta v)$ donde Δv es un valor configurable en el software. Entonces para una velocidad límite de $v_l = 60 \frac{km}{h}$, y una variabilidad de velocidad de $\Delta v = 20 \frac{km}{h}$ la función de probabilidad será $U(40 \frac{km}{h}, 80 \frac{km}{h})$.

De igual manera, a pesar de que cada automóvil tiene una aceleración máxima a_{max} correspondiente a las limitaciones mecánicas del vehículo (por ejemplo, para el auto chico $a_{max} = 24000 \frac{km}{h^2}$), los hábitos de manejo de cada individuo cambian la aceleración deseada. El simulador asigna aceleraciones a cada vehículo con la función de probabilidad uniforme $U(a_{max} - \Delta a, a_{max})$ donde Δa , la variación de la aceleración, es también un valor configurable en el software.

Para modelar entonces diferentes comportamientos se deben coordinar la velocidad con la aceleración. Es decir, si a un conductor se le asigna una velocidad por arriba de la media, describiendo a un conductor con prisa, su aceleración debe estar también por encima de la media. Para garantizar este fenómeno el procedimiento de asignación de valores es como sigue:

Se realiza un experimento de la variable aleatoria $X \sim U(0,1)$ obteniendo el resultado x . El valor de velocidad se calcula entonces como $v = v_l + \Delta v(2x - 1)$ y el valor de aceleración como $a = a_{max} - x\Delta a$. De esta manera se garantiza que velocidades altas

corresponden a aceleraciones altas (conductores con prisa) y velocidades bajas a aceleraciones bajas (conductores precavidos).

2.3.3 TOMA DE DECISIONES DE RUTAS

Las rutas que toman los vehículos varían de escenario a escenario y de ciudad a ciudad. Es por eso que el simulador recibe como datos de entrada los siguientes parámetros:

Para cada calle de entrada al escenario se deben considerar todas las posibles rutas que los vehículos pueden tomar a partir de esa calle de entrada, esto es $r_1, r_2, r_3, \dots, r_n$. Para cada ruta se debe especificar en el software la probabilidad de que un vehículo tome esa ruta de tal manera que

$$P(r_i) \in [0,1] \quad 1 \leq i \leq n$$

$$\sum_{i=1}^n p(r_i) = 1$$

Una vez especificadas en el software todas las rutas con sus probabilidades, para todas las calles, el cálculo de la ruta de un vehículo insertado en una calle específica se realiza llevando a cabo un experimento y de la variable aleatoria $Y \sim U(0,1)$. La ruta asignada al vehículo en cuestión será la ruta r_k donde k es el mayor número tal que $\sum_{i=1}^k p(r_i) \leq y$.

2.3.4 INSERCIÓN Y EXTRACCIÓN DE VEHÍCULOS AL ESCENARIO

La inserción de vehículos se realiza en todas las calles entrantes del escenario. Como parámetros de entrada del software se deben especificar la frecuencia de inserción de vehículos en cada una de las calles. Por ejemplo, un vehículo cada segundo, un vehículo cada dos segundos, un vehículo cada medio segundo, etc. Una vez especificados esos valores se inserta un vehículo en la calle siguiendo la frecuencia especificada. En caso de que la calle de entrada esté llena, la inserción del vehículo se pone en cola de espera y se pospone hasta que la calle esté despejada para recibir al vehículo en espera. En caso de un embotellamiento, donde la calle no avanza por un periodo de tiempo, la cola de inserción continúa creciendo indefinidamente.

Después de que un vehículo ha recorrido su ruta y llega al final de una calle de salida, se considera que el vehículo ha terminado el recorrido y es extraído del sistema. Sus valores de velocidad promedio y tiempo total de viaje son almacenados para el análisis global de eficiencia del escenario que se describe en la sección 5.8.1.

2.4 MODELO DE FLUJO

Para definir el flujo de vehículos en un escenario es necesario especificar en el software las calles de entrada, las calles de salida y todas las posibles rutas entre ellas. Estas rutas consisten en una sucesión de elementos, como calles y cruces, que el vehículo en la ruta

debe seguir para llegar del elemento origen al elemento destino. Además de especificar la lista de elementos es necesario especificar también el carril en el que el vehículo debe posicionarse en cada elemento para transitar correctamente por la ruta. Se define entonces una ruta r como la sucesión de pares ordenados.

$$r = \{(e_1, c_1), (e_2, c_2), (e_3, c_3), \dots, (e_m, c_m)\}$$

Donde m es el total de elementos que se deben transitar, pudiendo ser diferente para cada ruta. El valor c_i representa el carril en la que el vehículo debe posicionarse dentro del elemento e_i . Para rutas donde es posible transitar por más de un carril, se introduce la información como dos o más rutas diferentes.

El tránsito vehicular se modela entonces de la siguiente manera. Al insertar un vehículo al sistema se selecciona aleatoriamente la ruta r correspondiente de la manera descrita en la sección 2.3.3. El carril inicial queda entonces determinado por c_1 . Se procede a avanzar el vehículo en cada paso de tiempo hasta alcanzar el final del primer elemento e_1 . Una vez en este punto el vehículo entra en el elemento e_2 posicionándose en el carril correspondiente c_2 (los detalles de cómo se realiza este cambio de carril de manera suave se describen en la sección 2.5.3). Este proceso se repite hasta alcanzar el final del último elemento e_m terminando así el recorrido de la ruta. El software realiza la simulación de todos y cada uno de los vehículos, así como de sus interacciones de manera simultánea en cada paso de tiempo, presentando así una animación realista del flujo de tráfico en el escenario dado.

2.5 MODELO CONSTRUCTIVO

La construcción de un escenario involucra especificar todos y cada uno de los elementos de flujo. En este software se modelan tres tipos de elementos de flujo, la calle, la curva y el cruce. Elementos de flujo más complejos como glorietas, boulevards, bifurcaciones y accesos son modelados a partir de estos elementos simples como se explicará la sección 2.5.4. A continuación, se presentan los detalles constructivos de estos tres elementos básicos.

2.5.1 CALLES

El primer elemento constructivo es la calle, la cual se visualiza en este trabajo como un segmento de recta direccionado de ancho variable que puede contener uno o más carriles. Definimos entonces a una calle como el vector de valores:

$$Ca = \left((x_{ini}, y_{ini}), (x_{fin}, y_{fin}), N, a \right)$$

$$x_{ini}, y_{ini}, x_{fin}, y_{fin}, a \in \mathbb{R}, N \in \mathbb{N}$$

Los pares coordenados (x_{ini}, y_{ini}) y (x_{fin}, y_{fin}) representan el punto inicial y punto final en el plano del trazo de la calle respectivamente. El valor N corresponde al número de

carriles en la calle y el valor a representa el ancho de cada carril como se ilustra en la siguiente figura.

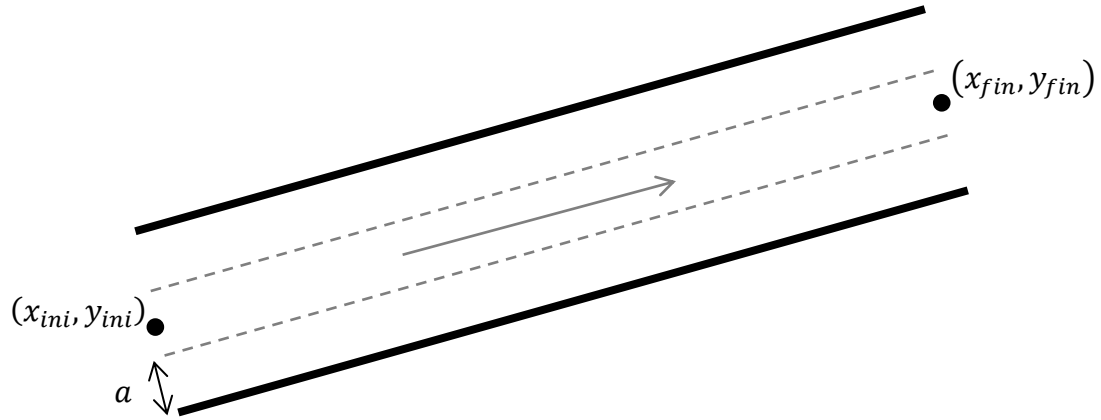


Figura 1: Calle

Para realizar el movimiento de un carro a través de la calle consideraremos el trazo de la calle como una trayectoria en el plano parametrizada por longitud de arco, es decir:

$$f: \mathbb{R} \rightarrow \mathbb{R}^2$$

$$f(d) = \left(x_{ini} + \frac{d}{l}(x_{fin} - x_{ini}), y_{ini} + \frac{d}{l}(y_{fin} - y_{ini}) \right)$$

$$f(d) = (x_{ini}, y_{ini}) + \frac{d}{l}(x_{fin} - x_{ini}, y_{fin} - y_{ini})$$

Donde l es la longitud de la calle:

$$l = |(x_{fin} - x_{ini}, y_{fin} - y_{ini})|$$

De esta manera, las posiciones de un automóvil que recorre el trazo de la calle desde $d = 0$ hasta $d = l$ están dadas por $f(d)$, siendo el punto $f(d)$ la posición del centro geométrico del automóvil (El cálculo de las posiciones de las esquinas del automóvil se describe en la sección 2.5.5). Además, el vector dirección del automóvil en cada punto de la curva está dado por la derivada de la función f evaluada en ese punto. En este caso, el vector dirección es constante, es decir, el mismo para cualquier punto en la trayectoria.

$$f'(d) = \frac{1}{l}(x_{fin} - x_{ini}, y_{fin} - y_{ini})$$

Para calles con más de un carril, se considera el carril de la derecha como el primer carril y el carril de la izquierda como el último carril. La posición del vehículo se calcula

utilizando la función trayectoria y el vector dirección antes descritos de la siguiente manera:

Consideremos una calle de $N \geq 1$ carriles de ancho a . La posición (x, y) de un vehículo, que ha avanzado una distancia $d \in [0, l]$ a lo largo de la calle en el carril $1 \leq n \leq N$ esta dada por:

$$(x, y) = f(d) + a \left(n - \frac{N + 1}{2} \right) u$$

$$u = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix} \frac{f'(d)}{|f'(d)|}$$

Donde el vector u es el vector ortogonal a $f'(d)$ de norma 1 que apunta siempre al carril de la izquierda. Este vector se calcula aplicando una matriz de rotación de 90° al vector dirección. Podemos entonces definir la trayectoria de la calle en el carril n por medio de la función:

$$\hat{f}_n(d) = f(d) + a \left(n - \frac{N + 1}{2} \right) u$$

La función de vector dirección se conserva idéntica para cualquier carril:

$$\hat{f}'_n(d) = f'(d)$$

2.5.2 CURVAS

El segundo elemento constructivo es la curva. Este elemento se introduce para replicar las condiciones sinuosas de las avenidas. Al igual que la calle, las curvas son de ancho variable y pueden contener uno o más carriles. Una avenida entonces puede ser completamente construida mediante la sucesión de calles y curvas. Definimos una curva como el vector de valores:

$$Cu = \left((x_{ini}, y_{ini}), (x_{fin}, y_{fin}), (x'_{ini}, y'_{ini}), (x'_{fin}, y'_{fin}), N, a, l \right)$$

$$x_{ini}, y_{ini}, x_{fin}, y_{fin}, x'_{ini}, y'_{ini}, x'_{fin}, y'_{fin}, a, l \in \mathbb{R}$$

$$N \in \mathbb{N}$$

Los pares coordenados (x_{ini}, y_{ini}) y (x_{fin}, y_{fin}) representan el punto inicial y punto final en el plano del trazo de la curva. Los valores (x'_{ini}, y'_{ini}) y (x'_{fin}, y'_{fin}) corresponden al vector dirección de entrada y de salida de la curva respectivamente. El valor N corresponde al número de carriles en la calle y el valor a representa el ancho de cada carril como se ilustra en la siguiente figura:

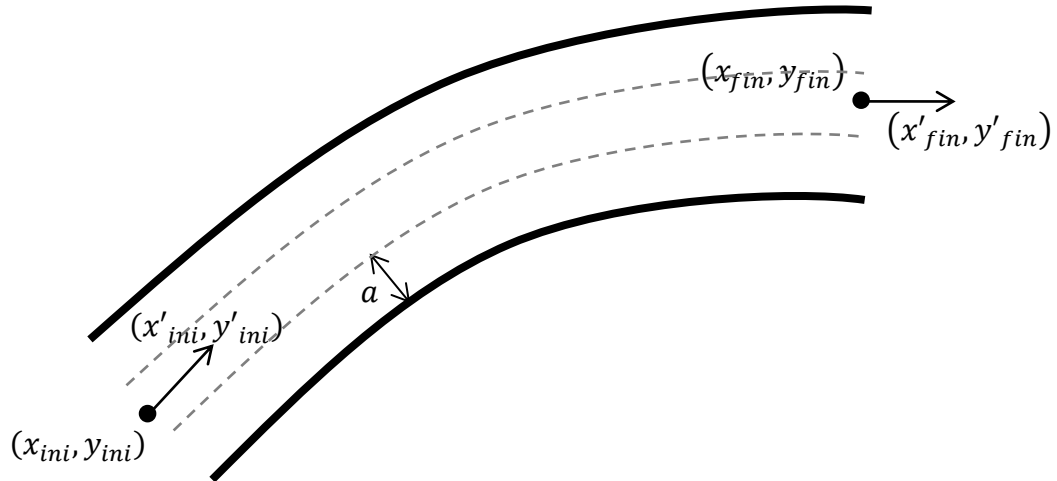


Figura 2: Curva

Para identificar de manera única a una curva es necesario proporcionar restricciones adicionales ya que existen un número infinito de curvas en el plano que cumplen con las condiciones previamente descritas. Una de estas restricciones es que la función trayectoria será un polinomio de grado tres en cada componente. Adicionalmente se introduce el parámetro de curvatura l que además de definir de manera única la curva, permitirá controlar el grado de dificultad del elemento.

La función trayectoria la escribimos de la siguiente manera:

$$f: \mathbb{R} \rightarrow \mathbb{R}^2$$

$$f(d) = (x(d), y(d))$$

Donde x , y son los polinomios de tercer grado.

$$x, y: \mathbb{R} \rightarrow \mathbb{R}$$

$$x(d) = a_3 d^3 + a_2 d^2 + a_1 d + a_0$$

$$y(d) = b_3 d^3 + b_2 d^2 + b_1 d + b_0$$

El vector dirección en cada punto queda descrito como:

$$f'(d) = (x'(d), y'(d))$$

$$x'(d) = 3a_3 d^2 + 2a_2 d + a_1$$

$$y'(d) = 3b_3 d^2 + 2b_2 d + b_1$$

Sea 0 el punto inicial de la trayectoria y sea l el punto final tal que $d \in [0, l]$ representa todos los puntos en la curva. El cálculo de los coeficientes se realiza sustituyendo los valores conocidos de posición y dirección en los puntos iniciales y finales de la curva.

$$(x_{ini}, y_{ini}) = f(0) = (a_0, b_0)$$

$$(x'_{ini}, y'_{ini}) = f'(0) = (a_1, b_1)$$

$$(x_{fin}, y_{fin}) = f(l) = (x(l), y(l)) = (a_3l^3 + a_2l^2 + a_1l + a_0, b_3l^3 + b_2l^2 + b_1l + b_0)$$

$$(x'_{fin}, y'_{fin}) = f'(l) = (x'(l), y'(l)) = (3a_3l^2 + 2a_2l + a_1, 3b_3l^2 + 2b_3l + b_1)$$

De la primera y segunda ecuación se obtiene el valor de cuatro coeficientes:

$$a_0 = x_{ini}$$

$$a_1 = x'_{ini}$$

$$b_0 = y_{ini}$$

$$b_1 = y'_{ini}$$

Sustituyendo estos valores en la tercer y cuarta ecuación obtenemos

$$(x_{fin}, y_{fin}) = (a_3l^3 + a_2l^2 + x'_{ini}l + x_{ini}, b_3l^3 + b_2l^2 + y'_{ini}l + y_{ini})$$

$$(x'_{fin}, y'_{fin}) = (3a_3l^2 + 2a_2l + x'_{ini}, 3b_3l^2 + 2b_3l + y'_{ini})$$

Igualando cada una de las componentes se obtienen los siguientes dos sistemas de ecuaciones de dos incógnitas cada uno:

$$a_3l^3 + a_2l^2 + x'_{ini}l + x_{ini} = x_{fin}$$

$$3a_3l^2 + 2a_2l + x'_{ini} = x'_{fin}$$

$$b_3l^3 + b_2l^2 + y'_{ini}l + y_{ini} = y_{fin}$$

$$3b_3l^2 + 2b_3l + y'_{ini} = dy_{fin}$$

Resolviendo los sistemas se obtienen el valor de los cuatro coeficientes restantes

$$a_3 = \frac{x'_{ini} + x'_{fin}}{l^2} - 2 \frac{x_{fin} - x_{ini}}{l^3}$$

$$a_2 = \frac{x'_{ini} - x'_{fin} - 3a_3 l^2}{2l}$$

$$b_3 = \frac{y'_{ini} + y'_{fin}}{l^2} - 2 \frac{y_{fin} - y_{ini}}{l^3}$$

$$b_2 = \frac{y'_{ini} - y'_{fin} - 3b_3 l^2}{2l}$$

De esta manera la curva Cu queda definida de manera única. Además, la función $f(d)$ describe las posiciones del vehículo a lo largo del trazo de la curva para $d \in [0, l]$. El vector dirección en cada punto queda determinado por $f'(d)$.

Para curvas con más de un carril, la posición del vehículo se calcula definiendo la función trayectoria:

$$\hat{f}_n(d) = f(d) + a \left(n - \frac{N+1}{2} \right) u$$

Donde N es el número de carriles de la curva, a el ancho del carril y n el carril que ocupa el automóvil. La función de vector dirección se conserva idéntica para cualquier carril:

$$\hat{f}'_n(d) = f'(d)$$

2.5.3 CRUCEROS

El último elemento constructivo es el crucero, el cual se visualiza en este trabajo no como un elemento físico en el escenario sino como un elemento lógico, un conector. Un crucero entonces no tiene dimensiones ni localización en el plano, un crucero consiste en un conjunto de conexiones entre elementos de entrada y elementos de salida del crucero.

Sea U el conjunto de todas las calles y curvas en el plano. Definimos una conexión entre dos elementos $e_1, e_2 \in U$ como el vector

$$C_{e_1, e_2} = (e_1, n_1, e_2, n_2)$$

Una conexión entonces enlaza la salida del carril n_1 del elemento e_1 , con la entrada del carril n_2 del elemento e_2 como se muestra en la siguiente figura.

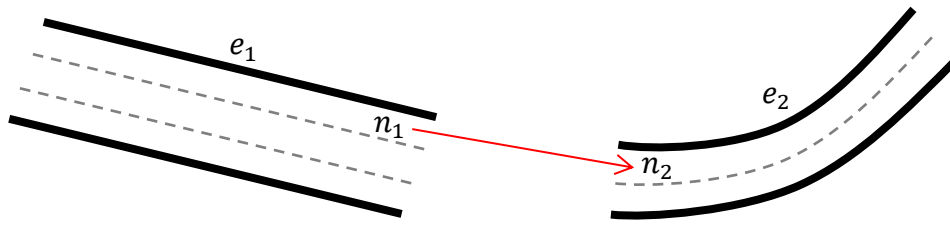


Figura 3: Conexión

Para realizar el movimiento de un vehículo a lo largo de la conexión se define un elemento simple tipo curva de un carril, que conecta el punto final de e_1 en el carril n_1 con el punto inicial de e_2 en el carril n_2 . Esta conexión está dada entonces por la curva definida por la siguiente ecuación:

$$Cu = (\hat{f}_{1n_1}(l), \hat{f}_{2n_2}(0), f'_1(l), f'_2(0), a_1, 1, l^*)$$

Donde f_1 y f_2 son las trayectorias de e_1 y e_2 en los carriles n_1 y n_2 respectivamente, l^* es un parámetro de curvatura fijo obtenido experimentalmente y a_1 es el ancho de carril de e_1 (es indistinto si se toma el ancho de e_1 o de e_2 ya que la curva es de un solo carril)

Consideremos ahora \mathcal{C} como el conjunto de todas las posibles conexiones entre elementos en U . Definimos un crucero Cr como un subconjunto del conjunto de conexiones:

$$Cr \subset \mathcal{C}$$

Es claro que no cualquier subconjunto de \mathcal{C} representa un escenario realista, de hecho, no cualquier conexión de acuerdo a esta definición representa una conexión posible en la vida real, sin embargo, todo crucero realista puede representarse como un subconjunto de \mathcal{C} . Queda en el usuario del software desarrollado, implementar escenarios que tengan sentido suministrando las especificaciones adecuadas. El propósito de esta definición de crucero es simplemente proporcionar una enunciación formal de este elemento para poder ilustrar y construir con certeza cualquier configuración de calles curvas y cruceros en la vida real. La siguiente sección muestra ejemplos de estas configuraciones geométricas.

2.5.4 ELEMENTOS COMPUESTOS

A partir de estos tres elementos simples es posible construir escenarios complejos que describen situaciones viales reales. A continuación, se presentan ejemplos de varios de ellos.

BOULEVARD

Un boulevard, o cualquier otra calle de doble sentido de uno o más carriles, se construye mediante dos elementos calle en sentidos opuestos como se muestra en la figura

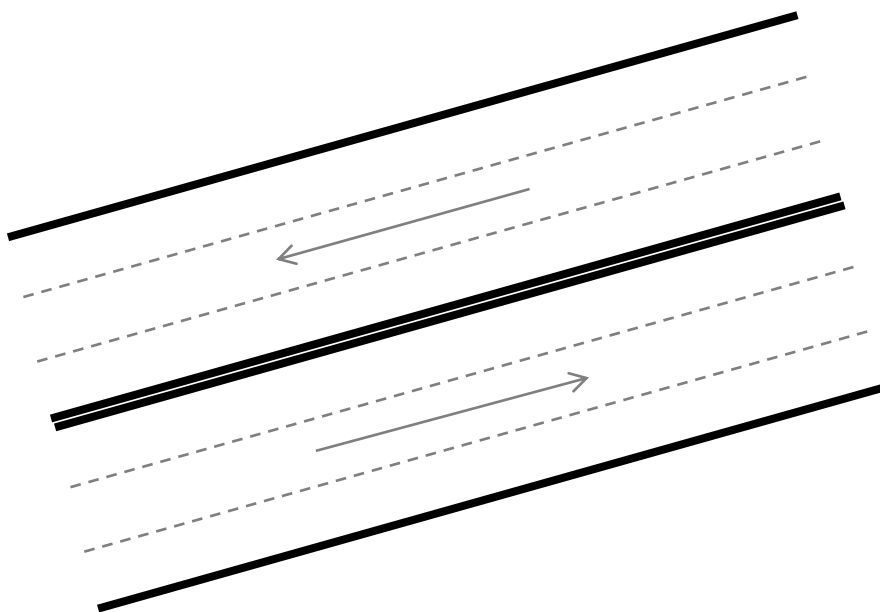


Figura 4: Boulevard

CAMINO SINUOSO

Un camino sinuoso se construye como combinación de calles y curvas del mismo número de carriles. A continuación, se presenta un ejemplo de un camino sinuoso y su descomposición en elementos simples.

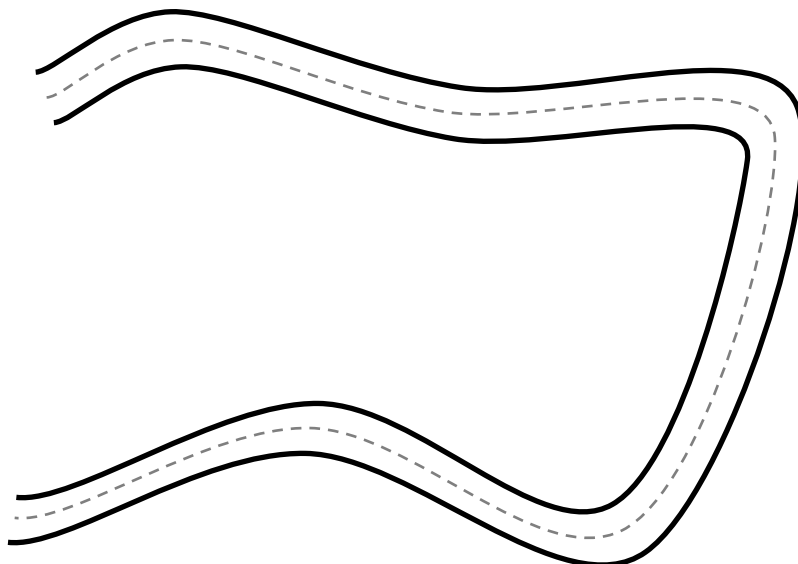


Figura 5: Camino sinuoso

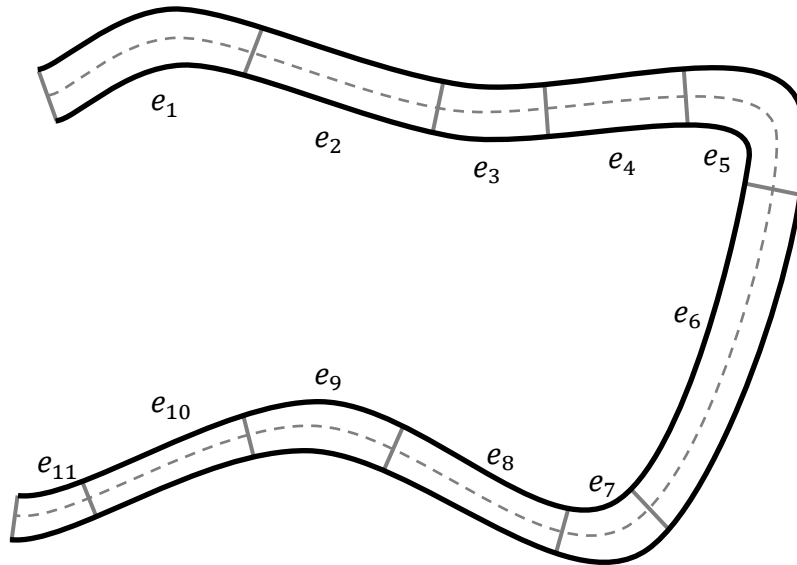


Figura 6: Descomposición de camino sinuoso en elementos simples

Es claro entonces que cualquier camino puede descomponerse en calles y curvas. Constructivamente esto se logra haciendo coincidir el punto final y vector dirección final de cada elemento con el punto inicial y vector dirección inicial del elemento siguiente. El movimiento vehicular en estas composiciones se realiza avanzando el automóvil por toda la trayectoria del elemento e_1 , una vez terminada la trayectoria, el vehículo continúa con la trayectoria del elemento siguiente y así sucesivamente. Cabe mencionar que el movimiento vehicular es suave ya que los vectores dirección del elemento actual y el elemento siguiente coincide por construcción.

BIFURCACIÓN

Una bifurcación consiste en la división de un camino. Este elemento compuesto se puede representar como la combinación de tres calles y un crucero como se muestra en la siguiente figura.

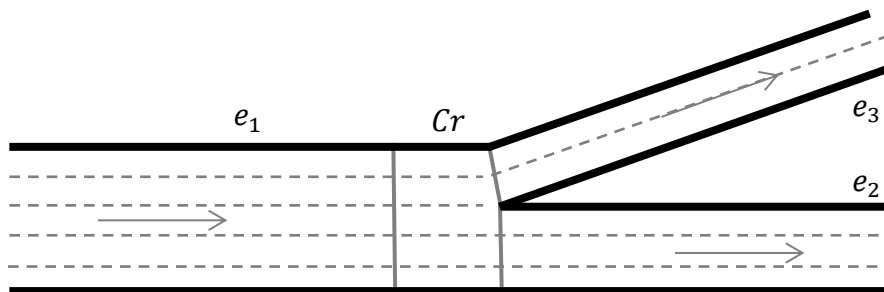


Figura 7: Bifurcación

Se tiene entonces la calle e_1 de cinco carriles que se dividen en dos calles. La e_2 de tres carriles y la e_3 de dos carriles. El crucero Cr se define como el conjunto de las siguientes cinco conexiones:

$$Cr = \{(e_1, 1, e_2, 1), (e_1, 2, e_2, 2), (e_1, 3, e_2, 3), (e_1, 4, e_3, 1), (e_1, 5, e_3, 2)\}$$

Como se puede observar, la función del crucero es conectar cada uno de los cinco carriles de e_1 con sus respectivos carriles destino en e_2 y e_3 .

SALIDA DE AUTOPISTA

La salida de autopista similar a la bifurcación se representa como combinación de tres calles y un crucero como se muestra en la figura.

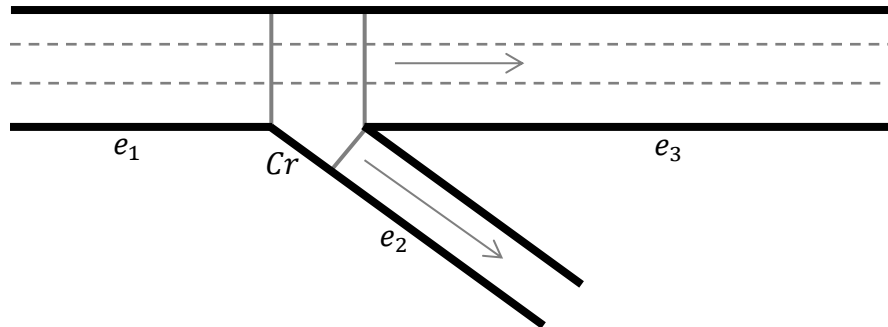


Figura 8: Salida de autopista

La diferencia entre una bifurcación y una salida radica en que el número de carriles de salida es mayor al número de carriles de entrada, por lo tanto, el crucero se define de la siguiente manera:

$$Cr = \{(e_1, 1, e_2, 1), (e_1, 1, e_3, 1), (e_1, 2, e_3, 2), (e_1, 3, e_3, 3)\}$$

Un vehículo transitando en el primer carril del elemento e_1 tiene entonces dos opciones: tomar el primer carril del elemento e_2 o tomar el primer carril del elemento e_3 . La toma de decisión dependerá de la ruta asignada al vehículo como se explicó en la sección 2.3.3

GLORIETA

Entre los elementos más complejos se encuentran las glorietas. Su construcción involucra la combinación de varias calles, curvas y cruceros. A continuación, se muestra un ejemplo de una glorieta de dos carriles con tres calles de entrada y tres de salida.

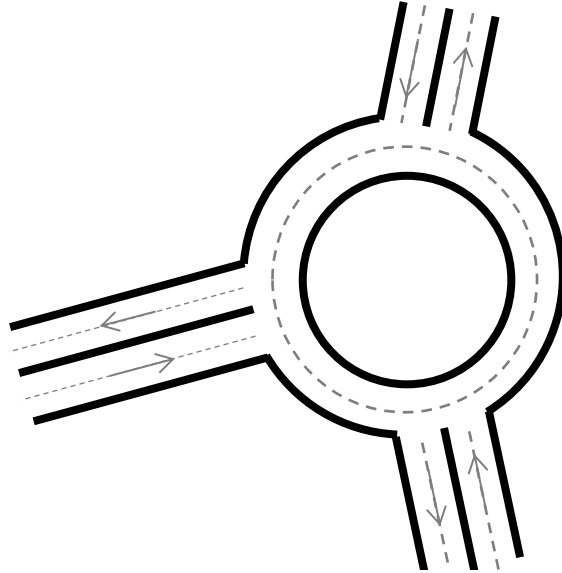


Figura 9: Glorieta

La descomposición de la glorieta en sus elementos simples se detalla en la siguiente figura.

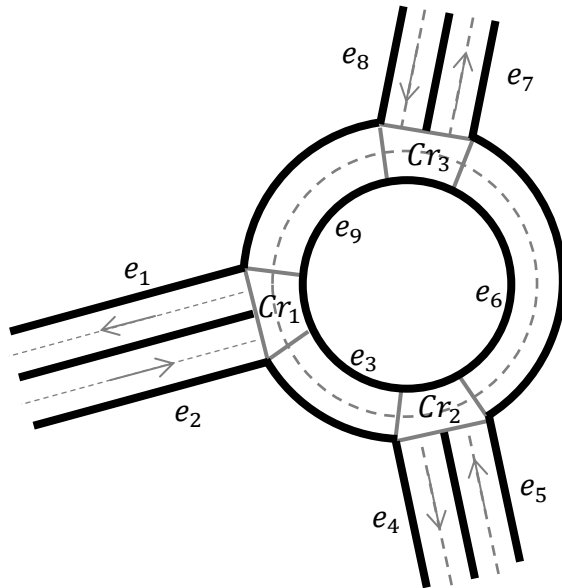


Figura 10: Descomposición de glorieta en elementos simples

Cada uno de los tres cruces conecta dos elementos de entrada con dos elementos de salida. A continuación, se presenta la definición del cruce Cr_1 ; los cruces Cr_2 y Cr_3 se construyen de manera análoga.

$$Cr_1 = \{(e_9, 2, e_3, 2), (e_9, 1, e_3, 1), (e_9, 1, e_1, 1), (e_9, 2, e_1, 2), (e_2, 1, e_3, 1), (e_2, 2, e_3, 2)\}$$

INTERSECCIÓN

Las intersecciones son probablemente los elementos de tráfico más complicados ya que reparten el tráfico de varias calles en varias direcciones. A pesar de la dificultad vial que representa una intersección, ésta puede ser representada con elementos simples, como se ilustra en la siguiente figura.

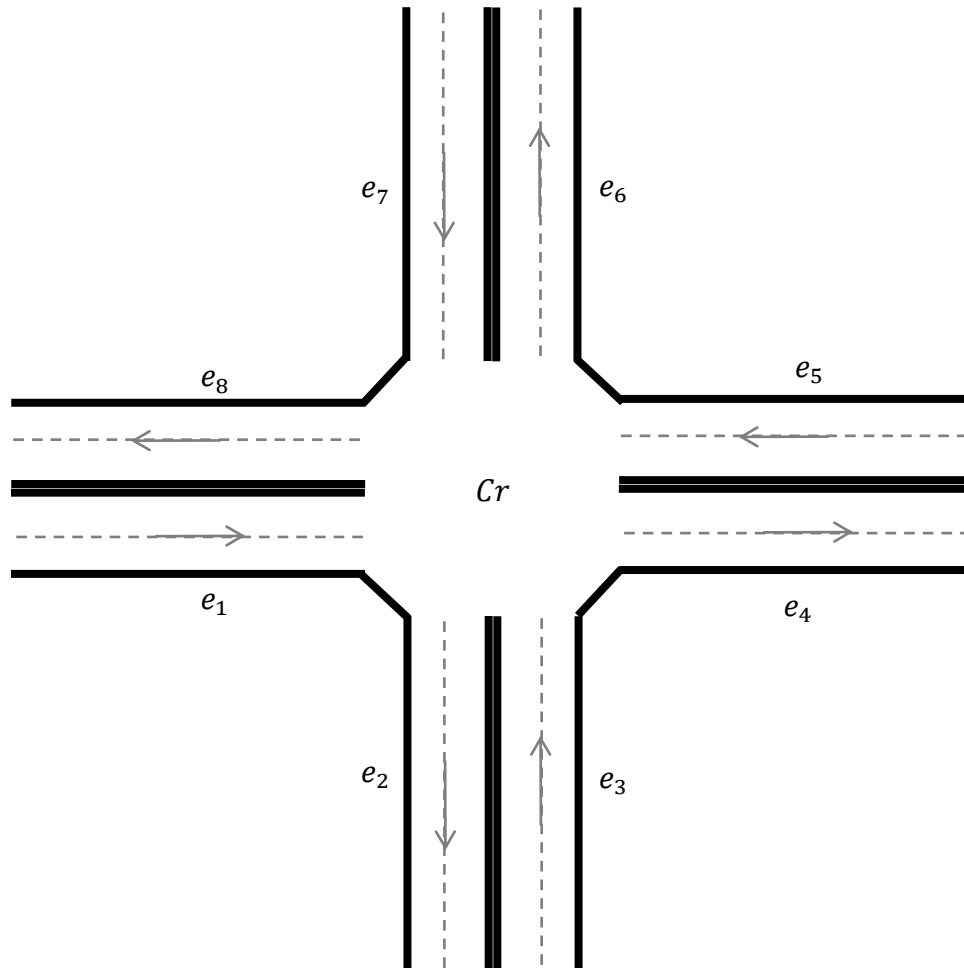


Figura 11: Intersección de 8 calles

Para esta intersección el cruceo Cr se define de la siguiente manera:

$$Cr = \left\{ \begin{array}{l} (e_1, 1, e_2, 1), (e_1, 1, e_4, 1), (e_1, 2, e_4, 2), (e_1, 2, e_6, 1) \\ (e_3, 1, e_4, 1), (e_3, 1, e_6, 1), (e_3, 2, e_6, 2), (e_3, 2, e_8, 1) \\ , (e_5, 1, e_6, 1), (e_5, 1, e_8, 1), (e_5, 2, e_8, 2), (e_5, 2, e_2, 1) \\ , (e_7, 1, e_8, 1), (e_7, 1, e_2, 1), (e_7, 2, e_2, 2), (e_7, 2, e_4, 1) \end{array} \right\}$$

Queda entonces claro que con los elementos simples descritos en esta sección es posible construir elementos complejos de tal manera que cualquier escenario real puede ser replicado utilizando solamente calles curvas y cruceos.

2.5.5 VEHÍCULOS

Un elemento fundamental del modelo constructivo es el vehículo. Un vehículo es representado como un polígono en el plano que se mueve de acuerdo a la función trayectoria del elemento en el que se encuentre. Para este trabajo representaremos a un vehículo como un rectángulo en el plano cuya orientación depende de la función trayectoria. Como se mencionó en la sección 2.3.1 el ancho a y largo l de un vehículo varía de acuerdo a una función de probabilidad, por lo tanto, definimos un vehículo como el vector:

$$V = (a, l) \in \mathbb{R}^2$$

La posición del centro del vehículo y su dirección, dada la distancia d recorrida a lo largo del elemento simple, quedarán determinadas por las funciones $\hat{f}_n(d)$ y $f'(d)$ respectivamente. El cálculo de las cuatro esquinas $V_1, V_2, V_3, V_4 \in \mathbb{R}^2$ del rectángulo que representa al vehículo se realiza de la siguiente manera.

Sea $V_c = \hat{f}_n(d) \in \mathbb{R}^2$ el centro del vehículo y sea u el vector de norma 1 ortogonal a la dirección del vehículo (ver definición en la sección 2.5.1). Las esquinas del rectángulo están dadas por:

$$\begin{aligned} V_1 &= V_c + \left(\frac{l}{2} \frac{f'(d)}{|f'(d)|}, \frac{a}{2} u \right) \\ V_2 &= V_c + \left(\frac{l}{2} \frac{f'(d)}{|f'(d)|}, -\frac{a}{2} u \right) \\ V_3 &= V_c + \left(-\frac{l}{2} \frac{f'(d)}{|f'(d)|}, -\frac{a}{2} u \right) \\ V_4 &= V_c + \left(-\frac{l}{2} \frac{f'(d)}{|f'(d)|}, \frac{a}{2} u \right) \end{aligned}$$

Con el cálculo de las esquinas del vehículo, queda completamente definida la posición del rectángulo del vehículo para cualquier elemento simple, en cualquier carril n , y en cualquier posición d a lo largo del elemento.

2.5.6 SEMÁFOROS

El último elemento del modelo constructivo es el semáforo. Para este trabajo se consideran semáforos de dos tiempos, verde y rojo. Un semáforo se define entonces como el vector de valores:

$$S = (t_v, t_r, r, s) \in \mathbb{R}^4$$

Los valores t_v, t_r corresponden a la duración en segundos del tiempo verde y rojo, respectivamente. El valor $r < t_v$ es el tiempo de retraso de inicio del estado verde. En este tiempo r el semáforo permanece en rojo para permitir a los vehículos en el cruce terminar su recorrido. Una vez terminado el tiempo r , el semáforo cambia a verde y permanecerá en verde los restantes $t_v - r$ segundos. El semáforo S tiene entonces el siguiente ciclo de 3 tiempos:

Tiempo	Duración
Espera inicio verde	r
Verde	$t_v - r$
Rojo	t_r

Tabla 3: Ciclo de semáforo

El ciclo del semáforo se repite indefinidamente hasta terminar la simulación. El valor s es el desfase en segundos respecto al tiempo inicial t_0 . Un semáforo con $s = 0$ comenzará su tiempo “Espera inicio verde” en el tiempo t_0 . Un semáforo con $s = 5$ comienza su tiempo “Espera inicio verde” cinco segundos después del tiempo inicial t_0 . Este desfase es útil para sincronizar varios semáforos en un cruce. Por ejemplo, para dos semáforos que controlan el tráfico en una intersección con dos calles entrantes, se requiere que cuando uno esté en su estado verde, el otro esté en su estado rojo y viceversa. El desfase será útil también para la implementación de las olas verdes donde el semáforo siguiente debe comenzar su estado verde s segundos después que el anterior.

En un escenario real se asignan semáforos a las diferentes calles y curvas al final de sus trayectorias, de acuerdo a la distribución espacial real. Los vehículos que transitan por los elementos que tiene semáforo asignado, modifican su velocidad de acuerdo a los vehículos delante de ellos, al estado del semáforo y a la distancia del vehículo al semáforo.

2.6 MODELO DE COLISIONES Y AJUSTE DE VELOCIDADES

La detección de colisiones es parte fundamental en el modelado microscópico de vehículos. La velocidad de movimiento de cada agente es directamente afectada por la cercanía a otros vehículos en su periferia. Esta medida de cercanía varía de situación a situación como se describe a continuación.

2.6.1 COLISIONES Y VELOCIDADES EN CALLES Y CURVAS

En una carretera los vehículos viajan a altas velocidades sin importar la cercanía a otros vehículos en los carriles laterales. Para replicar este fenómeno, la detección de choques dentro de un elemento calle o curva depende solamente de los vehículos que se encuentran en el mismo carril del vehículo en cuestión. Definimos la distancia de colisión frontal d de un vehículo como sigue: Sea f la función de trayectoria del elemento de longitud l en el que transitan los vehículos $V_1 = (a_1, l_1)$ y $V_2 = (a_2, l_2)$ ambos en el mismo carril n . Sean

$\hat{f}_n(d_1), \hat{f}_n(d_2)$ las posiciones de estos vehículos a lo largo de la trayectoria del elemento. Sea V_2 el vehículo más cercano a V_1 que cumple que $d_1 < d_2$. La distancia de V_1 a V_2 está dada por $d = d_2 - d_1 - \frac{l_1 + l_2}{2}$. Esta distancia representa la distancia entre la parte frontal del V_1 y la parte trasera de V_2 .

Se define una distancia de reacción d_r que corresponde a la distancia en la cual un conductor comienza a frenar el vehículo. Se define también d_{min} como la distancia mínima permitida entre dos vehículos, Estos parámetros son configurables en el software (Para este trabajo se considera $d_r = 8$ metros y $d_{min} = 0.5$ metros).

Definimos ahora la velocidad de colisión v_c para el vehículo V_1 en función de d_r y d_{min} de la siguiente manera:

$$v_c = \begin{cases} v_l, & d > d_r \\ v_l \frac{d - d_{min}}{d_r - d_{min}}, & d_{min} \leq d \leq d_r \\ 0, & d < d_{min} \end{cases}$$

Donde v_l es la velocidad límite del vehículo, descrita en la sección 2.3.2. Esta velocidad de colisión representa la velocidad deseada máxima del vehículo que garantiza evitar colisiones con el vehículo enfrente de él.

Para calles o curvas con semáforos en rojo, se calcula la velocidad de semáforo v_s como:

$$v_s = \begin{cases} v_l, & d_s > d_r \\ v_l \frac{d_s}{d_r}, & 0 \leq d_s \leq d_r \end{cases}$$

$$d_s = l - d_1 - \frac{l_1}{2}$$

Esta velocidad de semáforo representa la velocidad deseada máxima del vehículo que garantiza detenerse antes del cruce cuando el semáforo está en rojo. En el caso de calles o curvas con semáforo en verde o sin semáforo, la velocidad de semáforo se considera igual a la velocidad límite del vehículo $v_s = v_l$.

Una vez calculadas estas velocidades. La velocidad a la que transitará el vehículo V_1 será $v = \min(v_c, v_s)$ dado que esta velocidad garantiza un frenado adecuado para evitar colisiones.

2.6.2 COLISIONES Y VELOCIDADES EN CRUCEROS

En un cruce la detección de colisiones es más delicada. Dado que la naturaleza del cruce es recibir carros de varias direcciones, un conductor tiene precaución, no solo de automóviles enfrente de él sino también de vehículos aproximándose en diferentes

direcciones. El cálculo de colisiones para un cruce involucra el cálculo de la distancia d correspondiente a la distancia mínima entre la defensa frontal del vehículo a todas las esquinas y lados de los demás vehículos. Una vez calculada esta distancia d , el cálculo de la velocidad v que garantiza un frenado adecuado para evitar colisiones se realiza de la misma manera que en la sección anterior.

En este capítulo se presentaron todos los modelos involucrados en la simulación vehicular. En el siguiente capítulo se describe cómo todos estos modelos son combinados para construir un escenario virtual que replique las condiciones viales de manera realista.

3 CONSTRUCCIÓN DE ESCENARIOS VIALES

En este capítulo se describe el procedimiento para construir un escenario vial en el simulador. Se describe el proceso de construcción desde la especificación de calles y curvas hasta la configuración de semáforos y flujos vehiculares.

3.1 REQUERIMIENTOS

Antes de construir un escenario en el simulador es necesario que el usuario cuente con conocimiento físico del escenario en cuestión. Las siguientes secciones describen a detalle la información que se necesita recopilar para la construcción del escenario en el simulador.

3.1.1 COORDENADAS GEOGRÁFICAS

El primer dato que se requiere son las coordenadas geográficas del escenario y sus elementos, es decir, se debe conocer las coordenadas espaciales de los puntos de interés involucrados como calles curvas, glorietas, bifurcaciones, etc. Para facilitar esta tarea, el software provee una herramienta en la que, dada una imagen de la vista aérea del escenario, y dada la escala de la imagen, el software proporciona las coordenadas espaciales de cualquier punto en la imagen mediante una interfaz gráfica (Figura 12). La escala de la imagen puede ser obtenida de mediciones en campo o como en este caso, utilizando la herramienta de medición de *Google Maps* (Figura 13).

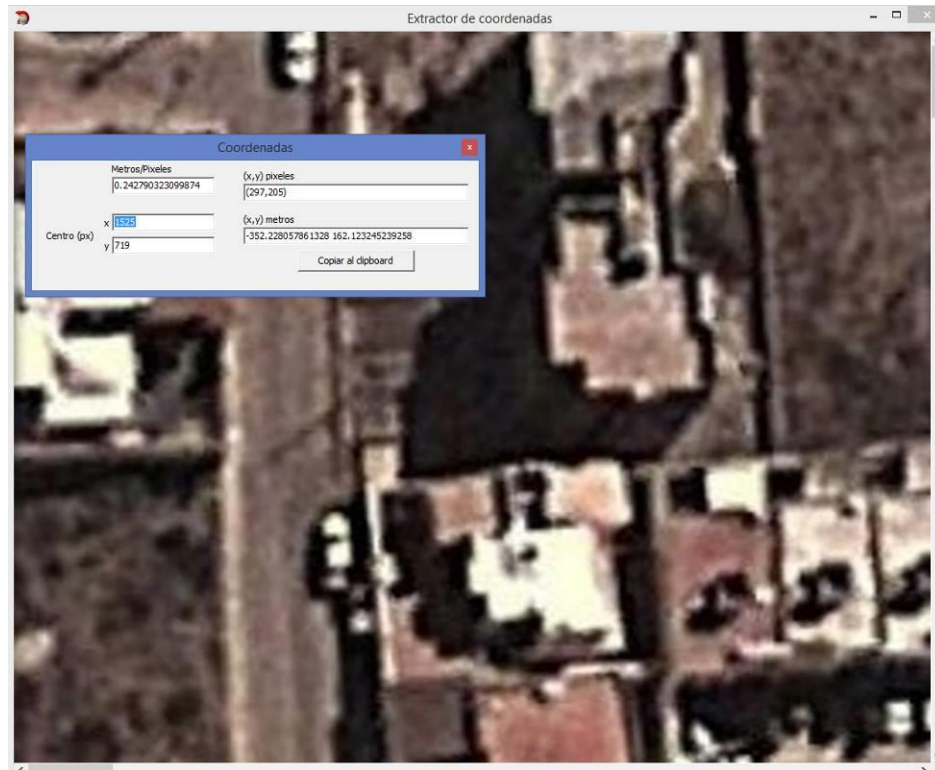


Figura 12: Herramienta de extracción de coordenadas



Figura 13: Medición de referencia inicial de *Google Maps*

3.1.2 TIPOS Y VELOCIDADES DE VEHÍCULOS

El segundo requerimiento es conocer la distribución del tipo de vehículos descrita en la sección 2.3.1 en el escenario de estudio. Es decir, qué porcentaje de vehículos chicos, medianos, grandes y extra grandes transitan por el escenario vial en cuestión. También se requiere conocer los límites de velocidad v_l de las vías del escenario y la variación de velocidad Δv , descritas en la sección 2.3.2, de los vehículos que transitan por el mismo.

3.1.3 RUTAS Y FRECUENCIA DE INSERCIÓN DE VEHÍCULOS

El siguiente requerimiento es conocer todas las posibles rutas de tráfico, así como el porcentaje de vehículos que transita por dichas rutas, como se explica en la sección 2.3.3. Así mismo, para todas las vías que insertan tráfico al escenario, es necesario conocer la frecuencia de inserción de vehículos promedio, a las diferentes horas de interés.

La obtención de los requerimientos descritos en las secciones 3.1.2 y 3.1.3 usualmente se realiza mediante un estudio de campo. Es decir, un equipo de individuos acude al sitio de interés a realizar las mediciones estadísticas correspondientes. Estas mediciones pueden realizarse con ayuda de dispositivos automatizados como radares de velocidad y contadores de tráfico vehicular como mangueras neumáticas, lazos inductivos y sensores piezoeléctricos. En caso de no contar con estos requerimientos es posible operar el software con datos empíricos para el estudio de diferentes fenómenos como aumento de flujo en calles, horas pico, olas verdes etc. Sin embargo, no se recomienda la toma de decisiones e implementación final de resultados con datos artificiales.

3.2 TRAZO DE CALLES CURVAS Y CRUCEROS

Una vez compilada la información en los requerimientos se procede a hacer el trazo del escenario. Es tarea del usuario determinar los elementos simples que conforman el escenario, así como la adecuada descomposición de elementos complejos como glorietas, intersecciones y bifurcaciones en elementos simples, como se describe en la sección 2.5.4.

Una vez determinados todos los elementos simples se procede a definirlos en el software. Para calles se requieren, las coordenadas (x, y) del inicio y fin de la calle, el número de carriles y el ancho de cada carril. Para las curvas, adicional a los datos anteriores, se requiere el vector dirección de entrada y salida de la calle y el parámetro de curvatura. El software provee una vista del escenario sobrepuesto a la imagen satelital para la adecuada elección de este último parámetro (ver Figura 15). Para la definición de los cruces es necesario proporcionar la lista de elementos calle o curva entrantes y las conexiones de carriles entre cada uno de ellos. Estos datos se proveen mediante un archivo de entrada descrito en la sección 5.4.4.

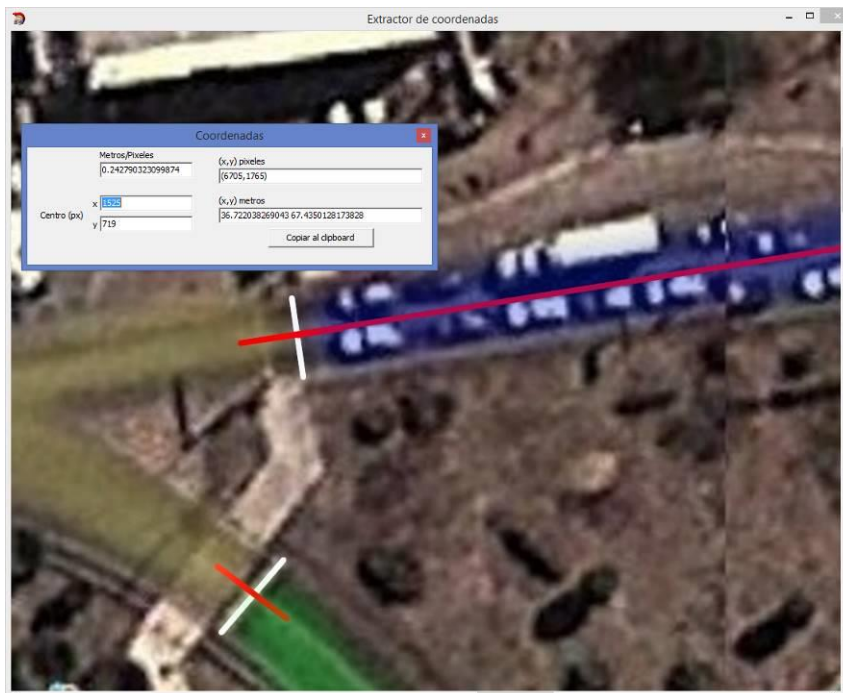


Figura 14: Vista de imagen satelital para extracción de coordenadas



Figura 15: Vista de curva sobrepuesta a imagen satelital

3.3 CONEXIONES Y FLUJO ENTRE ELEMENTOS

Habiendo ya definido todos los elementos simples se procede a definir las conexiones entre ellos. Estas conexiones son importantes para los algoritmos de colisión que deben buscar cercanía con vehículos no solo dentro del elemento en cuestión sino también en los elementos viales cercanos. Esto se realiza especificando los elementos anterior y siguiente de cada calle y curva en el escenario.

Para definir el flujo vehicular se deben especificar los elementos generadores de flujo. Esto es, los elementos que insertan vehículos al escenario. Para cada elemento generador se especifica su frecuencia de inserción de vehículos para los diferentes tiempos de la simulación. Así mismo, se introducen en el software todas y cada una de las rutas de tránsito y sus probabilidades, especificando la sucesión de elementos y carriles asociados.

Finalmente, se definen los semáforos introduciendo los tiempos verde, rojo, retraso y desfase de cada uno de ellos.

3.4 DATOS DE SALIDA DE LA SIMULACIÓN

Una vez especificado el escenario, el software queda configurado y listo para realizar las simulaciones de flujo vehicular. El software entonces, en cada paso de tiempo, realiza la inserción de vehículos y el cálculo de colisiones, velocidades, nuevas posiciones y direcciones de cada vehículo en el escenario. Una vez terminada la simulación se reportan los resultados. Los datos de salida que se obtienen al final de la simulación son los siguientes:

Reportes generales

- Reporte de la cantidad de vehículos insertados y vehículos restantes en el escenario minuto a minuto.
- Vehículos totales insertados
- Vehículos restantes en el escenario
- Distancia en metros de cada una de las rutas
- Velocidad promedio de vehículos (global, por elemento origen y por ruta)
- Tiempo promedio de los vehículos en el escenario (global, por elemento origen y por ruta)

Reportes de vehículos que terminaron el recorrido

- Número de vehículos que terminaron el recorrido (global, por elemento origen y por ruta)
- Tiempos promedio (global y por ruta) de vehículos que terminaron el recorrido
- Velocidad promedio de vehículos que terminaron la ruta (global, por elemento origen y por ruta)

- Distribución de vehículos que terminaron el recorrido (por elemento origen y por ruta),

A continuación, se presenta como ejemplo uno de los reportes generados por el software:

Cálculo a 30 cuadros por segundo)

Minutos totales de animación=20

Tiempo (minutos)= 1, Vehículos insertados= 85:

Vehículos en escenario= 77, Vehículos terminados= 8, vehículos en cola= 0

Tiempo (minutos)= 2, vehículos insertados= 170:

Vehículos en escenario= 111, Vehículos terminados= 59, vehículos en cola= 0

Tiempo (minutos)= 3, vehículos insertados= 255:

Vehículos en escenario= 139, Vehículos terminados= 116, vehículos en cola= 0

Tiempo (minutos)= 4, vehículos insertados= 340:

Vehículos en escenario= 158, Vehículos terminados= 182, vehículos en cola= 0

Tiempo (minutos)= 5, vehículos insertados= 425:

Vehículos en escenario= 179, Vehículos terminados= 245, vehículos en cola= 1

Tiempo (minutos)= 6, vehículos insertados= 510:

Vehículos en escenario= 182, Vehículos terminados= 303, vehículos en cola= 25

Tiempo (minutos)= 7, vehículos insertados= 595:

Vehículos en escenario= 190, Vehículos terminados= 368, vehículos en cola= 37

Tiempo (minutos)= 8, vehículos insertados= 680:

Vehículos en escenario= 187, Vehículos terminados= 438, vehículos en cola= 55

Tiempo (minutos)= 9, vehículos insertados= 765:

Vehículos en escenario= 186, Vehículos terminados= 502, vehículos en cola= 77

Tiempo (minutos)= 10, vehículos insertados= 850:

Vehículos en escenario= 192, Vehículos terminados= 568, vehículos en cola= 90

Tiempo (minutos)= 11, vehículos insertados= 935:

Vehículos en escenario= 211, Vehículos terminados= 616, vehículos en cola= 108

Tiempo (minutos)= 12, vehículos insertados= 1020:

Vehículos en escenario= 212, Vehículos terminados= 686, vehículos en cola= 122

Tiempo (minutos)= 13, vehículos insertados= 1105:

Vehículos en escenario= 208, Vehículos terminados= 751, vehículos en cola= 146

Tiempo (minutos)= 14, vehículos insertados= 1190:

Vehículos en escenario= 197, Vehículos terminados= 821, vehículos en cola= 172

Tiempo (minutos)= 15, Vehículos insertados= 1275:
 Vehículos en escenario= 194, Vehículos terminados= 882, Vehículos
 en cola= 199

Tiempo (minutos)= 16, Vehículos insertados= 1360:
 Vehículos en escenario= 199, Vehículos terminados= 945, Vehículos
 en cola= 216

Tiempo (minutos)= 17, Vehículos insertados= 1445:
 Vehículos en escenario= 203, Vehículos terminados= 1009,
 Vehículos en cola= 233

Tiempo (minutos)= 18, Vehículos insertados= 1530:
 Vehículos en escenario= 214, Vehículos terminados= 1063,
 Vehículos en cola= 253

Tiempo (minutos)= 19, Vehículos insertados= 1615:
 Vehículos en escenario= 205, Vehículos terminados= 1133,
 Vehículos en cola= 277

Tiempo (minutos)= 20, Vehículos insertados= 1700:
 Vehículos en escenario= 214, Vehículos terminados= 1191,
 Vehículos en cola= 295

***Tiempo promedio global: 2.5913
 Minutos totales (vehículos terminados): 3099.42
 Minutos totales (vehículos restantes): 489.62
 Minutos totales (vehículos en cola): 816.167
 Minutos totales: 4405.21
 Número de vehículos (terminados): 1191
 Número de vehículos (restantes): 214
 Número de vehículos (en cola): 295
 Número de vehículos: 1700

Tiempos promedio (minutos) de vehículos que terminaron la ruta
 Tiempos por ruta:
 0.599877 0.908908 1.66641 1.30312 1.51444
 0.528274 1.62364 2.32824 1.91961 1.54167
 3.80047 2.66733 3.28091 1.87643 3.1813
 4.19963 5.07582 3.28189 5.23289 x

Tiempo promedio global (Vehículos que terminaron la ruta):
 2.60237

Número de vehículos que terminaron la ruta
 Por ruta:
 18 114 32 110 1
 56 65 6 62 1
 105 10 134 133 3
 92 100 144 5 x

Número global de vehículos que terminaron la ruta: 1191
 Vehículos totales insertados: 1700
 Vehículos restantes en el escenario: 509

Distancias de rutas (metros):
 517.761 747.253 712.087 783.716 781.696
 391.485 670.221 631.784 703.413 700.366
 852.169 688.764 603.304 673.264 675.531
 694.805 962.116 442.294 996.927 x

Velocidad promedio (km/h) de vehículos que terminaron la ruta
 Velocidad por ruta:

53.829 50.0426 26.5466 37.6662 30.9696
 47.4845 14.0758 16.5695 11.3914 27.2575
 7.6549 18.0803 12.6989 4.51481 12.9915
 3.54131 5.46666 11.8515 7.48561 x

velocidad promedio global: 18.5111

Distribución de vehículos que terminaron la ruta

Por ruta:

0.0151134 0.0957179 0.0268682 0.0923594 0.000839631
 0.0470193 0.054576 0.00503778 0.0520571 0.000839631
 0.0881612 0.00839631 0.11251 0.111671 0.00251889
 0.077246 0.0839631 0.120907 0.00419815 x

Distribución de vehículos que terminaron la ruta

Por elemento origen:

0.0654545 0.414545 0.116364 0.4 0.00363636
 0.294737 0.342105 0.0315789 0.326316 0.00526316
 0.272727 0.025974 0.348052 0.345455 0.00779221
 0.269795 0.293255 0.422287 0.0146628 x

Número de vehículos que terminaron la ruta

Por elemento origen:

275
 190
 385
 341

***Tiempo promedio global: 2.5913

Además, se obtiene el archivo de posiciones de cada vehículo en cada paso de tiempo en la simulación. Este archivo es utilizado para la animación gráfica y análisis visual del flujo vehicular en el escenario como se muestra en la Figura 16.



Figura 16: Visualización de archivo de posiciones

El propósito de estos datos de salida es proporcionar al usuario mediciones del flujo vehicular en el escenario que le auxilien a una mejor toma de decisiones sobre modificaciones físicas en el escenario. Estas mediciones son solo posibles gracias al simulador ya que se consideran datos de velocidad y tiempo de todos los vehículos en el escenario. La obtención de resultados similares en campo requeriría medir la velocidad promedio y el tiempo de cada uno de los vehículos que transitan en el escenario.

En este capítulo se presentaron los requerimientos necesarios para la correcta construcción de un escenario vial y los datos de entrada requeridos para una adecuada simulación. En el siguiente capítulo se presenta la manera en que se optimiza el flujo vehicular en un escenario dado.

4 OPTIMIZACIÓN DE ESCENARIOS VIALES

En este capítulo se describe el procedimiento y algoritmos utilizados para la optimización de un escenario. Es decir, dado un escenario se presentará la manera de modificarlo de tal manera que el flujo vehicular sea óptimo.

4.1 DEFINICIÓN MATEMÁTICA DEL PROBLEMA DE OPTIMIZACIÓN

Una vez construido un escenario se requiere optimizar el flujo vehicular. Esto es, modificar los elementos que definen el escenario para aligerar el flujo de tráfico. Para lograr este objetivo es necesario responder con detalle a las siguientes preguntas: ¿Qué elementos son los que se van a modificar? ¿De qué manera específica se van a modificar? y ¿Qué significa aligerar el tráfico? Es decir, ¿Qué medida o valor, después de realizar la simulación, es el que se quiere minimizar o maximizar?

Todos los elementos en el escenario son modificables, sin embargo, no todas las modificaciones son igual de viables. Por ejemplo, si bien modificar el número de carriles en las calles es posible, realizar el cambio en campo es bastante costoso y no siempre factible. La misma dificultad se presenta al querer cambiar glorietas por cruceros. Por esta razón, el elemento del escenario que se seleccionó para ser modificado de manera automática por un algoritmo de optimización es el semáforo; específicamente sus tiempos de desfase s y periodos verde y rojo (t_v y t_r). El algoritmo de optimización implementado proporcionará el ajuste óptimo de tiempos de semáforo para cada uno de los escenarios de prueba. Si se desea experimentar con modificaciones constructivas como incremento de número de carriles, cambios de dirección de calles y construcción de vías alternas, será necesario definir diferentes escenarios y comparar los resultados.

En cuanto a la definición de aligerar el tráfico, es necesario considerar una medida global que beneficie a todos los vehículos que transitan en el escenario y que además sea independiente de la ruta que transitan. La medida que se seleccionó es el tiempo promedio de viaje de todos los vehículos involucrados en el escenario. Esta medida es adecuada ya que, además de las características antes mencionadas, penaliza tiempos muertos en embotellamientos. Otra ventaja de esta medida es que al ser unidimensional facilita la selección y convergencia del método de optimización.

El problema de optimización queda entonces definido de la siguiente manera: Se requiere minimizar el tiempo promedio de viaje de todos los vehículos, variando los tiempos s , t_v y t_r de todos los semáforos en el escenario.

Se puede ver el análisis completo del escenario durante todo el tiempo de simulación como una evaluación de una función h donde el parámetro de entrada es la lista de tiempos s , t_v y t_r de los m semáforos en el escenario, y el parámetro de salida w es el tiempo promedio de viaje de todos los vehículos después de realizar la simulación. Nuestra función a optimizar queda entonces definida como:

$$h: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$h(z) = w$$

$$z = (z_1, z_2, z_3, \dots, z_n), \quad n = 3m$$

$$(z_i, z_{i+1}, z_{i+2}) = (s, t_v, t_r), i \in \{1, 4, 7, \dots, n - 2\}$$

Habiendo definido entonces la función a optimizar, la siguiente sección presentará la estrategia de minimización que permitirá encontrar la configuración ideal de semáforos para el escenario.

4.2 ALGORITMOS DE DESCENSO

De la sección anterior se obtuvo una función que permite calificar la eficiencia de un escenario dado. La función recibe como entrada la configuración de los semáforos y arroja un valor que asigna una calificación al escenario, en este caso el tiempo promedio de viaje. La estrategia de optimización busca entonces encontrar el escenario que minimice la función h , es decir, encontrar tiempos de semáforo tales que el tiempo promedio de viaje sea mínimo.

Para lograr encontrar esta configuración óptima se necesita una estrategia de minimización. Existen dos tipos de estrategias, los algoritmos de descenso y los algoritmos genéticos. Los algoritmos de descenso encuentran el mínimo de la siguiente manera: Dado un punto inicial en el espacio de búsqueda se encuentra iterativamente un punto mejor al inicial avanzando en la dirección que provea el máximo descenso. Se calcula entonces iterativamente una configuración de semáforos cada vez mejor a la anterior hasta cierto número de iteraciones o hasta que el nuevo escenario sea casi idéntico al anterior en valor o calificación.

Las ventajas de este tipo de métodos es que el número de iteraciones requeridas es muy bajo. Sin embargo, una de las desventajas es que se requiere una buena elección del punto inicial para garantizar que el escenario óptimo no sea un mínimo local sino el mínimo global. En escenarios simples con un número reducido de variables podría ser sencillo proveer un escenario inicial cercano al escenario óptimo, pero para escenario de gran

dimensionalidad será imposible o al menos muy difícil seleccionar una condición inicial adecuada.

Otra desventaja de los métodos de descenso es que se requiere el cálculo de las primeras derivadas de la función h . En algunos métodos se pide incluso realizar el cálculo de las segundas derivadas. Para la función h que se ha considerado en este trabajo el cálculo analítico de estas derivadas no es posible ya que la función no proviene de una expresión sino de una medición. Existen métodos numéricos de cálculo de derivadas, sin embargo, cada derivada direccional requiere de una o más evaluaciones de la función, dependiendo del método. Conforme el vector de entrada aumenta de dimensión, el número de derivadas a calcular en cada iteración crece de manera exponencial.

Adicionalmente al número de evaluaciones por iteración, se tiene el problema del costo de cada evaluación de la función h . Recordemos que el valor de salida de la función es el tiempo promedio de todos los vehículos que transitaron en el escenario durante todo el período de medición. Así que la evaluación de cada escenario puede llevar desde algunos segundos hasta minutos, dependiendo de la complejidad del escenario. Métodos que requieren de primeras y segundas derivadas ocuparan gran número de evaluaciones de la función por iteración, por lo que el cálculo de las matrices de derivadas para cada iteración puede llevar demasiado tiempo.

4.3 ALGORITMOS GENÉTICOS

Como alternativa a los algoritmos de descenso existen los algoritmos genéticos. Estos algoritmos tienen la ventaja de no requerir el cálculo de las derivadas de la función. Tampoco requieren de un punto inicial cercano a la solución óptima. Para los algoritmos genéticos se define una población inicial en el espacio de búsqueda. Mediante una función objetivo se asigna una calificación a cada elemento de la población. En nuestro caso, la función h fungirá como función objetivo. Se procede a realizar las operaciones de selección, cruce y mutación para producir una nueva generación. Después de varias iteraciones se selecciona el individuo con la menor calificación como el escenario óptimo.

Los algoritmos genéticos también presentan desventajas. La primera consiste en que se requiere un tamaño de población grande y el número de iteraciones necesarias para encontrar al individuo óptimo es usualmente elevado. Esto repercute nuevamente en el número de evaluaciones de la función. Sin embargo, al crecer la dimensión del problema, el número de evaluaciones necesarias es mucho menor que en el caso de los algoritmos de descenso. Dado que la dimensionalidad del problema presentado es directamente proporcional al número de semáforos en el escenario, el algoritmo genético presenta una opción más robusta para la optimización de escenarios con alto número de semáforos.

Otra desventaja de los algoritmos genéticos es que se requiere definir procedimientos adicionales al problema a resolver. Esto son: el método de generación de la población

inicial, la función objetivo y las operaciones de selección, cruce y mutación. Dada la definición de nuestro problema de optimización, la función objetivo ha quedado definida como la función h . Las funciones de selección, cruce y mutación se definirán en capítulos posteriores.

Considerando entonces las ventajas y desventajas de cada una de las estrategias presentadas, el método de optimización que se seleccionó para este trabajo son los algoritmos genéticos principalmente debido a que no se requiere especificar un punto inicial y además su implementación es más sencilla.

4.4 PROBLEMA DE TIEMPOS DE PROCESAMIENTO

La capacidad de los algoritmos genéticos para encontrar una buena solución es fuertemente dependiente del tamaño de la población y el número de generaciones. A mayor población e iteraciones, mejor será la solución encontrada, sin embargo, esto implica un mayor número de evaluaciones de la función h . Como se mencionó anteriormente, el tiempo de cálculo de cada evaluación puede llevar de varios segundos a minutos dependiendo de la complejidad del escenario. Por esta razón, será necesario recurrir a técnicas de programación en paralelo.

Es posible paralelizar cada una de las evaluaciones de la función internamente. Por ejemplo, se pueden calcular las colisiones de los vehículos de manera simultánea. También es posible calcular las nuevas velocidades y las nuevas posiciones de cada vehículo en paralelo. Sin embargo, dado que las evaluaciones de la función para diferentes datos de entrada son completamente independientes, correr evaluaciones completas en paralelo reduce cuellos de botella disminuyendo el tiempo total de cómputo.

Para disminuir los tiempos de cálculo y así permitir algoritmos genéticos con poblaciones e iteraciones mayores, se paralelizarán evaluaciones completas. Es decir, para cada generación el cálculo de las calificaciones de cada individuo se realizará en forma paralela utilizando hardware multi-núcleo y APIs de procesamiento en paralelo como OpenMP.

4.5 SINCRONIZACIÓN DE SEMÁFOROS

Una de las estrategias de aceleración de flujo vehicular es la sincronización de semáforos. Por ejemplo, la estrategia de la ola verde. Es importante aclarar que el algoritmo genético que se presenta en este trabajo es potencialmente capaz de encontrar este tipo de sincronizaciones como solución óptima.

Supongamos que cierta configuración de semáforos \hat{z} sincronizada a manera de ola verde es la solución óptima que minimiza el tiempo promedio de viaje de los vehículos en el sistema. Entonces, para esta configuración se cumple que

$$h(\hat{z}) \leq h(z), \forall z \in \mathbb{R}^n$$

El algoritmo genético comenzará con un subconjunto de puntos en \mathbb{R}^n . Iteración tras iteración las operaciones de cruce y mutación propondrán nuevos puntos candidatos en \mathbb{R}^n . De esta manera, el algoritmo genético está diseñado para explorar todo el campo de búsqueda y potencialmente encontrar el elemento \hat{z} que minimizan la función objetivo. Al ir explorando el dominio, el algoritmo genético encontrará soluciones cada vez más cercanas a la solución óptima \hat{z} .

4.6 DESCRIPCIÓN DEL ALGORITMO

Como se mencionó en la sección anterior, es importante tener una buena exploración del dominio. Un tamaño de población grande y un número alto de iteraciones contribuyen a una buena exploración del dominio. Este punto queda parcialmente cubierto utilizando cómputo paralelo ya que éste permite incrementar tamaños de población y número de iteraciones conservando e inclusive disminuyendo los tiempos de cálculo.

Adicional al cómputo paralelo, será necesario definir adecuadamente la función de cruce, la función de mutación y la población inicial para asegurar una buena exploración del dominio. A continuación, se describen cada una de ellas.

4.6.1 POBLACIÓN INICIAL

Dado que nuestro campo de búsqueda es \mathbb{R}^n , el generar la población inicial $I_0 \subset \mathbb{R}^n$ con una variable aleatoria uniforme para cada componente será suficiente para lograr una buena cobertura inicial del dominio. Una vez definida esta población inicial I_0 , El algoritmo genético generara poblaciones subsecuentes $I_1, I_2, I_3, \dots, I_q$ donde q es el número total de iteraciones a las que el algoritmo genético será sometido.

Es posible acotar el campo de búsqueda para una convergencia más rápida sin perder cobertura de la siguiente manera:

Notemos la definición de nuestro dominio

$$z = (z_1, z_2, z_3, \dots, z_n) \in \mathbb{R}^n, \quad n = 3m$$

$$(z_i, z_{i+1}, z_{i+2}) = (s, t_v, t_r), i \in \{1, 4, 7, \dots, n - 2\}$$

Las componentes t_v y t_r son tiempos en segundos de semáforo rojo y verde respectivamente. Queda claro que tiempos de semáforos demasiado grandes son físicamente inviables dada la molestia que causaría en los conductores por el tiempo de espera. De igual manera tiempos de semáforo muy pequeños no permitirían pasar una cantidad suficiente de vehículos ocasionando tráfico y molestia en los conductores en espera. Podemos entonces acotar los tiempos de semáforo definiendo tiempos mínimos y máximos de duración del semáforo verde y rojo:

$$t_v \in [t_{v_{min}}, t_{v_{max}}] \subset \mathbb{R}$$

$$t_r \in [t_{r_{min}}, t_{r_{max}}] \subset \mathbb{R}$$

Las componentes s representan el desfase en segundos de cada semáforo respecto al tiempo 0 de la simulación. Es posible también acotar este valor considerando que el desfase es cíclico, es decir, para cada semáforo en el escenario, su desfase s es equivalente a un desfase de $n(t_v + t_r) + s'$ donde $n \in \mathbb{Z}, s' < t_v + t_r$. Podemos entonces acotar las componentes s de tiempo de desfase de \mathbb{R} a $s' \in [0, t_{v_{max}} + t_{r_{max}}]$ para cada semáforo en el escenario.

Finalmente, si se considera que los tiempo máximo y mínimo verde deben ser igual a los tiempos máximo y mínimo rojo se tiene entonces que:

$$t_v, t_r \in [t_{v_{min}}, t_{v_{max}}]$$

$$s \in [0, 2t_{v_{max}}]$$

4.6.2 FUNCIÓN OBJETIVO

La función objetivo queda naturalmente definida como la función a optimizar $h: \mathbb{R}^n \rightarrow \mathbb{R}$ dado que el problema original a resolver es la minimización de la función h . La calificación de cada individuo $z = (z_1, z_2, z_3, \dots, z_n)$ está dada entonces por $h(z)$.

4.6.3 FUNCIÓN DE CRUZA

La definición de la función de cruce especifica la manera de combinar cualesquiera dos individuos de la población. La manera de combinar dos individuos queda especificada de manera natural como el punto medio dado que nuestra población son puntos en \mathbb{R}^n . Esto es, dados dos individuos, el punto generado de su cruce será el punto medio del segmento de recta que une a los dos puntos en cuestión. Este método de cruce se conoce en la literatura como cruce aritmética con valor lambda de 0.5.

4.6.4 FUNCIÓN DE SELECCIÓN

Para la función de cruce es necesario especificar también cómo seleccionar los dos candidatos a combinar. Para incrementar la diversidad de la población se implementó el método de torneo binario que se explicara a continuación:

Para la iteración k del algoritmo genético $0 \leq k < q$, se construyen los conjuntos I_k^A y I_k^B como copias exactas del conjunto I_k . Para cada elemento $z \in I_k$, se realiza un experimento de la variable aleatoria $X \sim Be(p_c)$. Donde $p_c \in [0,1]$ representa la probabilidad de cruce del algoritmo. Si el resultado del experimento no es éxito, se selecciona al individuo z como elemento de la nueva población \hat{I}_{k+1} . Si el resultado del experimento es éxito entonces se procede a cruzar al elemento z de la siguiente manera:

1. Seleccionamos de manera aleatoria sin remplazo un elemento de cada conjunto copia
2. Comparamos el valor de la función objetivo de cada elemento y conservamos al mejor elemento z^* . Este será el individuo seleccionado para la cruce
3. Realizamos la operación de cruce generando un nuevo individuo \hat{z} como el punto medio del segmento de recta que conecta a z y z^* obteniendo así el nuevo elemento de la población \hat{I}_{k+1}

Repitiendo el procedimiento anterior para cada elemento de I_k , generamos un nuevo conjunto \hat{I}_{k+1} . El método de torneo binario permite entonces generar combinaciones aleatorias de los elementos en la población, siempre dando preferencia a individuos con menor valor de función objetivo. A continuación, se procede a aplicar la operación mutación para así obtener la nueva generación I_{k+1} .

4.6.5 FUNCIÓN DE MUTACIÓN

La función de mutación es una operación que se realiza sobre un subconjunto de elementos de la nueva población candidato \hat{I}_{k+1} . El propósito de la mutación es realizar un cambio significativo en los elementos, de tal manera que se garantice una buena exploración del dominio. La función de cruce seleccionada presenta la desventaja de generar la nueva población siempre dentro de la envolvente convexa de la población anterior por lo que puntos fuera de la envolvente convexa nunca son explorados. La función de mutación seleccionada compensa esta desventaja al recalcular aleatoriamente algunas de las componentes del elemento de la siguiente manera:

Para cada elemento $z \in \hat{I}_{k+1}$, se realiza un experimento de la variable aleatoria $X \sim Be(p_m)$. Donde $p_m \in [0,1]$ representa la probabilidad de mutación del algoritmo. Si el resultado del experimento es éxito entonces se procede a mutar el elemento z de la siguiente manera:

Dado que z , un elemento de la población, está definido como:

$$z = (z_1, z_2, z_3, \dots, z_n) \in \mathbb{R}^n, \quad n = 3m$$

$$(z_i, z_{i+1}, z_{i+2}) = (s, t_v, t_r), i \in \{1, 4, 7, \dots, n - 2\}$$

Se calculan las nuevas componentes del vector de la siguiente manera:

$$(\hat{z}_i, \hat{z}_{i+1}, \hat{z}_{i+2}) = (b_s u_s + (1 - b_s) z_i, b_v u_v + (1 - b_v) z_{i+1}, b_r u_r + (1 - b_r) z_{i+2})$$

$$i \in \{1, 4, 7, \dots, n - 2\}$$

Donde b_s, b_v, b_r son el resultado de un experimento de las variables aleatorias $B_s, B_v, B_r \sim Be(p_{mc})$ respectivamente, donde $p_{mc} \in [0,1]$ representa la probabilidad de

mutación por componente, y donde u_s, u_r, u_v son el resultado de un experimento de las respectivas variables aleatorias:

$$B_s \sim U(0, t_{v_{max}} + t_{r_{max}})$$

$$B_v \sim U(t_{v_{min}}, t_{v_{max}})$$

$$B_r \sim U(t_{r_{min}}, t_{r_{max}})$$

Queda entonces definida la mutación como el cambio aleatorio de las componentes del vector con probabilidad p_{mc} para un subconjunto de la población seleccionado aleatoriamente con probabilidad p_m . Las probabilidades p_m y p_{mc} son datos de configuración del algoritmo genético proporcionados por el usuario.

Hemos definido entonces de manera completa el algoritmo genético mediante la definición de la población inicial y las funciones cruza, selección y mutación. Es importante resaltar que el algoritmo implementado no tiene un mecanismo de preservación de los mejores individuos, por lo que el software de optimización se encarga de llevar un registro del mejor individuo encontrado en cada iteración para así poder recuperar el mejor individuo global una vez finalizado el algoritmo genético.

4.7 FUNCIÓN REAL VS FUNCIÓN ESTADÍSTICA

Los métodos de optimización descritos anteriormente suponen que la función h que se quiere optimizar es una función real

$$h: \mathbb{R}^n \rightarrow \mathbb{R}$$

Como se explicó en la sección 2.3 (Modelos aleatorios) la simulación de tráfico está compuesta de gran cantidad de elementos aleatorios. Esto ocasiona que nuestra función a optimizar pierda su propiedad de ser una función real convirtiéndose en función aleatoria. Esto es, dado un elemento z el valor de $h(z)$ es el resultado de un experimento de cierta variable aleatoria H cuyo rango es \mathbb{R} . Dicho de otra manera, el valor $h(z)$ será un número real diferente para cada evaluación de la función h en el mismo elemento z .

Es necesario entonces encontrar un método que asocie la función aleatoria con una función real para poder así aplicar el método de optimización descrito. Como parte del alcance de este trabajo se explorarán y compararán los siguientes dos métodos:

4.7.1 MÉTODO 1: ESCENARIO FIJO

El primer método propuesto en este trabajo es pre-generar y guardar todos los elementos aleatorios de una corrida del escenario al cual llamaremos *escenario aleatorio semilla e*. Definimos entonces la función analítica:

$$h_e: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$h_e(z) = w$$

como la ejecución de la función h utilizando los parámetros de semáforo z pero tomando los elementos aleatorios pre-generados del escenario e . De esta manera la evaluación de la función garantiza generar el mismo resultado w en cada evaluación del vector z .

4.7.2 MÉTODO 2: ESTIMADOR DE LA MEDIA

Consideremos la variable aleatoria X resultado del experimento de correr la simulación aleatoria durante el tiempo de simulación y calcular el tiempo promedio de cada vehículo en el escenario. Podemos entonces redefinir nuestra función h como la función real

$$h: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$h(z) = \mu$$

$$z = (z_1, z_2, z_3, \dots, z_n), \quad n = 3m$$

$$(z_i, z_{i+1}, z_{i+2}) = (s, t_v, t_r), i \in \{1, 4, 7, \dots, n-2\}$$

Donde μ es la media de la función de distribución de X .

Dada la cantidad de elementos aleatorios en la simulación es imposible calcular de manera analítica μ . Es necesario recurrir entonces a un estimador de la media. Sean x_1, x_2, \dots, x_n experimentos de la variable aleatoria X , Calculamos la media muestral como:

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^n x_i$$

Y redefinimos entonces la función h como:

$$h: \mathbb{R}^n \rightarrow \mathbb{R}$$

$$h(z) = \hat{\mu}$$

A mayor tamaño de muestra n , mejor será la aproximación de la función h a el valor real de la media de la función de distribución.

4.7.3 VENTAJAS Y DESVENTAJAS ENTRE LOS MÉTODOS

Tenemos entonces dos métodos para convertir nuestra función estadística en una función real. El método 1, escenario fijo, presenta la ventaja de producir una función real y no una función aproximada. En cuanto al tiempo de cálculo presenta una ventaja considerable dado que solo se requiere una evaluación de la función comparado con las muchas evaluaciones del segundo método necesarias para obtener una buena aproximación de h .

El método 2, estimación de la media, presenta la ventaja de una mayor aleatoriedad al considerar múltiples experimentos de cada escenario. Adicionalmente el resultado obtenido de la función objetivo es por mucho una mejor representación de la realidad dado que modela de una mejor manera la variabilidad de día a día en el mismo escenario. La calificación obtenida por este método es un mejor reflejo de ambiente que se intenta modelar a comparación del método 1 que considera solo un evento aislado. Los resultados de este trabajo demostrarán cuál de los dos métodos es más adecuado para la modelación de escenarios viales.

En este capítulo se presentó a detalle el método de optimización seleccionado para encontrar los escenarios viales que minimizan el tráfico vehicular. Se presentó la definición matemática del problema a optimizar, se describió el algoritmo genético utilizado y sus ventajas sobre los algoritmos de descenso y se propusieron dos soluciones al problema de la variabilidad estadística. En el siguiente capítulo se presentará el software final desarrollado que implementa en su totalidad la construcción, modelación y optimización de escenarios viales.

5 APLICACIÓN: LA GLORIETA SANTA FE

En este capítulo se describe paso a paso cómo llevar a cabo la definición, construcción, configuración y optimización de un escenario real utilizando los elementos de software implementados en este trabajo.

5.1 LA GLORIETA SANTA FE



Figura 17: La Glorieta Santa Fe

El escenario real que se ha seleccionado como el alcance de este trabajo es la Glorieta Santa Fe en la ciudad de Guanajuato, Gto. La razón de esta selección es que éste escenario presenta el mayor problema de tráfico vehicular en la ciudad donde se localiza el Centro de Investigación en Matemáticas A.C. institución en la que se elaboró este trabajo de tesis. El objetivo de escoger este escenario es un intento de beneficiar a la comunidad alrededor del Centro de Investigación con el trabajo científico que se desarrolla en el mismo.

La Glorieta Santa Fe es el escenario que recibe todo el flujo vehicular de entrada a la ciudad proveniente de las ciudades de Silao y León. Es decir, regula todo el flujo vehicular de

entrada y salida de la ciudad en esta dirección. Además de la carga de estas dos ciudades, todas las mañanas recibe a una cantidad considerable de vehículos de la población trabajadora que se dirige de los suburbios hacia el centro de la ciudad.

Esta glorieta cuenta con 4 avenidas entrantes, 4 avenidas salientes, 2 retornos 1 avenida lateral, 8 semáforos vehiculares y 2 semáforos peatonales para regular el flujo vehicular dentro de la glorieta. La siguiente imagen muestra una vista satelital de la glorieta obtenida de *Google Maps*



Figura 18: Imagen satelital de la Glorieta Santa Fe

Para llevar a cabo la tarea de optimización de tráfico vehicular descrita en el capítulo anterior, fue necesario desarrollar una colección de seis diferentes paquetes de software. A lo largo del capítulo se irán describiendo uno a uno los paquetes de software necesarios para llevar a cabo la definición, construcción, configuración, optimización y visualización de un escenario de principio a fin. En este caso, la Glorieta Santa Fe.

5.2 OBTENCIÓN DE LA IMAGEN SATELITAL

Accediendo a *Google Maps* se puede obtener la imagen satelital de casi cualquier escenario vial en el mundo. Sin embargo, es necesario obtener una imagen de la mayor calidad posible para poder hacer acercamientos y mejorar la experiencia de usuario.

Google Maps provee imágenes de mediana calidad por lo que fue necesario construir una imagen final combinando tomas con el mayor zoom disponible de diferentes secciones del escenario. Para nuestro caso se obtuvieron 9 imágenes de la aplicación de *Google Maps*. Una vez obtenidas las imágenes se procedió a combinarlas utilizando el software de edición de imágenes *GIMP*. La imagen anterior es el resultado final de este esfuerzo.

Es necesario aplicar un segundo proceso para eliminar los automóviles de las calles. Esto para utilizar las capacidades del software visual desarrollado de superposición y animación de vehículos sobre la imagen obtenida. Este proceso fue también logrado utilizando *GIMP*. La imagen final limpia se muestra en la siguiente figura.



Figura 19: Imagen procesada de la glorieta

Finalmente, para lograr una mayor precisión al especificar la correspondencia entre píxeles de la imagen y coordenadas en el plano real se agrandó la imagen en un factor de 4x con el algoritmo de interpolación cúbica de *GIMP*. El resultado final es una imagen de 11360×8360 píxeles.

5.3 DIVISIÓN DEL ESCENARIO EN SUS ELEMENTOS BÁSICOS

En la sección 2.5 se definieron los tres elementos básicos: La calle la curva y el cruce. Se procede entonces dividir la Glorieta Santa Fe en sus elementos simples. Este es un

trabajo minucioso para el cual se recomienda utilizar algún software de edición de imágenes para marcar visualmente los elementos y así facilitar el trabajo. A continuación, se muestra el resultado final de dividir el escenario en sus 49 elementos básicos:



Figura 20: División de la glorieta en sus 49 elementos básicos

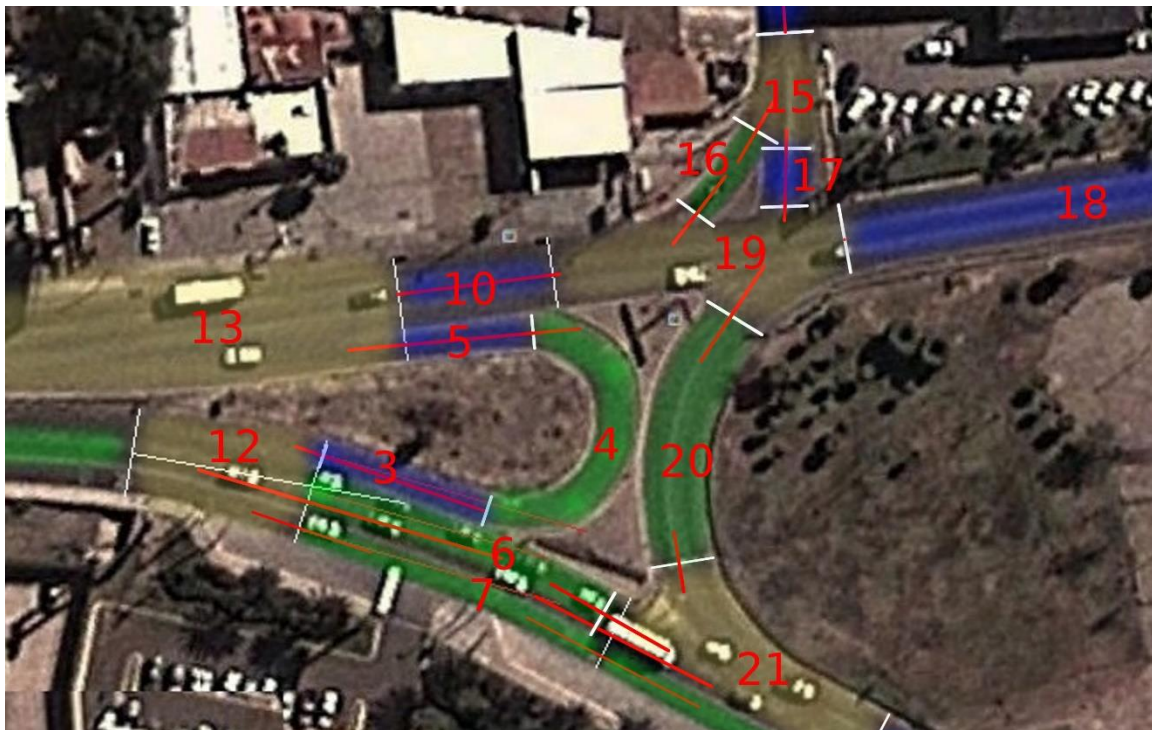


Figura 21: Elementos básicos (acercamiento 1)



Figura 22: Elementos básicos (acercamiento 2)

Como se puede observar en las tres figuras anteriores, los elementos tipo calle están marcados en amarillo, las curvas en verde y los cruceros en morado. Es importante numerar los elementos para la definición de conexiones y cruceros que se describirá más adelante. Otro detalle que se debe resaltar son los segmentos de recta de color rojo que se observan en la imagen. Estos segmentos fueron trazados pasando exactamente por punto medio de la calle o curva y en la dirección del movimiento vehicular. Estas marcas son necesarias para la definición de cada calle y curva, como se explica en las secciones siguientes.

5.4 DEFINICIÓN DE ELEMENTOS BÁSICOS EN EL SOFTWARE

Para ayudar al usuario a la definición espacial de cada uno de los elementos básicos, se construyó un software visual que toma como dato de entrada la imagen de elementos básicos de la sección anterior. El software ayuda al usuario a capturar con exactitud las coordenadas espaciales del inicio y fin de cada elemento calle o curva, así como los vectores dirección de entrada y salida. A continuación, se muestra una imagen del primer paquete de software. El *Extractor De Coordenadas*.

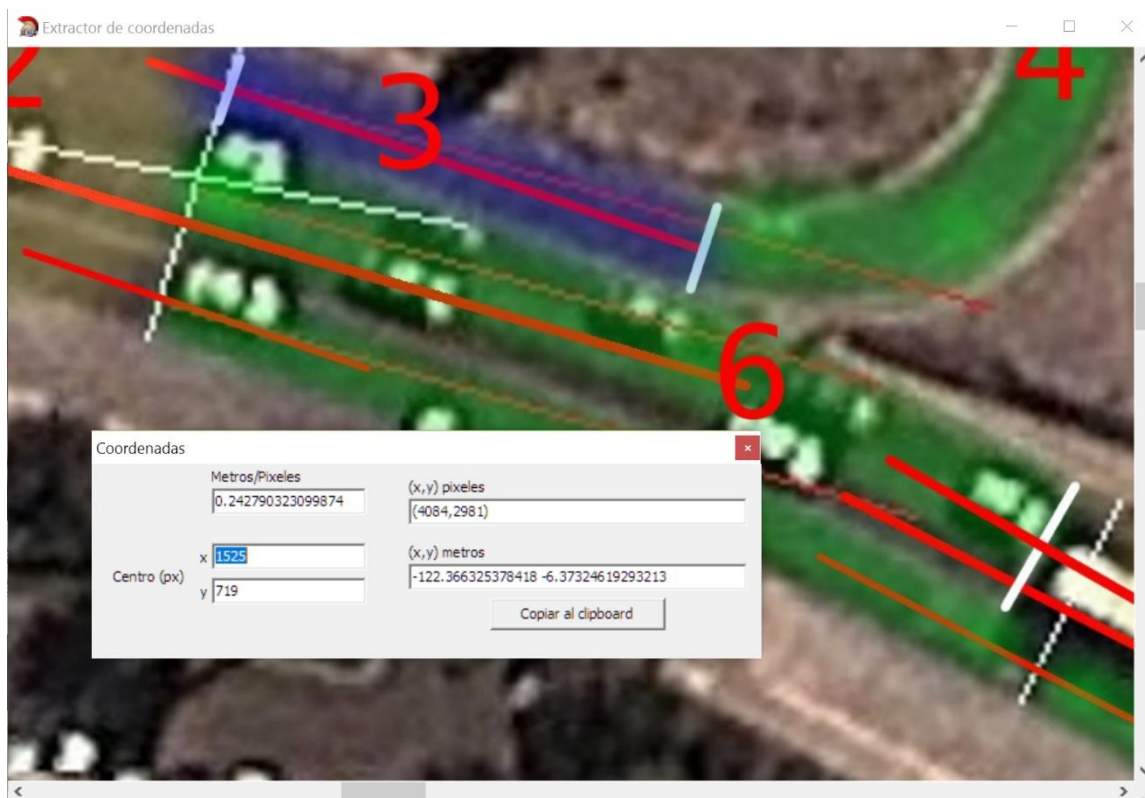


Figura 23: Software *Extractor De Coordenadas*

Es necesario definir una correspondencia entre los pixeles de la imagen y el mundo real. El software define esta correspondencia mediante la selección de un pixel que se empatará con el origen del plano real. Además de este pixel origen se debe especificar la correspondencia entre metros y pixeles.

Para nuestro escenario seleccionamos el centro de la Glorieta como el pixel origen. Para definir la correspondencia metros-pixeles se tomó una medición en la aplicación *Google Maps* como se muestra a continuación.

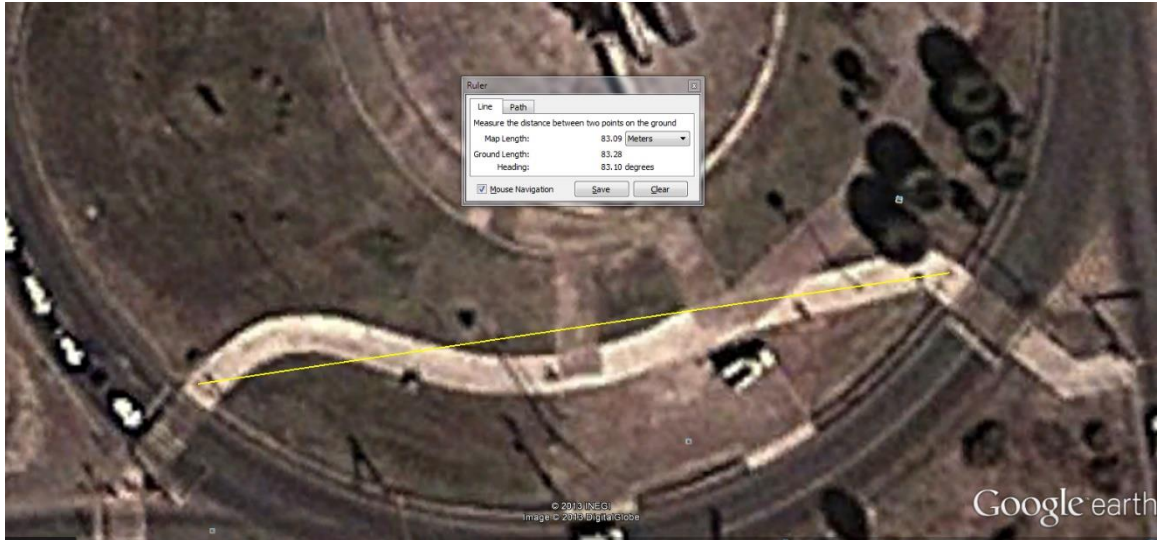


Figura 24: Medición del escenario usando *Google Maps*

Se procedió entonces a identificar los mismos puntos de medición en nuestra imagen. Dichos puntos corresponden a los pixeles (1554,1650) y (1218,1715). Calculando la distancia entre los dos puntos (342.2295 pixeles) y dividiendo entre los 83.09 metros reportados por *Google Maps*, se obtiene la equivalencia de 0.242790323 metros por pixel. Una vez configurado el *Extractor De Coordenadas* con este valor, se procede a extraer las coordenadas reales de cada uno de los pixeles de inicio y fin de cada calle o curva en la imagen.

5.4.1 DEFINICIÓN DE CALLES

Como se especificó en la sección 2.5.1, Para la definición de una calle son necesarios seis parámetros:

$$Ca = \left((x_{ini}, y_{ini}), (x_{fin}, y_{fin}), N, a \right)$$

Los primeros cuatro parámetros, el punto inicial y final de cada calle, se obtienen con el *Extractor De Coordenadas*. Dando clic en el punto medio del inicio de la calle el software proporciona los puntos equivalentes en el plano \mathbb{R}^2 . El número de carriles N se obtiene del conocimiento del escenario, y a , el ancho de cada carril, se puede obtener mediante una medición en campo o utilizando el software para obtener dos puntos que representen el ancho del carril y calcular su distancia euclidiana.

Es importante definir un ancho de carril igual para todas las calles y curvas en el escenario, o al menos para las calles y curvas interconectadas entre cruceo y cruceo. Esto para

asegurar un movimiento suave de los vehículos por el escenario. Para todo nuestro escenario se utilizó un ancho de carril constante de 3 metros.

Consideremos la sección del escenario observada en la siguiente figura:



Figura 25: Sección del escenario

Obsérvese que en esta sección los elementos 1, 3, 5, 10, 11 y 17 fueron definidos de tipo calle (color morado). Haciendo uso del *Extractor De Coordenadas* se obtuvieron los valores de la siguiente tabla para cada uno de estos elementos.

Elemento	x_{ini}	y_{ini}	x_{fin}	y_{fin}	N
1	-369.345	-8.012	-237.206	14.264	3
3	-161.031	15.235	-138.451	7.7086	1
5	-132.078	32.048	-149.680	30.470	1
10	-129.407	40.121	-150.530	37.390	3
11	-209.528	29.742	-369.830	5.2807	3
17	-97.238	57.541	-97.420	49.590	2

5.4.2 DEFINICIÓN DE CURVAS

Para la definición de una curva es necesario especificar 11 parámetros como se muestra en la definición dada en la sección 2.5.2.

$$Cu = \left((x_{ini}, y_{ini}), (x_{fin}, y_{fin}), (x'_{ini}, y'_{ini}), (x'_{fin}, y'_{fin}), N, a, l \right)$$

Los puntos comunes a las calles se definen de la misma manera que en la sección anterior. Estos son el punto inicial, punto final, número de carriles y ancho de carril.

Para los vectores dirección de entrada y salida $(x'_{ini}, y'_{ini}), (x'_{fin}, y'_{fin})$ el software permite especificar este vector como dos puntos en el plano. Para el vector de entrada (x'_{ini}, y'_{ini}) uno de esos puntos es el punto (x_{ini}, y_{ini}) . Y de la misma manera (x_{fin}, y_{fin}) define uno

de estos puntos para el vector de salida. Solo será necesario especificar un punto adicional para el vector de entrada y uno para el vector de salida. Como se mencionó anteriormente, en la imagen de división de elementos se marcaron segmentos de recta de color rojo que definen este vector dirección. Estos dos puntos pueden entonces obtenerse haciendo uso del *Extractor De Coordenadas*.

Es importante mencionar que, si dos elementos curva están conectados, por ejemplo, los elementos 7 y 8 en nuestro escenario, el vector dirección de salida del elemento 7 debe ser igual al vector dirección de entrada del elemento 8. De otro modo el movimiento de los vehículos al cambiar de un elemento a otro no será suave. Para asegurar esta correspondencia, los segmentos de recta de color rojo fueron trazados como una sola recta. De este modo, el mismo segmento, al ser utilizado para definir los datos del elemento 7 y el elemento 8 generara el mismo vector dirección en ambos elementos. A continuación, se presentan las definiciones de los elementos curva 7 y 8 en la imagen.

Elemento	$x_{1_{ini}}$	$y_{1_{ini}}$	$x_{2_{ini}}$	$x_{2_{ini}}$
7	-170.629	7.466	-164.369	5.341
8	-132.928	-6.920	-122.670	-12.261

Elemento	$x_{1_{fin}}$	$y_{1_{fin}}$	$x_{2_{fin}}$	$x_{2_{fin}}$
7	-122.670	-12.322	-109.316	-19.241
8	-15.478	-71.138	-7.769	-75.386

Elemento	N	l
7	1	-0.75
8	1	-0.75

El parámetro de curvatura l fue encontrado de manera experimental para cada una de las curvas en el escenario utilizando el software de simulación *SimTraf* el cual se presentara más adelante. En este software se fueron probando diferentes parámetros l hasta encontrar los parámetros adecuados que replicaran visualmente las curvaturas reales. Se muestra a continuación el pintado de los elementos curva 4 y 20, así como sus parámetros de curvatura configurados en el software.



Figura 26: Curvas y calles sobrepuestas en la imagen satelital

Elemento	N	L
4	1	0.75
20	2	-0.25

5.4.3 CONSTRUCCIÓN DE CRUCEROS

Los cruces no tienen localización espacial. Como se explicó en la sección 2.5.3, un cruce es un conjunto de conexiones del tipo:

$$C_{e_1, e_2} = (e_1, n_1, e_2, n_2)$$

Es decir, un conjunto de conexiones entre carriles de diferentes elementos calle o curva.

En la Figura 25 se pueden observar los elementos crucero 12,13,15,19 y 21 en amarillo. Consideremos el elemento 12. Este elemento reparte el flujo de los tres carriles de la curva 2 a los elementos 3 6 y 7. Físicamente esta sección del escenario reparte el carril 2.1 (Elemento 2, carril 1 visto de derecha a izquierda) al carril 7.1. El carril 2.2 es mapeado al 6.1 y finalmente el flujo del carril 2.3 se reparte al 6.2 y al 3.1. Definimos entonces al elemento 12 como $\{(2,1,7,1), (2,2,6,1), (2,3,6,2), (2,3,3,1)\}$

5.4.4 ARCHIVO ESCENARIO ESC

Una vez obtenidos todos los datos espaciales de cada uno de los elementos en el escenario se procede a construir un archivo *escenario.esc* (escenario) el cual será uno de los archivos de entrada para todos los elementos de software de este trabajo. La definición del archivo *esc* queda como sigue:

```
nElementos
  //Calle 1
nElemento tipo //1 1 (tipo: 1=calle 2=curva 3=crucero)
xIni yIni xFin yFin //Todas coordenadas y medidas en metros
nCarriles anchoCarril
  //Calle 2
nElemento //2 1
xIni yIni xFin yFin
nCarriles anchoCarril
  //Curva 1
nElemento //3 2
x0 y0 x1 y1 x2 y2 x3 y3
nCarriles anchoCarril parametroL
  //Curva 2
nElemento //4 2
x0 y0 x1 y1 x2 y2 x3 y3
nCarriles anchoCarril parametroL
  //Crucero1
nElemento //5 3
nCallesEntrantes

  //Conexiones elementos Flujo Calles y curvas
  // (-1=sin conexión)
  //Se deben especificar todos los elementos y en orden
nConexiones
  //Calles o curvas
elemento elementoAnterior elementoSiguiente
  //Crucero
  //primer calle entrante
elemento nCallesEntrantes
calleEntrante1
calleEntrante2
...
```

Este archivo representa entonces la definición espacial y conectiva de todos los elementos del escenario. Notemos que este archivo no contiene toda la información del crucero, es decir, la correspondencia entre carriles de entrada y carriles de salida en los elementos del crucero. En este archivo solo se especifican los elementos entrantes y salientes del crucero

(mediante el valor elementoAnterior en las calles y curvas). En la siguiente sección se hará uso de la información restante del cruce.

5.5 DEFINICIÓN DE FLUJOS EN EL ESCENARIO

Una vez capturada la información constructiva de la sección anterior, se procede a capturar la información del flujo vehicular en el escenario. Esta información deberá ser definida dentro de un archivo de texto *escenario.flu* (**flujo**) para su utilización posterior. A continuación, se detalla el contenido del archivo.

```
cuadrosPorSegundo tiempoTotal(minutos)
    //Calles generadoras
nElementosGeneradores
nElementosSalientes
    //flujos
    //elemento1
nElemento
nIntervalos
    //intervalo1
duración(min) vehiculosPorMinuto
    //intervalo2
duración(min) vehiculosPorMinuto

    //probabilidades
elementoEntrante1 p1 p2 p3 ....
elementoEntrante1 p1 p2 p3 ....
...
elementoEntranteN p1 p2 p3 ....

    //carriles de rutas
nElementoEntrante nElementoSaliente nRutas
    //ruta 1
nElementosEnRuta
carril1 carril2 carril3 ...
    //ruta 2
nElementosEnRuta
carril1 carril2 carril3 ...
```

El primer dato en el archivo son los cuadros por segundo a los cuales se realizará la simulación. En nuestro caso definimos *cuadrosPorSegundo* = 30. Se requiere definir también la duración total de la animación. Los resultados de este trabajo se basan en una duración total de 20 minutos (*tiempoTotal*). Experimentalmente se concluyó que 20 minutos son suficientes para cargar el escenario de tráfico vehicular y observar el desempeño de los semáforos. Sin embargo, el software permite experimentar con una duración diferente especificándolo en este archivo.

5.5.1 ELEMENTOS GENERADORES

El siguiente paso es definir los elementos por donde los vehículos ingresan al escenario (elementos generadores) y los elementos donde los vehículos terminar el recorrido y abandonan el escenario (elementos salientes).

Para cada elemento generador. El software permite definir comportamientos diferentes para tiempos diferentes. Es decir, en un elemento generador podrán insertarse 10 vehículos por minuto la primera hora de animación y luego cambiar a 30 vehículos por minuto para las siguientes dos horas. Esta variabilidad fue implementada en el software para modelar el comportamiento durante periodos prolongados de tiempo donde el flujo vehicular cambia. El software permite un número ilimitado de intervalos de inserción, así como intervalos y frecuencias de inserción independientes en cada elemento generador.

En el caso de la Glorieta Santa Fe, se tienen 4 elementos generadores. Los elementos 1, 14, 32 y 45 (Ver Figura 20). En los experimentos iniciales se intentó modelar el tráfico en la hora pico con una duración total de 3 horas de animación. Se definieron entonces para cada elemento generador 3 intervalos de inserción, uno para cada hora, donde se pretendía modelar un tráfico inicial ligero, después el tráfico pesado de la hora pico y al final un tráfico moderado.

Los resultados obtenidos mostraron que para tráficos ligeros casi cualquier configuración de semáforos obtiene un buen resultado. Es cuando el flujo vehicular aumenta la diferencia entre dos configuraciones de semáforos es evidente. Otro fenómeno observado es que para tráfico pesado se requieren solo unos pocos minutos para saturar el escenario y comenzar a observar la eficiencia o ineficiencia de los semáforos. Es decir, una vez que se empieza a generar una cola de vehículos en un semáforo, no importa si la cola es larga o corta, los semáforos reparten de igual manera el tráfico independientemente de los vehículos en espera.

Dadas las observaciones anteriores, y tomando en cuenta que a menor tiempo de animación menor es el procesamiento computacional y menor el tiempo de espera necesario para obtener resultados, se reajustaron los tiempos e intervalos de modelación. Para cada uno de los 4 elementos generadores se modelaron 20 minutos de animación con un solo intervalo de flujo. Los vehículos por minuto para cada elemento generador se muestran en la siguiente tabla.

Elemento	Vehículos por minuto
1	20
14	10
32	25
45	35

La configuración anterior intenta reflejar el comportamiento de la Glorieta Santa Fe en la hora pico de la mañana. Donde el tráfico vehicular proviene principalmente de la autopista (Elemento 32) y Tránsito del Estado (Elemento 45). El tráfico moderado proveniente del centro de la ciudad (Elemento 1) y el tráfico ligero de la Presa de la Olla (Elemento 14).

Es importante recalcar que para obtener un modelado preciso del tráfico en la glorieta es necesario tomar mediciones en campo de los vehículos por minuto en cada elemento generador. Para la optimización vehicular se observó mediante múltiples experimentos que para altos valores de vehículos por minuto (similares a los observados en la glorieta) las calles de entrada a la glorieta se llenan y comienzan a acumularse un gran número de vehículos en espera. Este fenómeno refleja adecuadamente lo que sucede en el escenario real. Sin embargo, al obtener la calificación del escenario (tiempo promedio de viaje de todos los vehículos involucrados en el escenario) los vehículos en la cola acaparaban la calificación produciendo márgenes de variación muy reducidos entre una configuración de semáforos y otra.

Esta variación impacta de manera negativa el algoritmo optimización cuando se toma la función objetivo como la media muestral y requiere un aumento considerable del tamaño de muestra para compensar el impacto, generando mayores tiempos de procesamiento.

La recomendación del autor de este trabajo es buscar valores de vehículos por minuto que no generen demasiados vehículos en cola, pero conservando la proporción de inserción en los elementos generadores del escenario. Por ejemplo, considérense los valores en la siguiente tabla.

Elemento	Vehículos por minuto
1	40
14	20
32	50
45	70

Estos valores conservan la proporción de inserción entre los 4 elementos. Es decir, en ambas tablas el elemento 14 inserta la mitad de vehículos que las 32. Sin embargo, la segunda tabla genera una cantidad considerable de vehículos en cola. Se recomienda entonces realizar la optimización con los datos de la primera tabla ya que se facilita la convergencia del algoritmo y los resultados finales demostraron ser similares.

5.5.2 DEFINICIÓN DE RUTAS

Los siguientes datos a definir son las rutas que los vehículos pueden tomar para transitar por el escenario. Primeramente, para cada elemento generador se debe conocer el número total de rutas diferentes que comienzan en dicho elemento. Por ejemplo. Para vehículos que entran al escenario por el elemento 1, provenientes del centro de la ciudad, existen 5

posibles rutas. Estas son, tomar el retorno antes de entrar a la glorieta, tomar el acceso sin semáforo hacia la caseta, tomar la glorieta y salir hacia la presa, tomar la glorieta y salir hacia tránsito del estado, y tomar la glorieta y luego el retorno hacia el centro comercial Roció.

Una vez conocidas todas las posibles rutas se procede a definir la probabilidad de que un vehículo tome cada una de las rutas que comienzan con el elemento entrante. Para el elemento 1 las probabilidades se dan en la siguiente tabla.

Elemento salida	Probabilidad
11	0.05
31	0.4
39	0.14
43	0.4
44	0.01

Para cada elemento entrante, se deben especificar entonces todas las posibles rutas a tomar y sus probabilidades. Las probabilidades utilizadas en esta simulación fueron el resultado de la intuición y experiencia el autor de este trabajo. Es claro que, para una toma de decisiones sobre el escenario real, las probabilidades tienen que ser medidas cuidadosamente en campo.

Después de describir las probabilidades de cada elemento y cada ruta en el archivo *flu*, se procede a definir el flujo de cada ruta a través de los elementos básicos. Es necesario especificar entonces la secuencia de elementos a seguir en cada ruta, así como el carril por el que se debe transitar en cada elemento. En caso de poder transitar por varios carriles es necesario alimentar el software con rutas diferentes, una para cada carril, dado que la modelación de cambio de carril arbitrario no fue implementada en este trabajo.

El flujo de elementos y los carriles correspondientes para la primera ruta se describen en la siguiente tabla, siendo el primer elemento en la tabla el elemento inicial y último elemento el elemento final de la ruta. Se debe notar que los elementos del tipo cruce carecen de carriles por lo que se especifica un 0 como su valor de carril.

Elemento	Carril
1	3
2	3
12	0
3	1
4	1
5	1
13	0
11	3

El archivo *flu* queda completado habiendo definido todas las rutas posibles en el escenario con sus carriles correspondientes. Este archivo contiene entonces toda la información correspondiente al flujo vehicular y es un dato de entrada necesario para todos los paquetes de software de este trabajo.

5.6 DEFINICIÓN DE SEMÁFOROS EN EL ESCENARIO

Teniendo definido el escenario (archivo *esc*) y los flujos dentro del escenario (archivo *flu*) se procede a la definición de semáforos en el escenario. Como se describió en la sección 2.5.6 un semáforo se define como:

$$S = (t_v, t_r, r, s) \in \mathbb{R}^4$$

Para definir entonces los semáforos en los escenarios se requiere especificar las rectas o curvas en las cuales el semáforo se encuentra físicamente, así como sus tiempos rojo verde, retraso y shift (desfase). Estos datos deberán ser introducidos en un archivo *sem* (**semáforos**) con el siguiente formato:

```
nSemaforos
nElemento1 tVerde tDelay tCicloCompleto tShift (todo en segundos)
nElemento2 tVerde tDelay tCicloCompleto tShift (todo en segundos)
...
```

Como parte de la experimentación de esta tesis, se tomaron mediciones en campo de los tiempos de semáforo en la glorieta Santa Fe. El archivo *sem* de esta configuración, *escenarioGlorieta.sem*, se muestra a continuación.

```
8
6 59 0 112 0
20 53 0 112 59
24 59 0 112 8
34 53 0 112 67
36 48 0 92 68
49 44 0 92 24
18 76 0 92 0
17 16 0 92 76
```

Notamos entonces que la Glorieta Santa Fe cuenta con 8 semáforos para regular el tráfico vehicular. Los semáforos están acomodados físicamente en pares. Esto es, los semáforos de los elementos 6 y 20 están coordinados de manera que en cualquier momento solo uno de ellos deja pasar vehículos. De la misma manera se encuentran coordinados los semáforos en los elementos 24 y 34, 36 y 49, y finalmente 18 y 17. Dada esta paridad se cumple entonces que *tCicloCompleto* siempre es la suma de los tiempos *tVerde* de ambos semáforos pares.

Notemos que el archivo *sem* no especifica los tiempos de semáforo en rojo. Esto debido a que tal valor puede obtenerse para cada semáforo mediante la resta del *tCicloCompleto* y *tVerde*. Nuestro escenario vehicular ha sido entonces completamente definido mediante los archivos *esc*, *flu* y *sem*. Toda la información para realizar la simulación vehicular se encuentra contenida en estos tres archivos.

5.7 GENERACIÓN DE AUTOMÓVILES EN EL ESCENARIO

En la sección 2.3.2 se describió cómo se lleva a cabo la generación de vehículos en el escenario. El software fue diseñado para generar los vehículos de manera aleatoria para cada simulación. Sin embargo, para implementar la función objetivo con escenario fijo es necesario pre-generar los vehículos a insertar. Esto es, aleatoriamente simular la inserción de vehículos, guardando cada vehículo con sus datos característicos en un archivo *escs* (escenario semilla) para su uso posterior. Los datos característicos que deberán ser guardados son su tipo, instante de inserción, velocidad, aceleración, y ruta a transitar.

Para lograr la generación del archivo *escs* se desarrolló el software *Generador De Tráfico* el cual recibe como parámetros de entrada el *escenario.esc*, *escenario.flu* y una semilla. Esto es, el número entero con el que se alimentara la función *srand()* de c++. *Generador De Tráfico* produce entonces un archivo de salida *escenario.escs*.

El archivo *escs* generado podrá ser utilizado para correr simulaciones garantizando que los vehículos insertados siempre serán los mismos. Esto permitirá comparar con precisión dos diferentes configuraciones de semáforos.

5.8 SIMULACIÓN DEL ESCENARIO

Una vez definidos los tres archivos del escenario (*esc*, *flu* y *sem*) y teniendo el archivo auxiliar de inserción de vehículos *escs*, estamos listos para realizar una simulación de tráfico en el escenario.

5.8.1 SOFTWARE REPORTADOR DE TRÁFICO

El elemento de software *Reportador De Tráfico* fue generado para este propósito. Este software recibe como parámetros de entrada los cuatro archivos (*esc*, *flu*, *sem* y *escs*) y procede a realizar una simulación cuadro por cuadro hasta completar el tiempo total de simulación.

El software *Reportador De Tráfico* realiza entonces las siguientes tareas:

1. Construye virtualmente el escenario utilizando las coordenadas de calles y curvas del archivo *esc*.
2. Conecta todos los elementos y carga las rutas con sus probabilidades mediante la especificación del archivo *flu*.

3. Cuadro por cuadro inserta vehículos en el escenario mediante la especificación del archivo *esc*.
4. Cuadro por cuadro el software calcula las distancias de cada vehículo a otros vehículos y semáforos en la cercanía. Con esta información se recalculan las velocidades de cada vehículo y su nueva posición dentro de los elementos del escenario.
5. El software extrae los vehículos que van terminando el recorrido y actualiza las estadísticas correspondientes.
6. Se reportan resultados parciales del estado del escenario y una vez terminada la simulación se da un reporte general y la calificación del escenario.

A continuación, se muestran los resultados arrojados por *Reportador De Tráfico* utilizando *escenarioGlorieta.sem* como parámetro de entrada.

Cálculo a 30 frames por segundo)

Minutos totales de animación=20

Tiempo (minutos)= 1, Vehículos insertados= 91:

Vehículos en escenario= 78, Vehículos terminados= 13, Vehículos en cola= 0

Tiempo (minutos)= 2, Vehículos insertados= 181:

Vehículos en escenario= 115, Vehículos terminados= 66, Vehículos en cola= 0

Tiempo (minutos)= 3, Vehículos insertados= 271:

Vehículos en escenario= 116, Vehículos terminados= 155, Vehículos en cola= 0

Tiempo (minutos)= 4, Vehículos insertados= 362:

Vehículos en escenario= 150, Vehículos terminados= 212, Vehículos en cola= 0

Tiempo (minutos)= 5, Vehículos insertados= 452:

Vehículos en escenario= 180, Vehículos terminados= 272, Vehículos en cola= 0

Tiempo (minutos)= 6, Vehículos insertados= 542:

Vehículos en escenario= 180, Vehículos terminados= 362, Vehículos en cola= 0

Tiempo (minutos)= 7, Vehículos insertados= 633:

Vehículos en escenario= 202, Vehículos terminados= 431, Vehículos en cola= 0

Tiempo (minutos)= 8, Vehículos insertados= 723:

Vehículos en escenario= 190, Vehículos terminados= 533, Vehículos en cola= 0

Tiempo (minutos)= 9, Vehículos insertados= 813:

Vehículos en escenario= 215, Vehículos terminados= 598, Vehículos en cola= 0

Tiempo (minutos)= 10, Vehículos insertados= 903:

Vehículos en escenario= 219, vehículos terminados= 684, vehículos en cola= 0

Tiempo (minutos)= 11, vehículos insertados= 994:
Vehículos en escenario= 214, vehículos terminados= 774, vehículos en cola= 6

Tiempo (minutos)= 12, vehículos insertados= 1084:
Vehículos en escenario= 204, vehículos terminados= 878, vehículos en cola= 2

Tiempo (minutos)= 13, vehículos insertados= 1174:
Vehículos en escenario= 216, vehículos terminados= 945, vehículos en cola= 13

Tiempo (minutos)= 14, vehículos insertados= 1265:
Vehículos en escenario= 230, vehículos terminados= 1023, vehículos en cola= 12

Tiempo (minutos)= 15, vehículos insertados= 1355:
Vehículos en escenario= 225, vehículos terminados= 1110, vehículos en cola= 20

Tiempo (minutos)= 16, vehículos insertados= 1445:
Vehículos en escenario= 247, vehículos terminados= 1168, vehículos en cola= 30

Tiempo (minutos)= 17, vehículos insertados= 1535:
Vehículos en escenario= 227, vehículos terminados= 1268, vehículos en cola= 40

Tiempo (minutos)= 18, vehículos insertados= 1626:
Vehículos en escenario= 211, vehículos terminados= 1360, vehículos en cola= 55

Tiempo (minutos)= 19, vehículos insertados= 1716:
Vehículos en escenario= 213, vehículos terminados= 1441, vehículos en cola= 62

Tiempo (minutos)= 20, vehículos insertados= 1806:
Vehículos en escenario= 204, vehículos terminados= 1526, vehículos en cola= 76

***Tiempo promedio global: 2.16715
Minutos totales (vehículos terminados): 3280.87
Minutos totales (vehículos restantes): 493.081
Minutos totales (vehículos en cola): 139.92
Minutos totales: 3913.87
Número de vehículos (terminados): 1526
Número de vehículos (restantes): 204
Número de vehículos (en cola): 76
Número de vehículos: 1806

Tiempos promedio (minutos) de vehículos que terminaron la ruta
Tiempos por ruta:
0.583965 0.865741 1.60902 1.42597 1.442
2.0912 3.30046 4.17167 3.73959 3.39389

4.97013 3.77206 3.97419 3.87461 3.55556
 1.21276 2.11196 0.912553 2.83456 x

Tiempo promedio global (Vehículos que terminaron la ruta):
 2.14998

Número de vehículos que terminaron la ruta

Por ruta:
 22 147 57 149 5
 38 50 8 56 1
 82 7 123 118 4
 192 230 232 5 x

Número global de vehículos que terminaron la ruta: 1526
 Vehículos totales insertados: 1806
 Vehículos restantes en el escenario: 280

Distancias de rutas (metros):
 517.684 747.114 710.012 783.333 781.312
 390.932 669.014 628.609 701.931 698.921
 853.231 688.478 602.375 672.925 674.932
 694.799 960.686 442.026 993.602 x

Velocidad promedio (km/h) de vehículos que terminaron la ruta

Velocidad por ruta:
 55.0552 53.3679 29.2733 20.2361 35.0146
 18.4661 7.78206 10.2041 7.9325 12.3561
 3.99918 12.4875 5.84478 5.10287 17.979
 10.2259 14.8659 33.1865 16.8112 x

Velocidad promedio global: 19.9989

Distribución de vehículos que terminaron la ruta

Por ruta:
 0.0144168 0.0963303 0.0373526 0.0976409 0.00327654
 0.0249017 0.0327654 0.00524246 0.0366972 0.000655308
 0.0537353 0.00458716 0.0806029 0.0773263 0.00262123
 0.125819 0.150721 0.152031 0.00327654 x

Distribución de vehículos que terminaron la ruta

Por elemento origen:
 0.0578947 0.386842 0.15 0.392105 0.0131579
 0.248366 0.326797 0.0522876 0.366013 0.00653595
 0.245509 0.0209581 0.368263 0.353293 0.011976
 0.291351 0.349014 0.352049 0.00758725 x

Número de vehículos que terminaron la ruta

Por elemento origen:
 380
 153
 334
 659

***Tiempo promedio global: 2.16715
 Análisis terminado
 Tiempo total de cálculo(min) 0.05

Se pueden hacer varias observaciones del reporte anterior:

1. Los vehículos en cola (vehículos que están esperando a ingresar en el escenario debido a que los elementos de entrada están llenos) se mantienen bajos con un máximo de 76 vehículos al final de la simulación. Esto muestra que los datos de vehículos por minuto en el archivo *flu* son adecuados para la optimización que se describe en la siguiente sección.
2. Toma solamente 5 minutos para llenar el escenario de vehículos. A partir del minuto 5, la cantidad de vehículos en el escenario se mantiene constante alrededor de 200 vehículos. Una animación de 20 minutos será entonces suficiente para capturar la eficiencia del escenario.
3. La calificación de este escenario, es de 2.16715. Esto es, para la configuración de semáforos *escenarioGlorieta.sem*, el tiempo que le lleva a un vehículo cruzar la Glorieta Santa Fe es en promedio 2 minutos 10 segundos.
4. La gran mayoría de los minutos totales acumulados corresponden a los vehículos que terminaron el recorrido (3280.87, comparado con los 493.081 minutos de los vehículos aun en el escenario y los 139.92 minutos de los vehículos en cola). Esto nos muestra que la medida de tiempo promedio de viaje está siendo tomada principalmente de los vehículos que terminaron, pero incluyendo una penalización de 1 segundo por los vehículos aun en el escenario y en cola (comparando el tiempo promedio de 2.14998 de solo los vehículos que terminaron el recorrido).
5. El archivo *escs* inserta un total de 1806 vehículos en 20 minutos de animación. O visto de otra manera. El flujo al que la glorieta es sometida en esta simulación es de 5418 vehículos por hora (1806 x 3).
6. La Glorieta Santa Fe con la configuración de semáforos descrita en *escenarioGlorieta.sem* tiene una capacidad de 4578 vehículos por hora (1526 vehículos terminados en 20 minutos x 3). Esto es, en una hora permite a 4578 vehículos provenientes de todas sus entradas realizar su recorrido a través de la glorieta. Este valor será muy útil al comparar la eficiencia lograda por el algoritmo de optimización.

El software reporta información detallada por ruta, como distancia, tiempo promedio y cantidad de vehículos. Si bien estos datos no son explotados en este trabajo, se decidió incluirlo en el reporte ya que provee una herramienta para comparar que las mediciones de flujos en campo correspondan a las que el software genera.

El software también provee en el reporte un detalle de velocidades de vehículos globales y por ruta. La velocidad máxima configurada para cada elemento es de $60 \frac{km}{h}$ y algunos vehículos al simular un comportamiento agresivo pueden presentar velocidades de hasta $80 \frac{km}{h}$. Sin embargo, el reporte muestra que la velocidad promedio es de solamente $20 \frac{km}{h}$.

Incluso en la ruta con mayor velocidad, la velocidad promedio máxima es de $55\frac{km}{h}$. Se decidió incluir esta información dado que puede ser muy útil al definir límites de velocidad en el escenario o incluso para tomar decisiones sobre configuraciones de semáforo que minimicen o maximicen las velocidades promedio.

Además del reporte generado, *Reportador De Tráfico* provee dos archivos de salida. Un archivo *ani* (**animación**) y un archivo *inf* (**información**). El archivo *ani* contiene todas las posiciones de todos los vehículos para cada cuadro. El archivo *inf* contiene los siguientes datos:

```
30      //Cuadros por segundo
36000  //Cuadros totales
```

Estos archivos permiten al software *SimTraf* que se describe más adelante, generar una animación visual del movimiento de cada vehículo cuadro por cuadro.

Al considerar la cantidad de información presentada por el software y el detalle de la misma se puede percibir el valor que este trabajo proporciona. Muchas de estas mediciones son imposibles de conseguir en campo. Esta información permitirá un análisis mucho más detallado de las condiciones viales y una toma de decisiones informada.

5.8.2 SOFTWARE ANALIZADOR DE TRÁFICO

Para tener un análisis integral de un escenario, es necesario medir y observar su comportamiento no solo bajo condiciones vehiculares particulares (archivo *escs*) sino también bajo condiciones aleatorias. Después de todo, los vehículos que visitarán el escenario un día no serán los mismos vehículos al día siguiente ni llegarán en el mismo orden. Es necesario entonces realizar un análisis aleatorio bajo múltiples situaciones vehiculares. El software *Analizador De Tráfico* fue implementado para llevar a cabo esa tarea.

Este software recibe como parámetros de entrada los archivos *esc*, *flu* y *sem*. A diferencia de *Reportador De Tráfico* que recibe el escenario de condiciones vehiculares, *Analizador De Tráfico* generará aleatoriamente las condiciones vehiculares sobre la marcha. El software en sustitución del archivo *escs* recibe un número de repeticiones. Esto es, *Analizador De Tráfico* realizara la simulación ejecutada por *Reportador De Tráfico* tantas veces como se le indique, generando un reporte estadístico como salida.

A continuación, se muestra el resultado de una corrida de *Analizador De Tráfico* para la configuración de semáforos en *escenarioGlorieta.sem* con 1000 repeticiones.

```
Iniciando análisis de escenario escenarioGlorieta.sem con 1000
repeticiones
1.93913
1.92586
1.98871
```

```

1.91518
1.99141
1.95252
1.97333
2.0036
2.03246
2.02735
1.98839
2.00213
2.01698
2.01539
...
2.05202
2.10686
2.01313
1.98707
2.02757
2.35971
2.04478
1.97809
2.23699
2.0454
2.08079
2.08511
2.03731
2.18784
6.84918
1.99341
1.93929
2.00697
2.29515
1.89178
2.22866
2.16438
2.15205
2.1597
2.69615
2.0552
2.30305
2.14883
2.15978
***Tiempo promedio global: 2.12546
***Tiempo mínimo global: 1.74423
***Tiempo máximo global: 6.84918
Varianza: 0.0877578
Desviación estándar: 0.296239

```

Del reporte generado podemos realizar las siguientes observaciones:

1. Cada uno de los renglones representa la calificación de una de las repeticiones. Es decir, cada entrada en el archivo de salida representa la calificación que *Reportador De Tráfico* generaría para un escenario *escs*.
2. Para ciertas condiciones vehiculares, *escenarioGlorieta.sem* es capaz de lograr tiempos promedio de viaje de 1 minuto 45 segundos (1.74423) Sin embargo, el tiempo promedio de viaje puede llegar hasta los 6 minutos 51 segundos en un “mal

día”. Esto no es un problema en el simulador sino al contrario, demuestra la fidelidad del simulador al replicar los fenómenos observados en el día a día del escenario real.

3. Notemos que, a pesar de estar utilizando las mismas frecuencias de inserción de vehículos, se obtienen diferencias de tiempo enormes. Esto es debido a la aleatoriedad del problema. Por ejemplo, es posible que en una corrida ingresen una mayor cantidad de vehículos con preferencia de manejo conservadora, ocasionando así un cambio en el flujo de todo el escenario.

A pesar de las variaciones en la calificación de escenarios, es posible tener una medida estable: la media muestral (2.12546 en la corrida anterior). La siguiente tabla muestra los resultados de cinco diferentes corridas de *Analizador De Tráfico* con 1000 repeticiones cada una. Notar cómo la media muestra (Tiempo promedio global) tiene muy poca variación.

Media	Varianza	Mínimo	Máximo
2.12546	0.08776	1.74423	6.84918
2.13203	0.0936	1.76612	6.14631
2.11795	0.07731	1.76874	6.24568
2.12871	0.09805	1.77488	6.67247
2.11179	0.074	1.74639	6.48189

Como podemos observar, a pesar de la gran variación entre escenarios mínimo y máximo la calificación de la media muestral permanece casi constante. Para fines de comparación estableceremos que la calificación final de la configuración de semáforos en *escenarioGlorieta.sem* es de 2.123188 (El promedio de las 5 calificaciones). Esto es 2 minutos 7 segundos.

Analizador De Tráfico será entonces nuestra herramienta para realizar comparaciones finales entre los resultados arrojados por la optimización y el escenario actual ya que este software proporciona esta medida considerando una gran variedad de condiciones vehiculares.

5.9 OPTIMIZACIÓN DEL ESCENARIO

5.9.1 SOFTWARE OPTIMIZADOR DE TRÁFICO

Una vez obtenido un punto de referencia (*escenarioGlorieta.sem*) y su calificación de eficiencia (2.123188), procedemos a utilizar el algoritmo genético para tratar de encontrar una configuración de semáforos que mejore el flujo vehicular. Para esta tarea se desarrolló el software *Optimizador De Tráfico* que se describe a continuación.

5.9.2 REDUCCIÓN DE DIMENSIONALIDAD

El escenario que se quiere optimizar cuenta con 8 semáforos. Por lo que el individuo descrito en el algoritmo de la sección 4.6 es un vector de tamaño 32 (4 variables por semáforo). La búsqueda del mejor individuo se deberá entonces realizar en el espacio \mathbb{R}^{32} .

Como se mencionó anteriormente, el escenario en cuestión tiene una propiedad especial, se tienen cuatro grupos de semáforos binarios. Esto es, mientras un semáforo del grupo esta prendido, el otro se encuentra apagado. Podemos aprovechar esta propiedad de la siguiente manera:

Sea un grupo de semáforos binarios (S_1, S_2) con valores:

$$\left((t_{v_1}, t_{r_1}, r_1, s_1), (t_{v_2}, t_{r_2}, r_2, s_2) \right) = (t_{v_1}, t_{r_1}, r_1, s_1, t_{v_2}, t_{r_2}, r_2, s_2)$$

Siendo semáforos binarios se cumple que:

$$t_{r_2} = t_{v_1}$$

$$t_{r_1} = t_{v_2}$$

$$s_2 = t_{v_1} + s_1$$

Entonces el vector $(t_{v_1}, t_{r_1}, r_1, s_1, t_{v_2}, t_{r_2}, r_2, s_2)$ puede ser construido a partir del vector $(t_{v_1}, t_{v_2}, s_1, r_1, r_2)$. Si además consideramos que el valor r es un valor constante para todos los semáforos, podemos reducir el semáforo binario al vector (t_{v_1}, t_{v_2}, s_1) .

Podemos entonces caracterizar cada grupo de semáforos binarios con solamente tres elementos. Esto reduce el espacio de búsqueda de \mathbb{R}^{32} a \mathbb{R}^{12} . Si escribimos el parámetro s de cada semáforo como el desfase respecto al primer semáforo, entonces el parámetro s del primer semáforo binario siempre es 0 y el espacio de búsqueda reduce a \mathbb{R}^{11} . Podemos entonces implementar los individuos del algoritmo genético como el siguiente vector:

$$(t_{v_1}, t_{v_2}, t_{v_3}, t_{v_4}, t_{v_5}, t_{v_6}, t_{v_7}, t_{v_8}, s_3, s_5, s_7) \in \mathbb{R}^{11}$$

5.9.3 REDUCCIÓN DEL ESPACIO DE BÚSQUEDA

Notemos que el vector de individuos contiene ocho valores de duración del semáforo en verde y tres valores de desfase. El espacio de búsqueda se puede reducir como se explicó en la sección 4.6.1 a:

$$t_v \in [t_{v_{min}}, t_{v_{max}}]$$

$$s \in [0, 2t_{v_{max}}]$$

Se tiene entonces que para definir el espacio de búsqueda se requiere especificar solamente dos valores: $t_{v_{min}}$ y $t_{v_{max}}$

5.9.4 DEFINICIÓN DE ARCHIVO OPT

El software *Optimizador De Tráfico* recibe como parámetros de entrada los archivos *esc* y *flu*. El archivo *sem* no será necesario dado que el software proporcionará el archivo *sem* que minimiza el tiempo promedio de viaje. En su lugar será necesario alimentar al software con un archivo *opt* (**optimización**). La descripción del archivo *opt* es como sigue:

```
4 //nCruceros
0 //tDelay
20 200 // tverdeMin tverdeMax
0 400 //tDesfaseMin tDesfaseMax
8 //nSemaforos
2 6 20 // Crucero 1: nSemaforos nElemento1 nElemento2 ...
2 24 34 // Crucero 2: nSemaforos nElemento1 nElemento2 ...
2 36 49 // Crucero 3: nSemaforos nElemento1 nElemento2 ...
2 18 17 // Crucero 4: nSemaforos nElemento1 nElemento2 ...
```

El valor *nCruceros* representa el número de grupos de semáforos binarios en el escenario. *tDelay* representa el tiempo de retraso constante para todos los semáforos. En este trabajo se consideró un retraso de 0 segundos dado que es el tiempo de retraso que la glorieta tiene actualmente para todos sus semáforos.

El archivo requiere la definición de los tiempos mínimo y máximo de verde, así como los mínimos y máximos de desfase. Estos últimos son configurables en caso de que se requiera experimentar con desfases menores a $2t_{v_{max}}$, o mayores si las soluciones óptimas se encuentran cerca de la frontera y se requiera extender el límite para conservar la continuidad.

Finalmente, el archivo requiere la definición de los grupos de semáforos binarios. El software está preparado para escenarios más complejos donde los cruceros tienen más de dos semáforos coordinados.

5.9.5 OPERACIÓN DE OPTIMIZADOR DE TRÁFICO

Es necesario especificar parámetros adicionales intrínsecos al algoritmo genético:

- *Número de individuos*: Tamaño de la población.
- *Proporción de población mutada*: Proporción de la población a la que se le realizara mutación.
- *Proporción de entradas mutadas*: Para el vector de tamaño 11 se requiere la proporción de entradas a las que se le realizara la mutación.
- *Probabilidad de cruza*: La probabilidad de que a un individuo se le realice cruza. De lo contrario el individuo es clonado a la siguiente generación.

El software *Optimizador De Tráfico* realiza entonces las siguientes tareas:

1. Construye una población inicial de tamaño *Número de individuos*.
2. Califica cada elemento de la población, ya sea realizando una ejecución de *Reportador De Tráfico* con un mismo escenario semilla *escs* para todos los elementos (método de escenario fijo), o realizando una ejecución de *Analizador De Tráfico* con un número fijo de repeticiones para todos los elementos (método de estimador de la media).
3. Realiza las operaciones de selección, cruza y mutación descritas en la sección 4.6 generando una nueva población.
4. Repite los pasos 2-3 hasta el número total de iteraciones.
5. Presenta un reporte de resultados.

A continuación, se muestra el reporte para una corrida de ejemplo con *Número de individuos* = 4 y 2 iteraciones con $t_{v_{min}} = 10$, $t_{v_{max}} = 35$, $s_{min} = 0$ y $s_{max} = 2t_{v_{min}} = 70$ utilizando el método de escenario fijo.

Iniciando optimización con los siguientes parámetros:

```

escenario.esc: esc/escenario.esc
escenario.flu: esc/escenario.flu
escenario.opt: esc/escenario.opt
escenario.escs: esc/escenario.escs
logFile: tmp/corrida.log
nIndividuos: 4
nRepeticionesCalificacion: 1
proporcionPoblacionMutada: 1
proporcionEntradasMutadas: 0.090909
probCruza: 0.8
nIteraciones: 2
directorioSalida: tmp
-----Estado Inicial-----
calificaciones:
1.57851
2.40574
2.11993
2.73468
Resultados corrida (ordenados):
1.57851
2.11993
2.40574
2.73468
Calificación peor individuo: 2.73468
Calificación promedio: 2.20972
Calificación mejor individuo: 1.57851
Calificación mejor global: 1.57851
-----Iteración 1 de 2-----
calificaciones:
2.40574
1.49637
1.53046
2.11993
Resultados corrida (ordenados):
1.49637

```

```

1.53046
2.11993
2.40574
Calificación peor individuo: 2.40574
Calificación promedio: 1.88812
Calificación mejor individuo: 1.49637
Calificación mejor global: 1.49637
-----Iteración 2 de 2-----
calificaciones:
1.68063
1.8452
3.71619
1.53046
Resultados corrida (ordenados):
1.53046
1.68063
1.8452
3.71619
Calificación peor individuo: 3.71619
Calificación promedio: 2.19312
Calificación mejor individuo: 1.53046
Calificación mejor global: 1.49637
-----

```

Optimización terminada

Mejor calificación global:1.49637
Peor calificación global:3.71619

```

Mejores calificaciones en cada iteración:
1.57851
1.49637
1.53046
Calificaciones promedio en cada iteración:
2.20972
1.88812
2.19312
Peores calificaciones en cada iteración:
2.73468
2.40574
3.71619
Mejor calificación global en cada iteración:
1.57851
1.49637
1.49637
Optimización terminada
Tiempo total de cálculo(min) 0.216667

```

Optimizador De Tráfico permite ejecutar la optimización en dos modalidades. La primera es utilizando un mismo escenario *escs* para todos los individuos. Es decir, la función objetivo se obtiene mediante el cálculo realizado por *Reportador De Tráfico*. La corrida anterior se ejecutó bajo esta modalidad.

La segunda modalidad consiste en generar escenarios aleatorios para cada evaluación. Es decir, la función objetivo se obtiene mediante el cálculo realizado por *Analizador De Tráfico* con cierto número de repeticiones para cada calificación.

Optimizador De Tráfico reporta la calificación de cada individuo iteración a iteración generando como resultado dos archivos *sem*. El peor escenario y el escenario óptimo. A continuación, se muestra una comparativa entre el mejor escenario y el peor escenario.

Optimo, archivo <i>sem</i>	Peor, archivo <i>sem</i>
8	8
6 19.576281 0.000000 44.188293 0.000000	6 30.278709 0.000000 57.124077 0.000000
20 24.612015 0.000000	20 26.845367 0.000000
44.188293 19.576281	57.124077 30.278709
24 20.698814 0.000000	24 17.307896 0.000000
47.613491 39.196548	40.777039 28.463943
34 26.914677 0.000000	34 23.469143 0.000000
47.613491 59.895363	40.777039 45.771839
36 23.074818 0.000000	36 24.031046 0.000000
42.381325 29.939447	49.726562 44.338650
49 19.306507 0.000000	49 25.695517 0.000000
42.381325 53.014267	49.726562 68.369698
18 31.452770 0.000000	18 33.212944 0.000000
45.951012 6.759263	55.782135 24.822025
17 14.498243 0.000000	17 22.569189 0.000000
45.951012 38.212032	55.782135 58.034969

Optimo, Reportador De Tráfico	Peor, Reportador De Tráfico
***Tiempo promedio global: 1.49637	***Tiempo promedio global: 3.71619
Minutos totales (vehículos terminados): 2522.73	Minutos totales (vehículos terminados): 1237.48
Minutos totales (vehículos restantes): 176.577	Minutos totales (vehículos restantes): 3909.5
Minutos totales (vehículos en cola): 3.13	Minutos totales (vehículos en cola): 1564.46
Minutos totales: 2702.44	Minutos totales: 6711.44
Número de vehículos (terminados): 1636	Número de vehículos (terminados): 880
Número de vehículos (restantes): 157	Número de vehículos (restantes): 480
Número de vehículos (en cola): 13	Número de vehículos (en cola): 446
Número de vehículos: 1806	Número de vehículos: 1806

Optimo, Analizador De Tráfico 1000 repeticiones	Peor, Analizador De Tráfico 1000 repeticiones
***Tiempo promedio global: 1.58658	***Tiempo promedio global: 1.6636
***Tiempo mínimo global: 1.34054	***Tiempo mínimo global: 1.21473
***Tiempo máximo global: 4.21752	***Tiempo máximo global: 4.87294
Varianza: 0.0290869	Varianza: 0.243153
Desviación estándar: 0.170549	Desviación estándar: 0.493106

Se puede observar que escenarios con menor calificación encontrados con un escenario fijo (*escs*) tienen también una media menor. Esto es, el método de escenario fijo cumple con el objetivo de encontrar escenarios que son realmente mejores bajo un análisis estadístico. Sin embargo, la diferencia en calificación resulta ser menor una vez realizado un análisis estadístico completo.

5.9.6 BÚSQUEDA DE PARÁMETROS

Antes de ejecutar el algoritmo genético en forma se requiere configurar los parámetros de optimización. Dado que los parámetros adecuados para este problema en específico son desconocidos, es necesario realizar varios experimentos. Para estos experimentos usaremos el método de escenario fijo. Es decir, un mismo escenario *esc* para todos los individuos y todas las corridas. Esto debido a que al eliminar el factor variable de la inserción de automóviles se facilita el encontrar los parámetros de configuración del algoritmo genético.

5.9.6.1 TIEMPO DE SEMÁFORO

Para el primer conjunto de experimentos se fijaron los parámetros:

- *Proporción de población mutada* = 1
- *Proporción de entradas mutadas* = 0.09091
- *Probabilidad de cruza* = 0.8
- *Población* = 300
- *Iteraciones* = 500

Y se procedió a variar los tiempos de semáforo. A continuación, se muestran los resultados obtenidos:

Espacio de búsqueda			Resultados			
Verde min.	Verde máx.	Desfase máx.	Peor individuo	Mejor individuo	Calificación final	Desviación estándar
10	180	360	8.2465	2.0776	2.04229	0.125625
10	180	360	8.31054	2.23463	2.12814	0.098807
10	180	360	8.40697	2.16725	2.11538	0.155582
10	180	360	8.17676	2.16618	2.08925	0.0890504
10	180	360	8.22166	2.15158	2.1567	0.105546
10	45	90	6.85277	1.2734	1.49271	0.106401
30	240	480	8.23743	2.53399	2.49032	0.245628
15	45	90	6.68365	1.32588	1.55893	0.406555
15	45	90	6.27337	1.33317	1.56384	0.552379
10	35	70	6.54484	1.21041	1.37934	0.216138
10	35	70	6.5316	1.21402	1.38559	0.251017

Al considerar los resultados se observa lo siguiente:

1. El algoritmo genético encuentra mejores resultados si se reduce el campo de búsqueda.
2. Contrario a la intuición, tiempos cortos de duración de semáforos favorecen a encontrar una mejor solución.
3. Los mejores resultados contienen tiempos de semáforo menores a 35 segundos y desfases menores a 70 segundos

5.9.6.2 TAMAÑO DE POBLACIÓN Y NÚMERO DE ITERACIONES

Dados los resultados anteriores, se fijaron ahora los parámetros:

- *Proporción de población mutada* = 1
- *Proporción de entradas mutadas* = 0.09091
- *Probabilidad de cruza* = 0.8
- *Tiempo verde mínimo* = 10
- *Tiempo verde máximo* = 35
- *Desfase máximo* = 70

Y se procedió a variar el tamaño de población y el número de iteraciones. Los resultados se muestran a continuación:

Espacio de búsqueda		Resultados		
Población	Iteraciones	Mejor individuo	Calificación final	Desviación estándar
300	100	1.22413	1.47347	0.36673
300	200	1.21909	1.44066	0.329552
300	300	1.2362	1.4506	0.327226
300	400	1.2097	1.47995	0.421478
300	500	1.21041	1.37934	0.216138
300	600	1.1915	1.4082	0.179779
300	700	1.21038	1.47277	0.427932
100	500	1.22491	1.45537	0.250726
200	500	1.20348	1.45	0.103677
400	500	1.21923	1.42832	0.183241
500	500	1.19985	1.42804	0.233983
600	500	1.21486	1.39304	0.29096
700	500	1.20747	1.47452	0.386036

Al considerar los resultados se observa lo siguiente:

1. Tamaños de población muy grandes no favorecen al algoritmo. Los mínimos fueron encontrados con poblaciones pequeñas de entre 200 y 300 individuos

2. Un resultado mínimo en el algoritmo con punto fijo no garantiza un mínimo en la evaluación estadística.

5.9.6.3 PARÁMETROS DE CRUZA Y MUTACIÓN

Dados los resultados anteriores, se fijan ahora los parámetros:

- *Tiempo verde mínimo* = 10
- *Tiempo verde máximo* = 35
- *Desfase máximo* = 70
- *Población* = 200
- *Iteraciones* = 600

Y se procedió a variar los parámetros de cruce y mutación. Los resultados se muestran a continuación:

Espacio de búsqueda			Resultados		
Proporción población mutada	Proporción entradas mutadas	Probabilidad de cruce	Mejor individuo	Calificación final	Desviación estándar
1	0.09091	0.8	1.21041	1.37934	0.21614
0.8	0.09091	0.8	1.18831	1.43064	0.12005
0.6	0.09091	0.8	1.21204	1.47933	0.29588
0.4	0.09091	0.8	1.20577	1.47466	0.44819
1	0.18182	0.8	1.20615	1.44854	0.34098
1	0.27273	0.8	1.20498	1.40058	0.27685
1	0.36364	0.8	1.21454	1.45174	0.3166
1	0.09091	1	1.22428	1.40329	0.09078
1	0.09091	0.6	1.22263	1.4559	0.32058
1	0.09091	0.4	1.19247	1.42205	0.21732
0.8	0.27273	0.4	1.22871	1.40296	0.36254

De todos los experimentos se concluye que los parámetros que encuentran configuraciones menores, y por lo tanto los parámetros más adecuados, son:

- *Tiempo verde mínimo* = 10
- *Tiempo verde máximo* = 35
- *Desfase máximo* = 70
- *Población* = 200
- *Iteraciones* = 600

- *Proporción de población mutada* = 0.8
- *Proporción de entradas mutadas* = 0.09091
- *Probabilidad de cruza* = 0.8

5.9.7 HARDWARE DISPONIBLE Y TIEMPOS DE PROCESAMIENTO

Como se comentó en la sección 4.7.2, el estimador de la media requiere de la definición de un número de repeticiones. Idealmente deberíamos utilizar las 1000 repeticiones que se han venido usando para comparar resultados finales, sin embargo, se tiene la restricción del tiempo de procesamiento requerido para computar la optimización que se explica a continuación.

5.9.7.1 HARDWARE DISPONIBLE

El Centro De Investigación en Matemáticas A.C. cuenta con equipo de hardware de última tecnología. Gracias al departamento de Ciencias de Computación del Centro, fue posible tener acceso a las 3 máquinas más potentes del Centro. A continuación, se muestran las características de hardware de cada una de ellas. Se agrega para fines de comparación una cuarta máquina, una laptop de gama media de última tecnología.

Máquina	Laptop	Teotl	Tlacotalpan	Atupal
Procesador	Intel® Core™ i7-6560U CPU @ 2.20 GHz	Intel® Xeon® CPU E5-4607 @ 2.20GHz	Intel® Xeon® CPU E5-4607 v2 @ 2.60GHz	Intel® Xeon® CPU E5-2699 v3 @ 2.30GHz
Núcleos	4	24	48	72
Memoria RAM	8 GB	503 GB	755 GB	755 GB

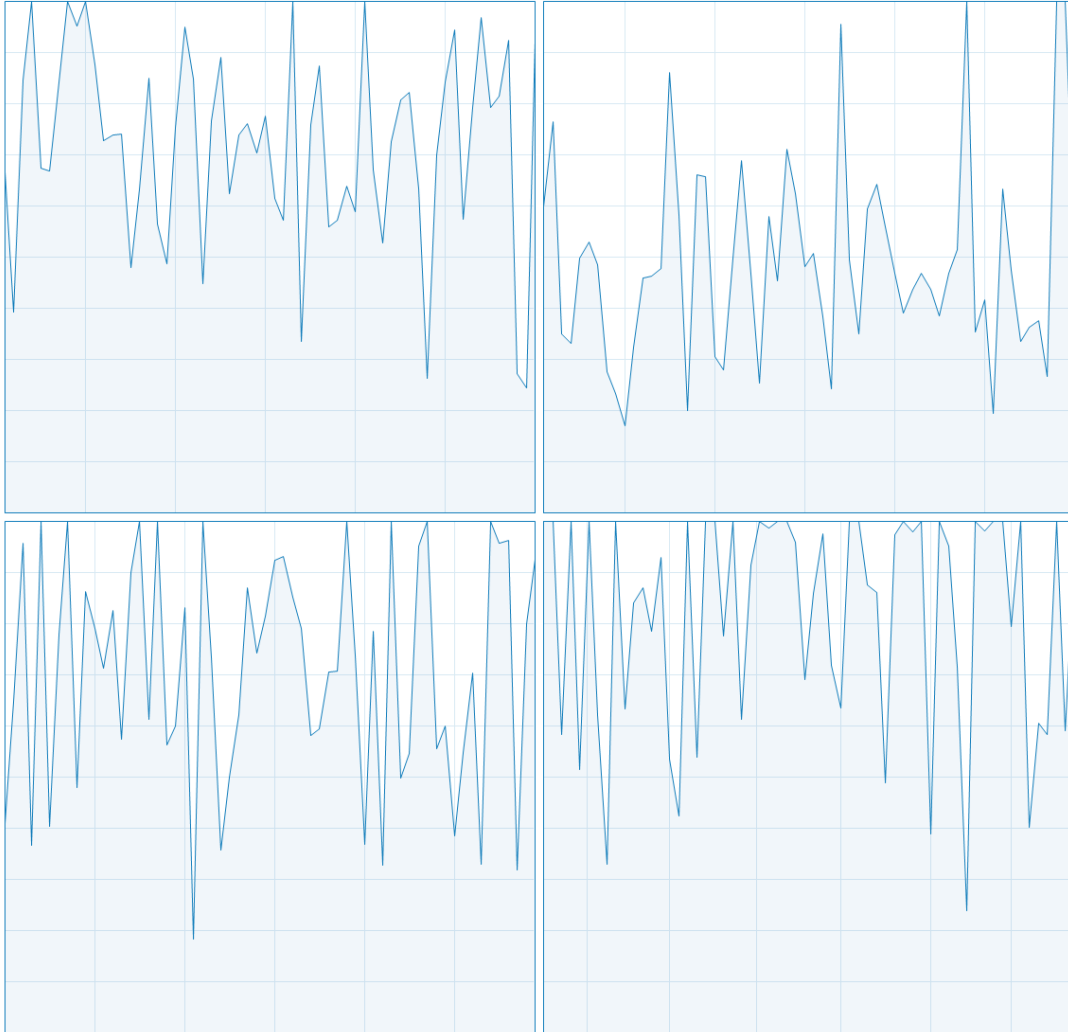
Dadas las capacidades multi-núcleo del hardware disponible, los softwares *Analizador de Tráfico* y *Optimizador de Tráfico* fueron programados con técnicas de paralización utilizando OpenMP. En las siguientes imágenes se muestra cómo las técnicas de paralización hacen uso eficiente de los recursos disponibles.

CPU

Intel(R) Core(TM) i7-6560U CPU @ 2.20GHz

% de uso durante 60 segundos

100 %



Uso	Velocidad		Velocidad máxima:	2.21 GHz
83%	2.72 GHz		Sockets:	1
Procesos	Subprocesos	Identificadores	Núcleos:	2
151	2072	92998	Procesadores lógicos:	4
Tiempo activo			Virtualización:	Habilitado
2:17:44:42			Caché L1:	128 kB
			Caché L2:	512 kB
			Caché L3:	4.0 MB

Figura 27: Carga de trabajo en Laptop al correr *Analizador De Tráfico*

```

ivan@atupal: ~
1  [||||| 100.0%] 19 [||||| 98.7%] 37 [||||| 100.0%] 55 [||||| 100.0%]
2  [||||| 100.0%] 20 [||||| 98.0%] 38 [||||| 100.0%] 56 [||||| 100.0%]
3  [||||| 100.0%] 21 [||||| 94.2%] 39 [||||| 100.0%] 57 [||||| 100.0%]
4  [||||| 100.0%] 22 [||||| 100.0%] 40 [||||| 100.0%] 58 [||||| 97.4%]
5  [||||| 100.0%] 23 [||||| 100.0%] 41 [||||| 100.0%] 59 [||||| 92.2%]
6  [||||| 100.0%] 24 [||||| 97.4%] 42 [||||| 100.0%] 60 [||||| 99.4%]
7  [||||| 100.0%] 25 [||||| 82.5%] 43 [||||| 99.3%] 61 [||||| 85.1%]
8  [||||| 100.0%] 26 [||||| 99.4%] 44 [||||| 100.0%] 62 [||||| 100.0%]
9  [||||| 100.0%] 27 [||||| 94.8%] 45 [||||| 100.0%] 63 [||||| 80.5%]
10 [||||| 100.0%] 28 [||||| 90.3%] 46 [||||| 100.0%] 64 [||||| 100.0%]
11 [||||| 100.0%] 29 [||||| 99.4%] 47 [||||| 100.0%] 65 [||||| 100.0%]
12 [||||| 100.0%] 30 [||||| 95.5%] 48 [||||| 100.0%] 66 [||||| 100.0%]
13 [||||| 100.0%] 31 [||||| 97.4%] 49 [||||| 100.0%] 67 [||||| 100.0%]
14 [||||| 100.0%] 32 [||||| 99.4%] 50 [||||| 100.0%] 68 [||||| 99.4%]
15 [||||| 99.4%] 33 [||||| 90.8%] 51 [||||| 100.0%] 69 [||||| 96.7%]
16 [||||| 100.0%] 34 [||||| 100.0%] 52 [||||| 100.0%] 70 [||||| 88.3%]
17 [||||| 100.0%] 35 [||||| 100.0%] 53 [||||| 100.0%] 71 [||||| 95.4%]
18 [||||| 100.0%] 36 [||||| 100.0%] 54 [||||| 99.4%] 72 [||||| 100.0%]
Mem | 34284/773930MB Tasks: 33, 74 thr; 73 running
Swp | 0/786306MB      Load average: 72.02 68.56 66.37
                                Uptime: 38 days, 03:15:55

  PID USER   PRI  NI  VIRT  RES  SHR  S  CPU% MEM%  TIME+  Command
36658 ivan    20   0 33.2G 31.2G 3028 R 98.56 4.1 719h ./traffic-optimizer esc/escenari
36675 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h04:23 ./traffic-optimizer esc/escenari
36709 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 9h53:25 ./traffic-optimizer esc/escenari
36727 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h02:48 ./traffic-optimizer esc/escenari
36714 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 9h56:28 ./traffic-optimizer esc/escenari
36718 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h02:31 ./traffic-optimizer esc/escenari
36715 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h00:54 ./traffic-optimizer esc/escenari
36690 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h01:54 ./traffic-optimizer esc/escenari
36703 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h01:33 ./traffic-optimizer esc/escenari
36722 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h03:03 ./traffic-optimizer esc/escenari
36696 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h04:05 ./traffic-optimizer esc/escenari
36677 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 9h55:55 ./traffic-optimizer esc/escenari
36694 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 9h57:58 ./traffic-optimizer esc/escenari
36707 ivan    20   0 33.2G 31.2G 3028 R 98.8 4.1 10h04:37 ./traffic-optimizer esc/escenari
36659 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 9h56:16 ./traffic-optimizer esc/escenari
36672 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 10h05:05 ./traffic-optimizer esc/escenari
36671 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 9h58:54 ./traffic-optimizer esc/escenari
36708 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 9h51:29 ./traffic-optimizer esc/escenari
36682 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 9h56:30 ./traffic-optimizer esc/escenari
36664 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 10h02:04 ./traffic-optimizer esc/escenari
36680 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 10h00:34 ./traffic-optimizer esc/escenari
36692 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 10h04:42 ./traffic-optimizer esc/escenari
36724 ivan    20   0 33.2G 31.2G 3028 R 98.1 4.1 9h59:58 ./traffic-optimizer esc/escenari

```

Figura 28: Carga de trabajo en Atupal al correr *Optimizador De Tráfico*

5.9.7.2 TIEMPOS DE PROCESAMIENTO

A continuación, se muestra una comparativa de los tiempos de espera necesarios para ejecutar una corrida de *Analizador de Tráfico* con 1000 iteraciones, una corrida de *Optimizador de Tráfico* con 500 individuos y 1500 iteraciones bajo el método de escenario fijo, y la proyección de tiempo de la misma corrida utilizando el método de estimador de media con 1000 repeticiones:

	Laptop	Teotl	Tlacotalpan	Atupal
<i>Analizador Tráfico</i>	12.82 min	1.47 min	1.28 min	0.58 min
<i>Optimizador de Tráfico (escenario fijo)</i>	11231.7 min (203.9 horas = 8.5 días) *estimado	1414.23 min (23.6 horas)	1176.58 min (19.6 horas)	570.33 min (9.5 horas)
<i>Optimizador de Tráfico (Estimador de media)</i>	23.3 años *estimado	2.7 años *estimado	2.2 años *estimado	1 año *estimado

Como podemos observar, el hardware disponible juega un papel muy importante en los resultados de este trabajo. Gracias a las tres máquinas disponibles y a la velocidad de procesamiento de cada una, ha sido posible obtener los resultados de esta tesis. Por poner una comparativa, para obtener los resultados hasta este punto en el documento fueron requeridos 108.8 días de tiempo máquina. La misma cantidad de experimentos hubieran tomado 4 años de uso continuo en la Laptop antes mencionada.

5.9.8 OPTIMIZACIÓN MEDIANTE ESTIMADOR DE LA MEDIA

Dadas las proyecciones de tiempo de la sección anterior para la ejecución de *Optimizador de Tráfico* con el método de media muestral con 1000 repeticiones, concluimos que no es viable la ejecución del algoritmo con dichos parámetros. Es necesario entonces ejecutar los experimentos con números de repeticiones apropiados que arrojen estimaciones de la media razonables en tiempos sensatos.

Todos los experimentos de *Optimizador de Tráfico* hasta el momento han sido realizados bajo el esquema de escenario fijo. Notemos que para los parámetros óptimos encontrados (*Población* = 200, *Iteraciones* = 600) Se requieren de 120'000 evaluaciones de escenario. Se realizaron experimentos para obtener el juego de parámetros adecuado (*Población*, *iteraciones* y *repeticiones*) que requiera el mismo número de iteraciones que el método de escenario fijo y que encuentre resultados óptimos. A continuación, se muestran los resultados:

Parámetros				Resultados		
Población	Iteraciones	Repeticiones	Total	Mejor individuo	Calificación final	Desviación estándar
200	60	10	120'000	1.31351	1.47224	0.416217
20	600	10	120'000	1.29426	1.41286	0.340332
100	24	50	120'000	1.346	1.37583	0.217424
50	48	50	120'000	1.39991	1.45835	0.339743
40	30	100	120'000	1.41126	1.43487	0.301099

Los parámetros que encuentran globalmente mejores individuos y que requieren un total de 120'000 repeticiones son:

- *Población* = 100
- *Iteraciones* = 24
- *Repeticiones* = 50

5.9.9 ESCENARIO FIJO VS ESTIMADOR DE LA MEDIA

Una vez encontrados parámetros óptimos para ambos métodos y que requieren el mismo número de cálculos de escenario, se realizaron 15 corridas de cada método para observar si alguno tiene una ventaja sobre el otro. A continuación, se muestran los resultados:

Escenario fijo	
mejor	calificación
1.19257	1.43143
1.21073	1.45311
1.19719	1.42239
1.18743	1.50073
1.18338	1.4575
1.21347	1.43675
1.22008	1.45464
1.21825	1.45906
1.20306	1.42562
1.2049	1.44492
1.1945	1.42822
1.20122	1.49968
1.20102	1.44525
1.19909	1.41745
1.19181	1.3737

Estimador de la media	
mejor	calificación
1.37355	1.48306
1.41599	1.42561
1.3786	1.3923
1.39871	1.40423
1.37417	1.42443
1.39881	1.43145
1.39465	1.46859
1.38667	1.40598
1.36279	1.41656
1.36405	1.43773
1.38231	1.42395
1.37348	1.43932
1.35557	1.39603
1.38366	1.45758
1.39253	1.42305

Al comparar los resultados de ambos métodos, se observó que el método de escenario fijo pudo encontrar mejores resultados. Sin embargo, las mejores calificaciones de ambos métodos son muy similares, tienen una diferencia de solamente 1.12 segundos. La recomendación del autor de este trabajo es utilizar el método de escenario fijo dado que la implementación es más sencilla.

5.10 VISUALIZACIÓN DE ESCENARIO

Los elementos de software presentados hasta este punto calculan analíticamente la eficiencia de un escenario y encuentran configuraciones de semáforo óptimas. Es necesario ahora poder presentar visualmente los resultados para corroborar su veracidad y observar el flujo de tráfico. Esto ayudará a tomar decisiones sobre cambios en el escenario real.

Para este propósito se desarrolló el software *SimTraf*. Este software presenta de manera visual la animación completa de un escenario dado. El software fue desarrollado específicamente para el sistema operativo Windows, haciendo uso del software CodeGear™ C++Builder® 2009, para el cual el autor de este trabajo de tesis cuenta con una Licencia Profesional.



Figura 29: Pantalla de Bienvenida del software *SimTraf*

5.10.1 ARCHIVOS NECESARIOS PARA LA ANIMACIÓN

Como datos de entrada *SimTraf* recibe seis archivos: *escenario.bmp* (imagen satelital del escenario), *escenario.esc*, *escenario.flu*, *escenario.sem*, *escenario.inf* y *escenario.ani*. Los dos últimos archivos son generados por *Reportador de Tráfico*, como se explicó en la sección 5.8.1.

5.10.2 OPERACIÓN DE SOFTWARE SIMTRAF

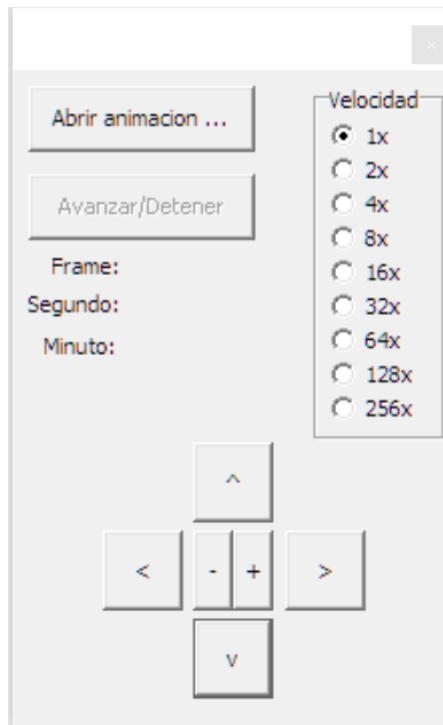


Figura 30: Pantalla de controles del software *SimTraf*

Después de iniciar *SimTraf*, se presenta la pantalla de controles. Se debe comenzar abriendo la animación mediante el botón “Abrir animación ...”. Esto cargará la imagen satelital, el escenario y los vehículos en pantalla y arrancará la animación comenzando en el cuadro 0.



Figura 31: Estado visual de *SimTraf* después de cargar la animación

SimTraf reporta entonces en la pantalla de controles el tiempo actual de la animación en cuadros (frames), segundos y minutos, como se muestra en la siguiente imagen.

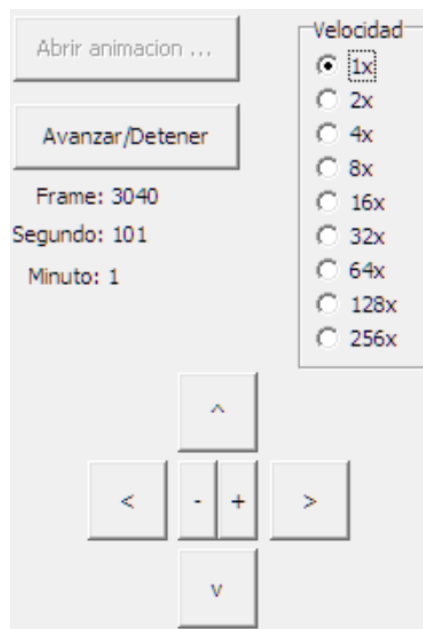


Figura 32: Información de progreso presentada por *SimTraf*

En cualquier momento se puede detener o reanudar la animación presionando el botón “Avanzar/Detener”. Así mismo, se puede cambiar la velocidad de animación en tiempo real simplemente seleccionando el valor deseado en el menú “Velocidad”.

5.10.3 ZOOM Y DESPLAZAMIENTO

El software permite navegar por el escenario libremente. Para acercar o alejar la vista se pueden utilizar los botones “+” y “-” en la pantalla de controles, utilizar el scroll del ratón, o bien, con gestos de acercamiento/alejamiento en pantallas táctiles. A continuación, se muestra la imagen de un acercamiento.



Figura 33: Acercamiento utilizando la característica de zoom del software

El software maneja pintado vectorial, por lo que el acercamiento no tiene límite. A continuación, se muestra una imagen de un acercamiento, donde se alcanzan a percibir los píxeles de la imagen. Es por esto que la obtención de una imagen de la mayor resolución posible es importante para mejorar la experiencia de usuario. Notar que debido al pintado vectorial los rectángulos no pierden resolución.



Figura 34: Capacidad de acercamiento del software

Al momento de abrir una animación, el software escala la imagen de tal manera que el escenario sea visible en su totalidad en pantalla completa. Al realizar acercamientos será necesario navegar a través del escenario utilizando las flechas en la pantalla de controles, utilizando el ratón, o bien, con el gesto de desplazamiento en pantallas táctiles.

En este capítulo se presentó a detalle cómo construir, analizar, optimizar y visualizar cualquier escenario vial en el mundo mediante el uso de los seis paquetes de software desarrollados en este trabajo. En el siguiente capítulo se presentan los resultados y conclusiones finales.

6 RESULTADOS Y CONCLUSIONES

En el capítulo anterior se mostraron los resultados analíticos obtenidos de la optimización de semáforos en la Glorieta Santa Fe. Tomando como base el escenario actual, se tomaron mediciones de la eficiencia de dicho escenario. El resultado fue que el tiempo total de viaje promedio de un vehículo que cruza la glorieta es de **2 minutos 7 segundos**.

6.1 RESULTADOS ANALÍTICOS

Después de realizar la optimización, el software pudo encontrar una configuración de semáforos con un tiempo total de viaje promedio de **1 minuto 22 segundos**. Esto es 45 segundos menos que el escenario de referencia. Cabe mencionar que esta medida es el promedio de ejecutar 1000 simulaciones aleatorias por lo que la configuración final de semáforos demostró tener un buen desempeño en múltiples situaciones viales.

El resultado obtenido genera entonces una mejora de **35.3%** en el tiempo de viaje promedio de los vehículos. La implementación de los resultados de este trabajo en campo tiene entonces el potencial de resolver los problemas planteados al inicio del documento: Reducir el estrés en los individuos y ahorrar tiempo y dinero a la población dado que el disminuir el tiempo de viaje de cada vehículo se reduce el gasto de gasolina.

Además de las ventajas de tiempo y estrés para cada conductor, se tienen ventajas sociales como la reducción de la contaminación ambiental y además el mejorar la utilidad de la infraestructura vial existente. Esto debido a que la configuración de semáforos óptima no solo reduce el tiempo de viaje de cada vehículo, sino que incrementa la capacidad del escenario vial de repartir vehículos. Como se mostró en la sección 5.8.1 La Glorita Santa Fe tiene la capacidad de repartir 4578 vehículos por hora utilizando la configuración de semáforos de referencia. Al modificar los semáforos de acuerdo al óptimo encontrado, la capacidad crece a 5088, es un incremento del **11.14%** en la capacidad del escenario.

6.2 RESULTADOS VISUALES

Haciendo uso del software *SimTraf*, podemos también comparar la eficiencia del escenario visualmente. Es decir, observar la situación vehicular en el escenario de referencia y el escenario óptimo. A continuación, se muestran imágenes del minuto 19 de animación en cada uno de los escenarios.



Figura 35: Situación vehicular en el minuto 19 en el escenario de referencia



Figura 36: Situación vehicular en el minuto 19 para el escenario óptimo

Como podemos observar, la diferencia es contundente. El escenario de referencia presenta largas colas de espera en las áreas con mayor tráfico mientras el escenario optimo deja fluir de una mejor manera a los vehículos eliminando las colas.

Estos resultados demuestran que las implementaciones descritas en este trabajo y los métodos de optimización implementados resolvieron satisfactoriamente la tarea de mejorar la situación de tráfico vehicular en la Glorieta Santa Fe.

6.3 VALIDACIÓN DE RESULTADOS

Es importante señalar que los resultados y las mejoras mostradas en esta sección fueron obtenidas con datos artificiales. Es decir, todos los datos de entrada del software, a excepción de los tiempos de semáforo de la glorieta Santa Fe, no fueron generados a partir de mediciones físicas en el escenario, sino solamente aproximados de acuerdo a la intuición del autor.

Si bien el tomar mediciones físicas queda fuera del alcance de este trabajo, se considera importante mencionar el esfuerzo adicional requerido para estar en la posición de tomar decisiones sobre un escenario físico.

El primer paso a realizar es el tomar mediciones de todos y cada uno de los datos de entrada en el software los cuales han sido ya detallados en la sección 3.1. Se requiere una medición estadística formal en campo para obtener los datos necesarios.

Una vez tomadas las mediciones, es necesario validar y ajustar el modelo. Para esto se requiere tomar muestras de tiempo total de viaje en todas las rutas en el escenario. Estas muestras deberán ser comparadas con los reportes generados por el software y hacer las adecuaciones correspondientes en las constantes del software (ver sección 7.2), y posiblemente en los modelos aleatorios utilizados. Este ejercicio deberá ser repetido para diferentes configuraciones de semáforos. De esta manera se garantiza que los resultados proporcionados por el software son equivalentes a los resultados obtenidos en campo.

Habiendo validado el software, se puede entonces proceder a calcular los tiempos de semáforo óptimos y hacer las modificaciones correspondientes en el escenario físico, mejorando así el flujo vehicular real en el escenario de estudio.

6.4 CONCLUSIÓN

El trabajo realizado generó las siguientes aportaciones:

- Se desarrolló un simulador de tráfico realista
- Se implementó software de construcción y análisis que replica escenarios reales

- Se implementó software de optimización que mejora las condiciones de tráfico en un escenario
- Se realizó un estudio de tráfico vehicular en un escenario real, la Glorieta Santa Fe.

Además de las aportaciones anteriores, se deja disponible para trabajo de investigación un laboratorio de tráfico vehicular en el que se pueden realizar diferentes trabajos derivados, por ejemplo:

- **Laboratorio constructivo:** Desarrollar una interfase amigable para el cliente final en la que se pueda realizar de manera automática modificaciones constructivas como agregar/quitar semáforos, modificar sentido de calles, agregar calles, ampliación y reducción de carriles, agregar/eliminar rutas, cambiar glorietas por cruceros, etc.
- **Laboratorio de comportamiento:** Experimentar con la implementación de diferentes medidas de comportamiento vehicular como: forzar comportamiento precavido/agresivo de conductores, implementar preferencias en cruceros, ceder el paso en cruceros, multas por detenerse en cruceros, modelación de accidentes, cierres de carril por reparaciones, etc. Evaluar el impacto del cambio de comportamiento en el aumento o disminución del tráfico vehicular.
- **Optimización de escenarios físicos:** realizar las mediciones y validación descritas en la sección anterior e implementar mejoras de tráfico en escenarios físicos.
- **Análisis de tráfico en diferentes escenarios:** Realizar el análisis de tráfico vehicular para toda la ciudad o para escenarios viales en otras ciudades.

Como conclusión final, solo queda decir que este trabajo es un ejemplo de los beneficios reales y tangibles que el uso del conocimiento científico puede lograr en la población. Es el deseo del autor que la ciencia en México continúe buscando maneras de beneficiar a la sociedad en que vivimos mediante trabajos aplicados enfocados al bienestar social.

7 APÉNDICE

7.1 SOFTWARE UTILIZADO

Para la realización de esta tesis se utilizaron los siguientes paquetes de software:

7.1.1 *GNU G++*

Todo el software analítico, Reportador De Tráfico, Analizador De Tráfico, y Optimizador De Tráfico, fue desarrollado y compilado con g++.

7.1.2 *OPENMP*

Toda la paralelización en el software Analizador de Tráfico y Optimizador de Tráfico fue lograda mediante el uso de la API para c++ de OpenMP

7.1.3 *CODEGEAR™ C++ BUILDER® 2009*

Todo el software visual, Extractor de Coordenadas y *SimTraf*, fue desarrollado con esta herramienta. El autor cuenta con la licencia de la versión Profesional de este software.

7.1.4 *GIMP*

Software de procesamiento y manipulación de imágenes utilizado para la edición de la imagen satelital y la imagen de bienvenida de *SimTraf*

7.1.5 *GOOGLE MAPS*

Aplicación Web utilizada para la obtención y medición de la imagen satelital de la Glorieta Santa Fe

7.1.6 *GIT*

Todo el control de versiones de este proyecto de tesis fue realizado mediante la implementación de un repositorio git.

7.2 CONSTANTES UTILIZADAS

A continuación, se presentan las constantes utilizadas en el software. Estas constantes están localizadas en el archivo Constantes.h y es utilizado por todos los elementos de software de este trabajo.

```

//Tsuru
//u: metros
const float ANCHO_CARRO_CH=1.64;
//u: metros
const float LARGO_CARRO_CH=4.33;
//u: kg
const float PESO_CARRO_CH=1389;
//u: km/h2
const float ACELERACION_CARRO_CH=24000;
const float DISTRIBUCION_CH=0.55;
//Honda Odyssey
//u: metros
const float ANCHO_CARRO_M=2;
//u: metros
const float LARGO_CARRO_M=5.15;
//u:kg
const float PESO_CARRO_M=2041;
//u: km/h2
const float ACELERACION_CARRO_M=18000;
const float DISTRIBUCION_M=0.85;
//Sprinter
//u: metros
const float ANCHO_CARRO_G=2.31;
//u: metros
const float LARGO_CARRO_G=5.69;
//u: kg
const float PESO_CARRO_G=3311.22;
//u: km/h2
const float ACELERACION_CARRO_G=14400;
const float DISTRIBUCION_G=0.95;

//Camión
//u: metros
const float ANCHO_CARRO_EG=2.6;
//u: metros
const float LARGO_CARRO_EG=11.58;
//u: kg
const float PESO_CARRO_EG=10205.82;
//u: km/h2
const float ACELERACION_CARRO_EG=10285;
const float DISTRIBUCION_EG=1;

//u: km/h
const float VELOCIDAD_DESEADA_PROMEDIO=60;
//u: km/h
const float VARIACION_VELOCIDAD_DESEADA_PROMEDIO=20;
//u: km/h2
const float VARIACION_ACCELERACION=5000;

```

7.3 TABLA DE FIGURAS

Figura 1: Calle.....	13
Figura 2: Curva	15
Figura 3: Conexión	18
Figura 4: Boulevard	19
Figura 5: Camino sinuoso	19
Figura 6: Descomposición de camino sinuoso en elementos simples	20
Figura 7: Bifurcación	20
Figura 8: Salida de autopista.....	21
Figura 9: Glorieta.....	22
Figura 10: Descomposición de glorieta en elementos simples	22
Figura 11: Intersección de 8 calles.....	23
Figura 12: Herramienta de extracción de coordenadas.....	29
Figura 13: Medición de referencia inicial de Google Maps	29
Figura 14: Vista de imagen satelital para extracción de coordenadas	31
Figura 15: Vista de curva sobrepuesta a imagen satelital	31
Figura 16: Visualización de archivo de posiciones.....	36
Figura 17: La Glorieta Santa Fe.....	47
Figura 18: Imagen satelital de la Glorieta Santa Fe.....	48
Figura 19: Imagen procesada de la glorieta	49
Figura 20: División de la glorieta en sus 49 elementos básicos	50
Figura 21: Elementos básicos (acercamiento 1)	51
Figura 22: Elementos básicos (acercamiento 2)	51
Figura 23: Software Extractor De Coordenadas	52
Figura 24: Medición del escenario usando Google Maps.....	53
Figura 25: Sección del escenario	54
Figura 26: Curvas y calles sobrepuestas en la imagen satelital	56
Figura 27: Carga de trabajo en Laptop al correr Analizador De Tráfico.....	80
Figura 28: Carga de trabajo en Atupal al correr Optimizador De Tráfico.....	81
Figura 29: Pantalla de Bienvenida del software SimTraf	84
Figura 30: Pantalla de controles del software SimTraf.....	85
Figura 31: Estado visual de SimTraf después de cargar la animación	86
Figura 32: Información de progreso presentada por SimTraf.....	86
Figura 33: Acercamiento utilizando la característica de zoom del software	87
Figura 34: Capacidad de acercamiento del software	88
Figura 35: Situación vehicular en el minuto 19 en el escenario de referencia	90
Figura 36: Situación vehicular en el minuto 19 para el escenario optimo.....	90

8 BIBLIOGRAFÍA

Beaty, William. "The Physics Behind Traffic Jams." Smart motorist, n.d. <<http://www.smartmotorist.com/traffic-and-safety-guideline/traffic-jams.html>>.

Conservancy, Software Freedom. *Git*. n.d. 2016. <<https://git-scm.com/>>.

Daniel Robles, Pablo Ñañez, Nicanor Quijano. "Urban Traffic Control and Simulation in Colombia." Bogotá: Universidad de los Andes, 31 March 2009.

Embarcadero. n.d. <<https://www.embarcadero.com/products/cbuilder>>.

Fotherby, Thomas. "Visual Traffic Simulation." Department of Computing Imperial College, June 2002. <<http://www.doc.ic.ac.uk/~tf98/Project/ExampleWebSite>>.

GIMP - GNU Image Manipulation Program. n.d. 2016. <<https://www.gimp.org/>>.

Google Maps. n.d. 2016. <<https://maps.google.com/>>.

Gorup, PTV. <http://company.ptvgroup.com/>. n.d. 2016.

Gross, Neil R. "SCATS Adaptive Traffic System." *TRB Adaptive Traffic Control Workshop*. Transcore, January 2000. <<https://www.transcore.com/>>.

Haberman, Richard. *Mathematical models; mechanical vibrations, population Dynamic and Traffic Flow*. Prentice-Hall, 1977.

NAN Nan, WANG Ning. "The Study on the Sustainable Development of Traffic in Small Town." Baoding: Hebei Agricultural University, n.d.

Padhye, Kalyanmoy Deb and Nikhil. *Improving a Particle Swarm Optimization Algorithm*. Kanpur Genetic Algorithms Laboratory, 2010.

Prof. L.H. Immers, S. Logghe. "Traffic Flow Theory." *Basics of Traffic Engineering*. HEVERLEE: KATHOLIEKE UNIVERSITEIT LEUVEN, May 2002.

Resnick, Mitchel. *Turtles, termites and traffic jams: Explorations in massively parallel microworlds*. MIT Press, 1998.

Sebastien Feve, Alexandre Huret. "Vehicle Steering Study." Ecole Nationale d'Ingenieurs de Metz, n.d.

Snider, Jarrod M. "Automatic Steering Methods for Autonomous." Pittsburgh: Robotics Institute Carnegie Mellon University, February 2009.

Stephan Schal, Wolfgang Hugemann. "Speed Bumps – Calming or Threatening Traffic?" *«Verkehrsunfall und Fahrzeugtechnik» («Traffic accident and vehicle technique»)*. Leverkusen,, 1992.

Taylor, Mark. "Applying ASCT to Small Scale Systems SCATS & ACSLite." Utah Department of Transportation, n.d.