

# Traffic Lights Optimization

INTEGRANTES:

LEANDRO RODRÍQUEZ LLOSA

LAURA V. RIERA PÉREZ

MARCOS M. TIRADOR DEL RIEGO

GRUPO: C-311

Tercer año. Ciencias de la Computación.

Facultad de Matemática y Computación, Universidad de La Habana, Cuba

Enero 2023

## I. REPOSITORIO DEL PROYECTO

<https://github.com/science-engineering-art/traffic-lights>

## II. DESCRIPCIÓN

En todo el mundo, la congestión del tráfico sigue siendo un problema importante en la mayoría de las ciudades, debido al creciente número de vehículos privados, de mercancías y de transporte público. Este fenómeno afecta, sobre todo en horas pico, a los usuarios de la red vial, los cuales pierden mucho tiempo en la carretera; además de incidir de manera negativa en el medio ambiente pues los carros se encuentran más tiempo encendidos liberando gases a la atmósfera.

Se puede pensar en varias soluciones para este problema:

1. Construcción de nuevas carreteras: Muchas veces esto no es posible debido a las condiciones geográficas, y más importante aún, es muy costoso, por lo que en general no es una solución viable.
2. Mejora del sistema de señalización vial: Es más sensata pues se relaciona inteligentemente con la infraestructura existente. Es de especial interés la mejora de los semáforos ya que estos controlan el flujo de la

red vial de la ciudad. En estos podemos tener:

- Plan de luces fijo (Estático): se fijan los tiempos de verde y rojo en cada línea de luces de una intersección así como su secuencia una sola vez teniendo en cuenta las previsiones de tráfico, y estas no cambian.
- Controladores de tiempo real (Dinámicos): en técnicas de tiempo real, el sistema debe ser capaz de adaptarse inmediatamente (o muy brevemente) a las condiciones del tráfico. Dicho sistema posee algoritmos que permiten controlar el tráfico, los cuales reciben información sobre el estado del tráfico, que ha sido recolectada por los sensores colocados en cada carril, y recalculan la duración y la sincronización de la luces para minimizar la congestión, es decir, para minimizar el tiempo promedio de espera en las luces, y la duración de colas.

## I. Objetivo

Creación de un algoritmo de control que determine de la secuencia de fases y el tiempo de de luz verde óptimos en los semáforos de las intersecciones, con el fin de hacer más fluido el tráfico y minimizar las colas.

### III. SIMULACIÓN

#### I. Modelo microscópico

Para modelar el flujo del tráfico se utiliza un modelo microscópico, en donde se representan los vehículos de forma independiente y se intenta replicar el comportamiento de un conductor. En consecuencia, debe ser un sistema multiagente, es decir, cada vehículo opera por sí mismo utilizando información de su entorno.

Cada vehículo está identificado por un número  $i$ . El  $i$ -ésimo vehículo sigue al  $(i-1)$ -ésimo vehículo. Para el  $i$ -ésimo vehículo, se denota por  $x_i$  su posición a lo largo del camino,  $v_i$  su velocidad y  $l_i$  su longitud. Sean, además,  $s_i$  la distancia parachoques a parachoques y  $\Delta v_i$  la diferencia de velocidad entre el  $i$ -ésimo vehículo y el vehículo que le precede (vehículo número  $i-1$ ).

$$\begin{aligned} s_i &= x_i - x_{i-1} - l_i \\ \Delta v_i &= v_i - v_{i-1} \end{aligned}$$

#### II. Modelo Conductor Inteligente

El Modelo de Conductor Inteligente describe la aceleración del  $i$ -ésimo vehículo en función de sus variables y las del vehículo que le precede. La ecuación dinámica se define como:

$$\begin{aligned} \frac{dv_i}{dt} &= a_i \left( 1 - \left( \frac{v_i}{v_{0,i}} \right)^\delta - \left( \frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \right) \\ s^*(v_i, \Delta v_i) &= s_{0,i} + v_i T_i + \frac{(v_i \Delta v_i)}{\sqrt{(2a_i b_i)}} \end{aligned}$$

donde:

- $s_{0,i}$  : es la distancia mínima deseada entre el vehículo  $i$  e  $i-1$ .
- $v_{0,i}$  : es la velocidad máxima deseada del vehículo  $i$ .
- $\delta$  : es el exponente de la aceleración y controla la "suavidad" de la aceleración.
- $T_i$  : es el tiempo de reacción del conductor del  $i$ -ésimo vehículo.
- $a_i$  : es la aceleración máxima del vehículo  $i$ .
- $b_i$  : es la desaceleración cómoda para el vehículo  $i$ .

- $s^*$  : es la distancia real deseada entre el vehículo  $i$  e  $i-1$ .

Interpretando los términos en  $s^*$ :

- $v_i T_i$  : es la distancia de seguridad del tiempo de reacción. Es la distancia que recorre el vehículo antes de que el conductor reaccione (frene). Dado que la velocidad es la distancia entre el tiempo, la distancia es la velocidad por el tiempo.
- $(v_i \Delta v_i) / \sqrt{(2a_i b_i)}$ : es una distancia de seguridad basada en la diferencia de velocidad. Representa la distancia que tardará el vehículo en reducir la velocidad (sin chocar con el vehículo de delante), sin frenar demasiado (la deceleración debe ser inferior a  $b_i$ ).

#### III. Modelo de red vial de tráfico

Para modelar la red vial se utiliza un grafo dirigido, donde las aristas representan las calles (cuadras) y los vértices las intersecciones. Cada vehículo tiene un camino que consta de múltiples calles. Se aplica el Modelo Conductor Inteligente para vehículos en la misma calle. Cuando un vehículo llega al final de la calle, se retira de la misma y es añadido a la siguiente.

#### IV. Generación de vehículos

Para la generación de vehículos se utiliza la distribución de Poisson. Esta es una distribución de probabilidad discreta, describe situaciones en las cuales los clientes llegan de manera independiente durante un cierto intervalo de tiempo y el número de llegadas depende de la magnitud del intervalo.

$$p(x) = f(x, \lambda) = \frac{\lambda^x \cdot e^{-\lambda}}{x!}$$

#### V. Semáforos

Los semáforos tienen dos zonas en las que los vehículos se comportan de forma diferente:

- Zona de ralentización: Zona en la que los vehículos reducen su velocidad máxima utilizando un factor de ralentización.

$$v_{0,i} := \alpha v_{0,i} \text{ donde } \alpha < 1$$

- Zona de parada: Zona en la que se detienen los vehículos. Esto se logra utilizando una fuerza de amortiguamiento a través de la siguiente ecuación dinámica:

$$\frac{dv_i}{dt} = -b_i \frac{v_i}{v_{0,i}}$$

## VI. Curvas de Bézier

Una curva de Bézier es una curva polinomial que aproxima a una serie de puntos llamados "puntos de control". Esta curva puede ser de cualquier grado, y podemos decir que de una curva de grado  $n$  aproxima a  $n + 1$  puntos de control.

Entonces para producir una curva en nuestro mapa se crean las calles (rectas) y se utilizan las curvas de Bézier como un spline de suavizado.

## IV. INTELIGENCIA ARTIFICIAL

### I. Algoritmo genético

#### i.1. Solución

La solución será un vector que contiene enteros correspondientes a los tiempos de luz verde de cada semáforo de la intersección.

Nota: Para realizar las operaciones de cruzamiento y mutación los tiempos se llevan a binario, teniendo así más dígitos, y por tanto posibilidades, con los que jugar y encontrar la nueva población.

#### i.2. Población inicial

La población inicial se hallará de manera aleatoria, en donde cada uno de los valores de tiempo de luz verde estará entre el tiempo promedio que le toma a un carro pasar por la intersección y el tiempo máximo que un vehículo puede estar esperando.

#### i.3. Función de evaluación *fitness*

Se consideraron tres formas de calcular el *fitness*:

- Según el máximo tiempo de espera.
- Según el tiempo total de paso de los carros por una calle.
- Según la media ponderada, calculada sobre todas las calles, del tiempo promedio que toma a los carros cruzar una calle

#### i.4. Selección de padres

Para escoger los padres se utilizó la selección basada en el rango (no el valor numérico) de los valores de *fitness* de los individuos en la población. La selección basada en el rango reduce los efectos potencialmente dominantes de individuos de alto *fitness*, comparativamente, en la población, estableciendo una cantidad predecible y limitada de presión de selección a favor de tales individuos. Al mismo tiempo, la selección basada en el rango exagera la diferencia entre valores de *fitness* agrupados de forma cercana para que los mejores se puedan muestrear más.

#### i.5. Mutación

La función de mutación recibe una población y un factor de mutación, que representa el porcentaje de la población a ser mutada. Para mutar un individuo, se escogen  $n$  de sus genes y en estos se niegan  $m$  de sus bits (ambos  $n$  y  $m$  se escogen de forma aleatoria).

#### i.6. Cruzamiento

- **Cruce por puntos múltiples:** Se seleccionan  $p$  puntos random y los segmentos (resultado de picar los individuos por dichos puntos) alternos de los individuos se intercambian para obtener nuevos descendientes.
- **Cruce geométrico:** Operador independiente de la representación definido sobre la distancia del espacio de búsqueda. Los descendientes deben estar en el segmento entre los individuos.
- **Intermediate crossover:** Crea los descendientes mediante un promedio ponderado de los dos individuos en apareamiento.

## II. Feature

Como característica adicional al proyecto se implementó un algoritmo  $A^*$  para calcular, dadas dos calles del mapa, la menor distancia o tiempo para llegar de una a la otra. La primera versión de  $A^*$  que implementamos usa la conocida heurística, admisible y completa, de distancia en línea recta. Luego se realizó una versión un poco más complicada del algoritmo que usa una simulación para estimar el tiempo que tomará a un vehículo llegar desde una calle a otra. La idea es que se generen vehículos que tomen todas las aristas posibles cuando se está calculando el valor de  $g$  en  $f = g + h$ . Finalmente se implementó una heurística propia que consiste en aproximar el tiempo que demoraría un carro en ir desde todas las calles hasta el destino (usando el algoritmo de Dijkstra), basándonos en las distancias, concurrencia de vehículos en las calles y tiempos de las luces de los semáforos.