

Movilidad inteligente: reducción de emisión de gases

Intelligent mobility: reduction of gas emissions

Laura Carballo, Andrea Villagra, Marcelo Errecalde
mlaura.car@hotmail.com, avillagra@uaco.unpa.edu.ar, merrecalde@unsl.edu.ar
Laboratorio de Tecnologías Emergentes (LabTEm)
Instituto de Tecnología Aplicada (ITA) - Unidad Académica Caleta Olivia
Universidad Nacional de la Patagonia Austral

Recibido: 09/05/2019. Aceptado: 05/07/2019

RESUMEN

El concepto de movilidad inteligente se enmarca en el conjunto de acciones, técnicas e infraestructuras encaminadas a la mejora de la movilidad y organización del tráfico en las ciudades. Durante las últimas décadas uno de los principales problemas en las diferentes ciudades del mundo, es el congestionamiento vehicular y los efectos que esto ocasiona. Un mayor nivel de contaminación, emisiones y consumo de combustible se convirtieron en serios problemas. Aumenta también la cantidad de semáforos en las grandes urbes y la complejidad para su programación. En este trabajo se utilizan dos algoritmos metaheurísticos, el algoritmo genético celular (cGA) y el algoritmo de Recocido Simulado (en inglés, *Simulated Annealing*, SA) para optimizar la planificación de los programas de ciclos semaforicos. Las soluciones obtenidas por los algoritmos aplicados a dos grandes escenarios cercanos a la realidad en áreas urbanas ubicadas en las ciudades de Málaga (España) y París (Francia), son simuladas por el popular micro-simulador SUMO (*Simulator Urban MObility*). La comparación con un enfoque paralelo de la literatura revela la bondad del algoritmo genético celular. Además, se muestra una reducción significativa en términos de las tasas de emisión y del consumo total de combustible por parte del algoritmo genético.

Palabras clave: Problema de planificación de luces de tráfico; Algoritmo Genético Celular; Algoritmo de Recocido Simulado; Emisión de gases; Simulador SUMO.

ABSTRACT

The concept of intelligent mobility is part of the set of actions, techniques, and infrastructures aimed at improving the mobility and organization of traffic in cities. During the last decades, one of the main problems in the different cities of the world is the traffic congestion and the effects that this causes. A higher level of pollution, emissions and fuel consumption became serious problems. It also increases the number of traffic lights in large cities and the complexity for its programming. In this work, two metaheuristic algorithms are used, the cellular genetic algorithm (cGA) and the simulated annealing algorithm (SA) to optimize the scheduling of traffic light programs. The solutions obtained by the algorithms applied to two large scenarios close to reality in urban areas located in the cities of Málaga (Spain) and Paris (France) are simulated by the popular SUMO micro simulator (*Simulator Urban MObility*). The comparison with a parallel approach to the literature reveals the goodness of the cellular genetic algorithm. In addition, there is a significant reduction in terms of emission rates and total fuel consumption by the genetic algorithm.



Key words: Traffic lights scheduling problem; Cellular Genetic Algorithm; Simulated Annealing Algorithm; Emission of gases; SUMO simulator.

1. INTRODUCCIÓN

Las grandes ciudades luchan diariamente con el grave problema de la congestión vehicular y sus efectos como la contaminación ambiental, desgaste de los vehículos por constante aceleración y frenado, retrasos y prolongadas horas de espera. Los mencionados efectos acarrearán una disminución en la calidad de vida debido a problemas de estrés al que constantemente se someten conductores y peatones al circular por las diferentes ciudades. La reducción de la productividad es otra consecuencia, debido a los tiempos perdidos en el camino a los lugares de trabajo.

La búsqueda de soluciones a este problema ha cobrado notable interés, sobre todo por la relación directa existente con los altos niveles de contaminación presentados en los últimos años, por el incremento en la población mundial, el crecimiento de las ciudades y la multiplicación de vehículos que circulan por las rutas. Estamos ante un problema de índole global, que afecta diariamente a millones de personas. Sin embargo, la optimización del tráfico es una tarea bastante compleja por la gran cantidad de parámetros involucrados en su análisis. Una de las alternativas de solución a este problema es la optimización de la duración de los ciclos de los semáforos, ya que se consideran una opción de bajo costo, al no requerir infraestructura adicional.

Numerosos autores han considerado diversos aspectos de este problema de tráfico y en particular la planificación de los ciclos de los semáforos en variadas intersecciones o cruces. Durante la última década, las metaheurísticas se han utilizado ampliamente para abordar este problema.

Telku et al. (2007) consideran el problema de optimizar la señal verde y los tiempos de ciclo en una red urbana, de tal manera que la optimización anticipa el impacto en los patrones de enrutamiento del tráfico utilizando un Algoritmo Genético (GA). La propuesta se aplica a un caso de estudio de la ciudad de Chester en el Reino Unido. Los resultados muestran un mejor rendimiento de la temporización de la señal optimizada por el método GA en comparación con otro método de la literatura. Esta mejora es más significativa con una red más congestionada, mientras que, en una situación de congestión relativamente leve, la mejora no es muy clara.

Sánchez et al. (2004) proponen una arquitectura para la optimización de los ciclos semafóricos utilizando un GA como técnica de optimización y simuladores de autómatas celulares (CAS) dentro de la función de evaluación. En Sánchez et al. (2008) se ha aplicado la arquitectura para un caso de la vida real obteniendo resultados promisorios.

También, para este problema del control del ciclo semafórico en García-Nieto et al. (2013) proponen un enfoque de optimización de enjambre de partículas (PSO) (Eberhart y Kennedy, 1995) utilizando como función de evaluación un simulador de tráfico microscópico, aplicándolo en un caso de estudio de una red de tráfico de la vida real obteniendo beneficios significativos con la propuesta.

Souravlias et al. (2016) proponen para este problema un enfoque paralelo mediante el uso de Evolución Diferencial (DE) (Storn, y Price, 1997), se utiliza el popular micro-simulador SUMO (Krajzewicz et al., 2006) y se evalúa el enfoque en escenarios de problemas cercanos a la realidad que consisten en dos grandes áreas urbanas ubicadas en las ciudades de Málaga, España y París, Francia. Los resultados obtenidos con la propuesta presentan mejoras significativas comparadas con resultados de otros algoritmos de la literatura.

Utilizando Optimización Basada en Colonia de Hormigas (ACO) (Dorigo, 1992) He y Hou (2012) se centran en tres parámetros para comparar el rendimiento de su modelo de optimización: el menor retardo de tiempo, el menor número de paradas y la mayor capacidad de tráfico. Los resultados se comparan con valores obtenidos por un GA y la ecuación de Webster, una fórmula muy efectiva en condiciones de tráfico altamente saturadas, y muestran que ACO tiene una mayor ventaja de rendimiento en relación con un menor retardo de tiempo, menos paradas y mayor capacidad de tráfico.

Yuan et al. (2014) emplearon ACO y el tiempo efectivo de verde como variables de decisión para determinar los planes de sincronización de señales en una red de tráfico regional. Presentan un modelo de tiempo discreto que ayuda a analizar los factores que causan cuellos de botella formados por combinación de la sincronización de las señales en las intersecciones con propiedades estáticas de los carriles de giro a la izquierda y en línea recta. Los autores creen que los indicadores de cuellos de botella juegan un papel importante en la identificación y optimización de cuellos de botella de tráfico. Este trabajo es una contribución novedosa en términos de identificar cuellos de botella en el tráfico y en el enfoque propuesto que reduce los impactos de la congestión del tráfico.

Baskan y Haldenbilen, (2011) y Baskan et al. (2009) desarrollan una variante de la técnica ACO, denominada ACORSES, donde se han utilizado heurísticas para reducir el espacio de búsqueda de la solución potencial de un problema de optimización de tráfico de dos niveles. En Jabbarpour et al. (2014) se presenta una revisión global de sistemas de control de congestión y tráfico de vehículos basados en hormigas.

Péres et al. (2018) proponen un enfoque evolutivo multiobjetivo para la optimización de los programas de ciclos de luces de tráfico con el objetivo de mejorar el flujo de vehículos y reducir las emisiones de los vehículos. Han encontrado mejoras significativas en cuanto al tiempo de viaje de los vehículos y a la emisión producida. Los experimentos se llevaron a cabo en tres áreas de la ciudad de Montevideo, Uruguay. Recientemente, Nesmachnow et al. (2019) han aplicado su propuesta al problema de Autobús de Tránsito Rápido logrando mejoras con respecto a versiones secuenciales.

Si bien los algoritmos metaheurísticos últimamente han sido aplicados a este problema por parte de la comunidad científica, todavía hay mejoras en cuanto a la disminución de la emisión de gases y consumo de combustible que pueden plantearse. En particular, utilizando un algoritmo que ha resultado ser muy competitivo en la resolución de problemas de optimización de la vida real como es el Algoritmo Genético Celular.

El presente trabajo se centra en la aplicación de dos técnicas metaheurísticas el Algoritmo Genético Celular (Dorronsoro y Alba, 2008) y el algoritmo de Recocido Simulado (Kirkpatrick, 1983) aplicados a la optimización de la duración de los ciclos de los semáforos, para topologías de entornos reales de grandes ciudades (instancias de Málaga y de París). Cabe aclarar que se usan estas instancias porque son las utilizadas en trabajos previos y sobre

el cual se realizan comparaciones. Se realiza un análisis en profundidad de los resultados obtenidos por los algoritmos, en particular en cuanto a los gases emitidos.

El artículo está organizado de la siguiente manera: en la Sección 2 se describe el problema, en la Sección 3 se muestran detalles del simulador utilizado y se presenta una descripción de las emisiones vehiculares, en la Sección 4 se especifican los algoritmos utilizados para resolver el problema SA y cGA. La Sección 5 explica cómo se representa el problema y la estrategia de solución. En la Sección 6 se presenta el diseño de experimentos y en la Sección 7 se analizan los resultados. Finalmente, en la Sección 8 se describen las conclusiones y trabajos futuros.

2. FORMULACION DEL PROBLEMA

Una red de tráfico urbano está compuesta por intersecciones, luces de tráfico, rutas, direcciones y vehículos en movimiento sobre sus rutas. Las luces de tráfico están ubicadas en intersecciones y el control del flujo de los vehículos esta dado por sus programas de estados de color y tiempo de los ciclos (duración de las fases). Para todas las luces de trafico ubicadas en una misma intersección la combinación de estados de color durante un ciclo es siempre válido y sigue las reglas de tráfico para intersecciones a fin de evitar colisiones.

Para describir matemáticamente el problema de estudio se emplea la representación utilizada por García-Nieto et al. (2013). Este problema se puede ver como un problema que tiene múltiples objetivos. Se pretende:

- maximizar el número de vehículos que llegan a su destino, NV_D o minimizar el número de vehículos que no llegan a destino, NV_{ND} , durante el tiempo de simulación T_S ;
- minimizar el tiempo total de viaje de los vehículos, TT_V (desde que arriba a la simulación hasta que llega a destino);
- minimizar el tiempo de paradas y de espera de todos los vehículos, TT_{EP}
- maximizar el radio P de colores verde y rojo en cada estado de fase de todas las intersecciones, definida como:

$$P = \sum_{i=0}^{intr} \sum_{j=0}^{fs} d_{i,j} \frac{v_{i,j}}{r_{i,j}} \quad (1)$$

donde $intr$ corresponde la cantidad de todas las intersecciones, fs representa la cantidad de todas las fases, y $v_{i,j}$, $r_{i,j}$ representan el numero de señales en color verde y color rojo, respectivamente, en la intersección i y la fase j , con duración $d_{i,j}$.

Se combinan todos los objetivos en una única función objetivo representada de la siguiente forma:

$$f_{obj} = \frac{TT_V + TT_{EP} + NV_{ND} T_S}{NV_D^2 + P} \quad (2)$$

Las emisiones de los vehículos se ven influenciadas por la programación de los tiempos de los ciclos de las luces de tráfico (Kun y Lei, 2007).

Este problema de planificación de la duración de los ciclos semafóricos es complejo y presenta enormes espacios de búsqueda. Por ejemplo, una simple intersección con 8 fases de semáforo representa 55^8 (más de $8,3 \times 10^{14}$) posibles soluciones. Además, se requiere del uso de simuladores para evaluar la calidad de las configuraciones de semáforos candidatas. Estos simuladores consumen mucho tiempo y es necesario contar con un algoritmo eficiente para la obtención de dichas soluciones.

3. SIMULADOR DE TRÁFICO SUMO Y EMISIONES VEHICULARES

En esta sección, se describe la herramienta de simulación utilizada para evaluar las soluciones obtenidas por los algoritmos (SA y cGA) y características de los gases emitidos por los vehículos y devueltos por SUMO luego de la simulación.

El simulador SUMO es una herramienta de código abierto, altamente portátil, es un paquete microscópico de simulación de tráfico vial diseñado para manejar grandes redes viales. Entre las características principales del simulador se pueden destacar las siguientes: (a) Permite la simulación microscópica de los diferentes elementos involucrados en el tráfico: vehículos, peatones, transporte público, entre otros. (b) Interacción en línea, mediante TraCI (*Traffic Control Interface*). (c) Simulación de tráfico multimodal, por ejemplo, vehículos, transporte público y peatones. (d) Planificación de programas de control de semáforos, los mismos que pueden ser importados o generados automáticamente por SUMO. (e) Variedad de formatos para la importación de redes, entre los que se pueden destacar OpenStreetMap, VISUM, VISSIM y NavTeq. (f) Implementado en C++ y hace uso de bibliotecas portátiles.

Los ficheros de configuración, así como los que contienen la información de evaluación de la red, se encuentran en formato XML, y se dispone de las librerías necesarias para su conversión a formato CSV, para su mejor evaluación y análisis.

Después de la simulación, se genera un archivo (.tripinfo.xml), que contiene información detallada. Información sobre la posición del vehículo en cada momento. Esta información se utiliza para evaluar los programas de ciclo de semáforo, además de información sobre los valores de emisión de los vehículos. SUMO permite simular factores medioambientales basados en el modelo HBEFA (*HandBook Emission Factors for Road Transport*) (Hausberger et al., 2009). La simulación devuelve información de los siguientes gases CO₂, CO, NO_x, PM_x, y HC.

Dentro de los gases generados en la combustión, no todos son nocivos para la salud. En particular, los gases emitidos por un motor de combustión interna de gasolina son, principalmente, de dos tipos: inofensivos y contaminantes. Los primeros están formados, fundamentalmente, por Nitrógeno, Oxígeno, Dióxido de Carbono, vapor de agua e Hidrógeno. Los segundos o contaminantes están formados, fundamentalmente, por el Monóxido de Carbono, Hidrocarburos, Oxidos de Nitrógeno y materia Particulada.

El dióxido de carbono (CO₂) se produce cuando se queman combustibles que contienen carbono (por ejemplo, gasolina, combustible diesel). El carbono se combina durante esta operación con el oxígeno aspirado. El CO₂, a pesar de ser un gas no tóxico, reduce el estrato de la atmósfera de la Tierra que generalmente sirve como protección contra la penetración de los rayos UV (la tierra se calienta), produce alteraciones de clima (efecto “invernadero”).

El monóxido de carbono (CO) se produce debido a la combustión incompleta de carbono que contiene y combustibles altamente tóxicos. Es un gas incoloro, inodoro, explosivo y altamente tóxico. Bloquea el transporte de oxígeno por parte de los glóbulos rojos. Es mortal, incluso en una baja concentración en el aire que respiramos. En una concentración normal en el aire ambiental se oxida al corto tiempo, formando dióxido de carbono CO₂. Los óxidos nítricos (NO_x) son combinaciones de nitrógeno N₂ y oxígeno O₂ (por ejemplo, NO, NO₂, N₂O). Los óxidos de nitrógeno se producen al existir una alta presión, alta temperatura y exceso de oxígeno durante la combustión en el motor. El monóxido de nitrógeno (NO), es un gas incoloro, inodoro e insípido. Al combinarse con el oxígeno del aire, es transformado en dióxido de nitrógeno (NO₂), de color pardo rojizo y de olor muy penetrante, provoca una fuerte irritación de los órganos respiratorios. Las medidas destinadas a reducir el consumo de combustible suelen conducir lamentablemente a un ascenso de las concentraciones de óxidos nítricos en los gases de escape, porque una combustión más eficaz produce temperaturas más altas. Estas altas temperaturas generan a su vez una mayor emisión de óxidos nítricos.

Los hidrocarburos (HC) son restos no quemados del combustible, que surgen en los gases de escape después de una combustión incompleta. La mala combustión puede ser debido a la falta de oxígeno durante la combustión (mezcla rica) o también por una baja velocidad de inflamación (mezcla pobre), por lo que es conveniente ajustar la riqueza de la mezcla. Los hidrocarburos HC se manifiestan en diferentes combinaciones y actúan de diverso modo en el organismo. Algunos de ellos irritan los órganos sensoriales, mientras que otros son cancerígenos.

La materia particulada (PM_x) es generada en su mayor parte por los motores diesel, se presentan en forma de hollín o cenizas. Los efectos que ejercen sobre el organismo humano todavía no están aclarados por completo.

4. ALGORITMOS PROPUESTOS

En esta sección se describen los dos algoritmos utilizados para resolver el problema de planificación de los ciclos semafóricos. Primero se describe la metaheurística de trayectoria SA y en segundo lugar la metaheurística poblacional cGA.

4.1 Descripción de SA

Simulated Annealing es una generalización de un método de Monte Carlo para la evaluación de las ecuaciones de estados y estados congelados de sistemas de n-cuerpos (Metropolis, 1953). SA se basa en una analogía de la termodinámica que se ocupa de la manera en que los metales se enfrían. Si se enfría un metal líquido lentamente, sus átomos forman un cristal puro correspondiente al estado de energía mínima para el metal. El metal llega a un estado de menor energía si se enfría rápidamente. SA es una técnica de búsqueda local probabilística, para resolver problemas de optimización, que ha recibido significativa atención en las últimas dos décadas. Esta metaheurística se ha convertido en uno de los muchos enfoques diseñados para dar solución óptima o cercana al óptimo. Es simple de formular, maneja problemas discretos y continuos, y tiene un bajo requisito de memoria, consume menos tiempo de CPU que un GA porque busca la solución óptima utilizando una iteración punto a punto en lugar de una búsqueda a través de una población de individuos. SA es un sistema dinámico, por ende, es necesario trabajar mediante la simulación de relajaciones sucesivas de un sistema físico ficticio a temperaturas cada vez más bajas. Eso significa que, SA evoluciona a través de una

secuencia de transiciones entre estados; las cuales son generadas por probabilidades de transición. En consecuencia, SA puede ser matemáticamente modelado por cadenas de Markov, donde se genera una secuencia de cadenas usando una probabilidad de transición cuyo cálculo involucra a la temperatura actual.

Como toda metaheurística basada en trayectoria, SA trabaja con una sola solución candidata. El pseudocódigo de SA se muestra en el Algoritmo 1 (Figura 1). La solución tentativa inicial se crea al azar. Entonces, para obtener una vecindad de la solución, se aplica una perturbación de esa solución. Si la nueva solución es mejor que la solución anterior (es decir, tiene una función de aptitud inferior) se convierte en la solución actual. Si no, puede ser utilizada de todos modos, pero con una probabilidad regulada por un parámetro de temperatura decreciente llamado probabilidad de Boltzmann $e^{-((f(s') - f(s))/\tau)}$, donde $f(s')$ es el valor de aptitud de la solución actual, $f(s)$ es el valor de aptitud anterior, T es el parámetro temperatura. Después de eso, la temperatura se decreta en cada iteración, disminuyendo así la posibilidad de que peores soluciones que sean aceptadas (l representa la longitud de la cadena de Markov, línea 4 en el Algoritmo de SA). El proceso iterativo termina cuando se alcanza el criterio de parada (por ejemplo, número máximo de iteraciones) y se devuelve la solución encontrada.

Algoritmo 1 Pseudocódigo de SA

```

1:  $\tau \leftarrow t$  /* Temperatura Inicial */
2:  $s \leftarrow s_0$  /* Solución inicial candidata */
3: repeat
4:   for  $i \leftarrow 1$  to  $l$  do
5:      $s' \leftarrow$  Perturbar( $s$ )  $s' \in$  vecindario( $s$ )
6:     if  $f(s') < f(s)$  then
7:        $s \leftarrow s'$  /*  $s'$  reemplaza a  $s$  */
8:     else
9:       Aceptar  $s'$  como nueva solución con probabilidad  $(e^{-\frac{f(s')-f(s)}{\tau}})$ 
10:    end if
11:  end for
12:  Actualizar  $\tau$ 
13: until condicionParada
14: return  $s$ 

```

Figura 1: Pseudocódigo de SA

4.2 Descripción de cGA

El cGA es una clase particular de Algoritmo Evolutivo (EA) y una subclase de los Algoritmos Genéticos con una población estructurada espacialmente, es decir, los individuos de la población pueden aparearse con sus vecinos (Dorrnsoro y Alba, 2008). Estos pequeños vecindarios solapados ayudan a explorar el espacio de búsqueda debido a una difusión lenta de las soluciones permitiendo la exploración y una explotación que se lleva a cabo en cada vecindario a través de los operadores genéticos. En un cGA la población está usualmente estructurada en una rejilla toroidal de 2 dimensiones. La Figura 2 muestra una población toroidal. La estructura de vecindario comúnmente utilizada es $L5$ (Whitley, 1993). El vecindario contiene cinco individuos: el individuo central, y los individuos respectivamente en las posiciones Norte, Este, Oeste y Sur.

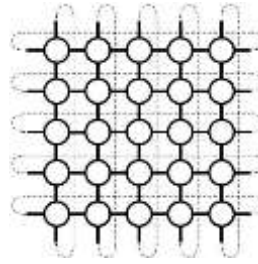


Figura 2: Población toroidal

Como familia de un GA, el cGA codifica las variables de decisión de un problema de búsqueda en cadenas de variables de longitud finita de algún alfabeto de cierta cardinalidad. Las cadenas son soluciones candidatas y se llaman cromosomas. De la misma forma que en un GA a cada una de las variables que forman el cromosoma se las denomina gen y alelo a los distintos valores que pueden tomar los genes. Codificado el problema que resolver a través de uno o varios cromosomas (también llamados individuos) y teniendo definida la función de aptitud, se evolucionan las soluciones al problema, es decir la población de soluciones teniendo en cuenta los siguientes pasos: Inicialización, evaluación, selección, recombinación, mutación y reemplazo.

En el Algoritmo 2 (Figura 3) se presenta el pseudocódigo de un cGA estándar. Comienza generando y evaluando una población inicial. Luego se aplican iterativamente los operadores genéticos (selección, recombinación, mutación y reemplazo) a cada individuo hasta que se alcance la condición de finalización. La población se estructura en una rejilla toroidal de dos dimensiones (*rejilla_cGA*) y el vecindario definido contiene cinco individuos (línea 4). El individuo que se está considerando (individuo (*x,y*)) siempre es seleccionado como primer padre (línea 5). El segundo padre se selecciona con Selección por Torneo (línea 6). Se aplican los operadores genéticos (recombinación y mutación) a los individuos (líneas 7 y 8). Se utiliza aquí un operador de crossover de dos puntos (*DPXI*) es cual produce un solo individuo (que contiene la porción más grande del mejor padre) y el operador tradicional de mutación binaria (intercambio de un bit). Luego, el algoritmo calcula el valor de fitness del nuevo individuo, es decir del hijo (línea 9) y lo inserta en la población, reemplazando al individuo actual (línea 10) siguiendo la política de reemplazo definida.

Algoritmo 2 Pseudocódigo de cGA

```

1: for  $s \leftarrow 1$  to  $MAX\_PASOS$  do
2:   for  $x \leftarrow 1$  to  $ANCHO$  do
3:     for  $y \leftarrow 1$  to  $ALTO$  do
4:       vecinos  $\leftarrow$  CalculaVecinos (rejilla_cGA,individuo( $x,y$ ));
5:       padre1  $\leftarrow$  individuo( $x,y$ );
6:       padre2  $\leftarrow$  Seleccion(vecinos);
7:       hijo  $\leftarrow$  Recombinacion (rejilla_cGA,  $P_c$ , padre1, padre2);
8:       hijo  $\leftarrow$  Mutacion(rejilla_cGA,  $P_m$ , hijo)
9:       Evaluacion(hijo);
10:      Añadir(rejilla_cGA, individuo( $x,y$ ), hijo);
11:    end for
12:  end for
13: end for
  
```

Figura 3: Pseudocódigo de cGA



5. REPRESENTACION Y ESTRATEGIA DE SOLUCION

Atendiendo a las sugerencias de trabajos previos (García-Nieto et al., 2012, 2013) una posible solución al problema se representa por medio de un vector de enteros positivos, cada número entero representa la duración en segundos para cada fase de cada intersección dentro del ciclo semafórico. En este contexto del problema a resolver, el principal objetivo consiste en encontrar programas de ciclos optimizados para todos los semáforos situados en una determinada área urbana. Se define un ciclo como el periodo de tiempo en el que un conjunto de semáforos (en un cruce) permanecen con sus estados de luces/colores fijos. Al mismo tiempo, estos programas deben también coordinarse con semáforos en intersecciones adyacentes, mejorando así el flujo de vehículos que circulan.

En cuanto a la representación como se dijo anteriormente y siguiendo la especificación de SUMO para la programación de ciclos, se codifican las posibles soluciones como vectores de enteros. De esta forma, cada elemento del vector (variable) representa una duración de fase de los semáforos implicados en una determinada intersección. La Figura 4 muestra un ejemplo de representación de la intersección $id = "i"$, la cual consta 4 fases con duraciones de 50, 4, 13 y 4 segundos (pasos de la simulación). En estas fases, los estados tienen 4 luces de tráfico (colores). Cada uno de ellos corresponde a uno de los cuatro semáforos ubicados en la intersección que se está analizando. En este ejemplo, la primera fase contiene los estados "GrGr" que significa que una luz está en verde (G) la otra está en rojo (r), la siguiente está en verde (G) y la última en rojo (r), durante 50 segundos. La siguiente fase cambia el estado de las cuatro luces de tráfico a "yryr" por 4 segundos, y así siguiendo. La última fase sigue a la primera y es lo que representa al ciclo en la simulación.

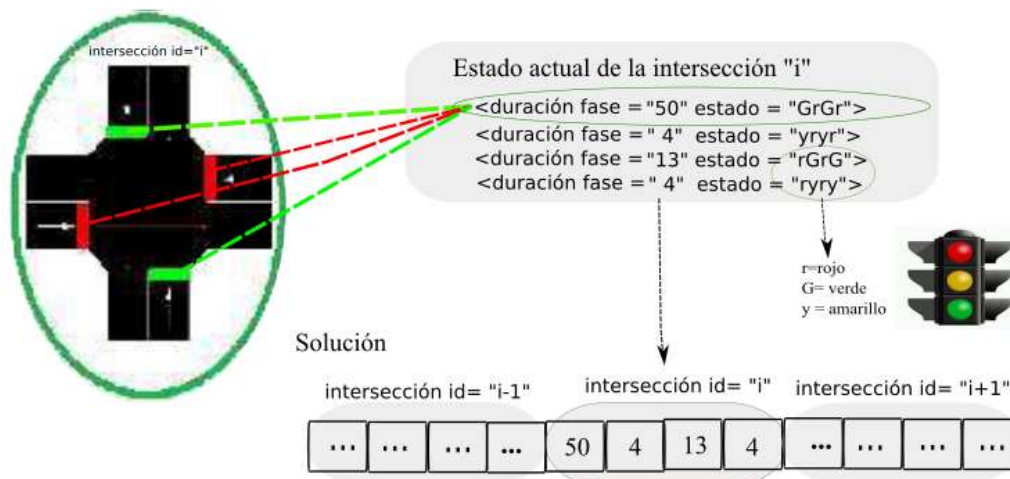


Figura 4: Una intersección con cuatro luces de tráfico, estado de la intersección para SUMO. Representación de la solución para cGA y SA

Cada vector solución, que representa el programa de ciclos de los programas de luces de tráfico se evalúa por el algoritmo correspondiente (cGA o SA) utilizando la Ecuación 2.

En los algoritmos las soluciones candidatas son inicializadas en el rango [5,60]. Cada valor de este rango representa las unidades de tiempo (en segundos), que la luz de tráfico correspondiente mantiene el mismo color en la señal, en este caso rojo o verde. Para la señal de color amarillo, se fijó un tiempo contante (4 segundos), que no cambia durante la simulación (valor definido por el municipio de la instancia de Málaga, España).

La estrategia de optimización se muestra en la Figura 5. Cuando el algoritmo utilizado (SA o cGA) genera una nueva solución, esta es inmediatamente utilizada para actualizar el programa de ciclos, entonces se ejecuta SUMO, para simular el escenario de una instancia con calles, direcciones, obstáculos, luces de tráfico, vehículos, rutas, etc. Con el nuevo estado del programa de ciclos. Luego de la simulación, SUMO devuelve la información global necesaria para calcular la calidad de la solución (*fitness*).

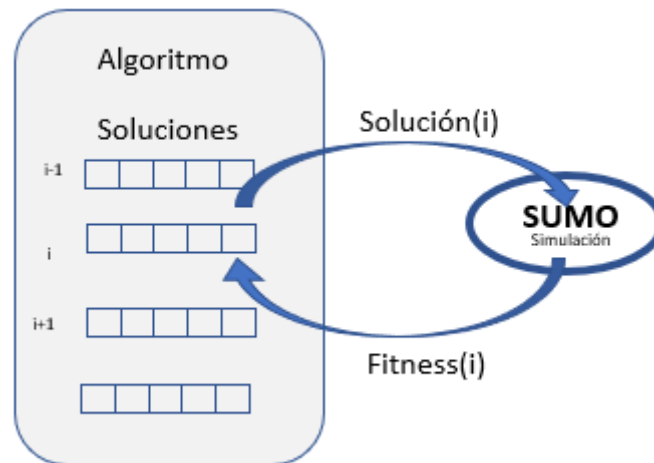


Figura 5: Estrategia de optimización. El algoritmo invoca a SUMO para evaluar cada solución

6. DISEÑO DE EXPERIMENTOS

En esta sección se presentan características de las instancias utilizadas, la parametrización de los algoritmos y el diseño de los experimentos.

Para evaluar la performance de los algoritmos se tomaron dos instancias de problemas ubicadas en dos grandes áreas metropolitanas (Málaga y París) que han sido utilizadas en el trabajo de Souravlias (Souravlias et al., 2016) y con el cuál se comparan los resultados. Las instancias consideradas fueron creadas a partir de la extracción de información de mapas digitales reales. La instancia de Málaga cuenta con 190 luces de tráfico y la instancia de París con 378 luces de tráfico.

En cuanto a los algoritmos, las parametrizaciones para cGA son las siguientes: (a) el tamaño de la población es de 100 soluciones (individuos); (b) para la recombinación, uno de los padres es el individuo actual considerado en el bucle de cGA, mientras que el otro es obtenido utilizando la Selección de torneos en su vecindario; (c) el método de recombinación es DPX1, es cual produce un solo individuo (que contiene la porción más grande del mejor padre), con una aplicación de probabilidad establecida en 1.0; (d) la mutación utilizada es entera con una probabilidad de establecida en $1 / L$ donde $L =$ longitud de la solución. Finalmente, (e) el criterio de reemplazo es "Si no es peor" con un reemplazo síncrono. El número máximo de evaluaciones fue establecido en 30.000. En cuanto a SA la temperatura inicial se estableció en 200, cantidad de vecinos 50, grado de disminución de la temperatura $\alpha=0,01$. Los valores utilizados se han establecido en base a experimentaciones realizadas y de trabajos anteriores donde se han utilizado los algoritmos (Villagra et al., 2016) y en (Souravlias et al., 2016).

Respecto al procedimiento de simulación, cada vehículo inicia su propio viaje desde un punto de partida hasta un destino con una velocidad máxima de 50 km/h. Este límite de velocidad es típico en áreas urbanas. El tiempo de simulación se fijó en 2.200 segundos para la instancia Málaga y 3.400 segundos para la instancia de París, ya que es de mayor tamaño. La simulación se realizó ejecutando el Simulador de tráfico SUMO versión 0.31.0 para Linux.

Todos los algoritmos se implementaron en Java y se ejecutan en el clúster que consta de 16 nodos (64 núcleos) equipados con una CPU IntelCore2 Quad (Q9400) @ 2.66 GHz y 4 GB de RAM, 14 nodos (28 núcleos) equipados con Intel Pentium D a 2,8 GHz y 1 GB de RAM, un nodo (ocho núcleos) equipado con dos Intel Xeon (E5405) a 2.00 GHz y 8 GB de RAM, y un nodo (ocho núcleos) equipado con un Intel Core i7 (920) a 2.67 GHz y 4 GB de RAM. El software responsable de la gestión de estos recursos heterogéneos es HTCondor Versión 7.8.4 (Thain et al., 2005). Se realizaron 30 ejecuciones independientes de cada algoritmo con cada una de las instancias. En este trabajo, los mejores valores están marcados en negrita. En todos los experimentos se han analizado las condiciones que deben cumplirse para utilizar pruebas paramétricas y pruebas no paramétricas para el análisis estadístico con un error de probabilidad de $p = 0,05$ utilizando RStudio Versión 1.1.463. Las diferencias estadísticamente significativas entre los algoritmos se muestran con el símbolo (\bullet), mientras que la no significación se muestra con (-).

Se realizan dos experimentos para analizar el desempeño de los algoritmos utilizados. En el primer experimento se analiza el desempeño de los algoritmos (cGA y SA) y cuán precisos son comparados con los resultados de otro algoritmo de la literatura (Souravlias et al., 2016) aplicado al mismo problema. En el segundo experimento se analizan los resultados obtenidos por el algoritmo que obtiene los mejores resultados realizando un análisis en profundidad en cuanto a los gases emitidos. Para este análisis se utiliza el programa de análisis y minería de datos RapidMiner 9.2¹. Además, se ha generado conjunto de resultados del mejor algoritmo compuesto por 1000 soluciones representativas que se seleccionaron de una ejecución típica del algoritmo. A este conjunto de resultados se lo ha denominado Result-1000 y para el análisis se identificaron utilizando *K-means*² la cantidad de *clusters* con respecto a los gases analizados. Se muestra el comportamiento luego de identificados los *clusters* y se analiza la correlación entre el fitness y los gases emitidos.

7. ANALISIS DE RESULTADOS

En esta sección se presentan y analizan los resultados obtenidos por los experimentos realizados.

Antes de presentar los resultados obtenidos, es importante comentar que los algoritmos utilizados para obtener la planificación de los ciclos de las luces de tráfico obtienen un 100% de vehículos que alcanzan su destino, es decir que los 1200 vehículos de la simulación llegan a destino. Uno de nuestros objetivos es maximizar el número de vehículos que llegan a destino y hemos encontrado en la literatura trabajos previos que utilizan un número de vehículos considerablemente menor (500) y no logran este objetivo (García-Nieto et al., 2012, 2013).

¹ <https://rapidminer.com/>

² *K-means* es un algoritmo de clasificación no supervisada (clusterización) que agrupa objetos en k grupos basándose en sus características.

En la Tabla 1 se muestran los resultados en mediana de la función objetivo (fitness) obtenida por el algoritmo DE (Souravlias et al., 2016) y los utilizados en este trabajo. En la última columna denominada T/UMV corresponde a la aplicación de test estadísticos. En este caso, al compararse los resultados de dos algoritmos, se aplica *T-test* o *U de Mann-Whitney* según corresponda. Podemos observar que cGA es el algoritmo que obtiene los menores resultados en ambas instancias y además las diferencias con respecto a SA son estadísticamente significativas, lo que también se puede constatar en la Figura 6 donde se ve claramente las diferencias en las medianas de los valores de fitness, en ambas instancias se observa que los resultados de cGA son más compactos y menores que SA.

Instancia	DE	SA	cGA	T/UMV ³
Málaga	0,4809	0,6267	0,4309	(•)
Paris	0,7332	1,0881	0,6844	(•)

Tabla 1: Valores medios obtenidos por DE, SA y cGA en ambas instancias de problema

Algoritmo/ Gases	Málaga					Paris				
	CO ₂	CO	HC	NO _x	PM _x	CO ₂	CO	HC	NO _x	PM _x
SA	771,55	6,99	1,08	5,01	0,11	1007,65	10,29	1,54	6,59	0,16
cGA	703,58	5,99	0,94	4,58	0,10	902,02	8,22	1,26	5,90	0,13
T/UMW	(•)	(•)	(•)	(•)	(•)	(•)	(•)	(•)	(•)	(•)

Tabla 2: Promedio de gases emitidos (Kg/s) por cada algoritmo (SA y cGA) para ambas instancias

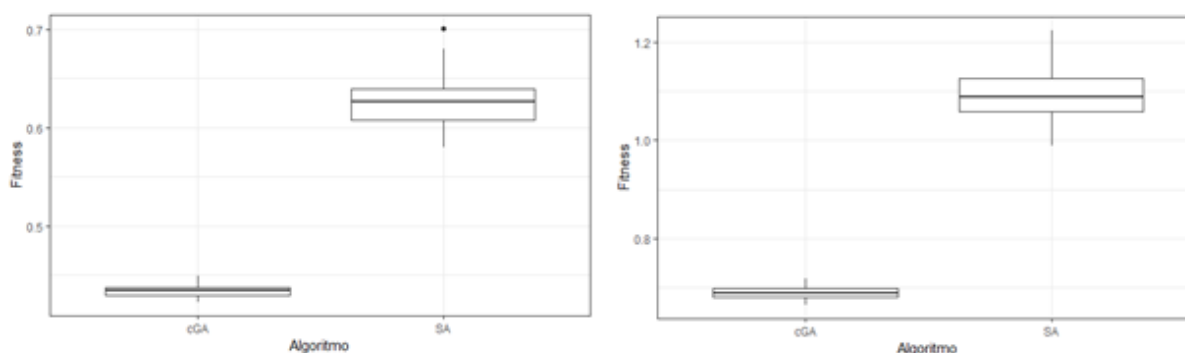


Figura 6: Box-plots de los mejores valores (Fitness) en mediana obtenidos por los algoritmos, SA y cGA para la instancia de Málaga (izquierda) y París (derecha)

Se analiza a continuación los resultados obtenidos por ambos algoritmos en cuanto a la emisión de gases. En la Tabla 2 se muestran para cada algoritmo los resultados obtenidos en cuanto a la emisión de gases y como era de esperar para ambas instancias cGA obtiene los menores valores. En todos los casos las diferencias entre los resultados obtenidos por cGA y SA son estadísticamente significativos

Para ver gráficamente las diferencias, como los resultados de los gases emitidos por los algoritmos son bastante similares, se seleccionaron solamente dos gases para cada instancia. Se muestra CO₂ y CO para Málaga mientras que HC y PM_x para Paris. La Figura 7 muestra

³ Se aplica a los resultados obtenidos por SA y cGA

los Box-plot correspondientes. Podemos observar la clara diferencia a favor de cGA con respecto a SA.

En el segundo experimento se realiza un análisis en cuanto al fitness y gases emitidos del algoritmo que obtuvo los mejores resultados. Como cGA fue el algoritmo que tuvo el mejor desempeño Result-1000 está formado por soluciones representativas de una ejecución independiente de cGA.

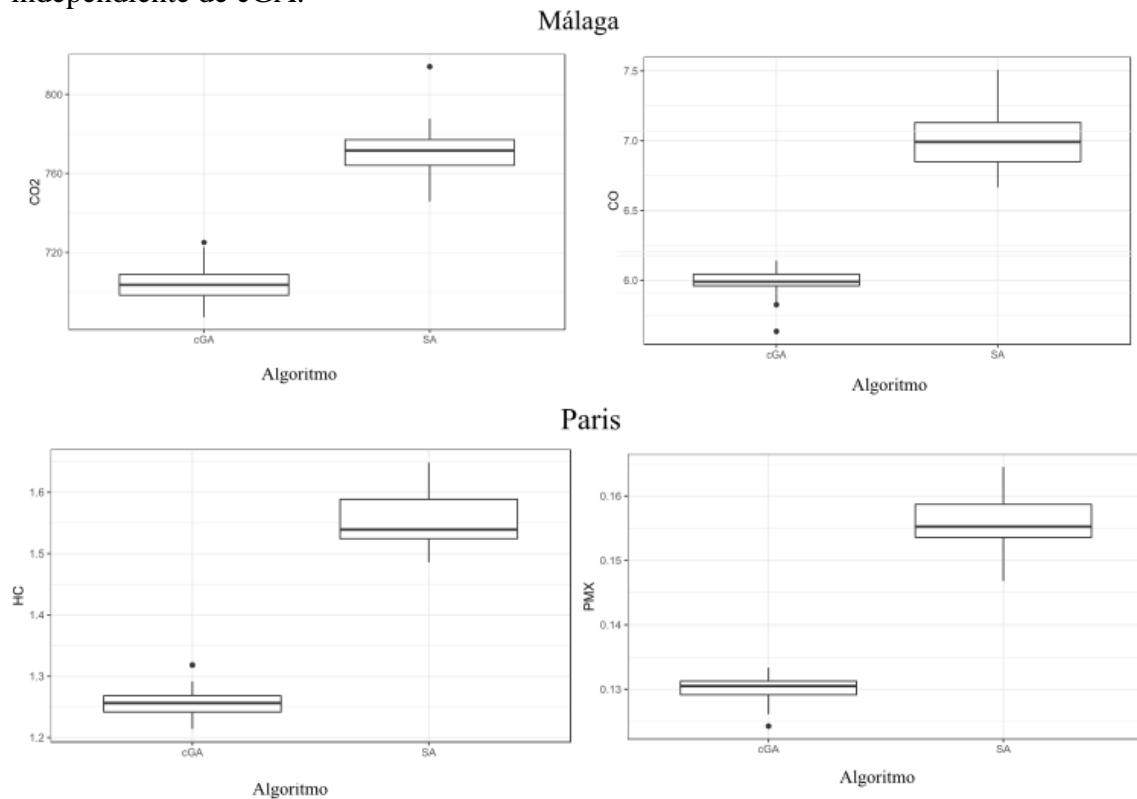


Figura 7: Box-Plot de emisión de gases de cGA y SA para instancias de Málaga y de París

Para seleccionar el número adecuado de *clusters* se aplicó *K-means* utilizando diferentes números de clústers (de 2 a 10) y para cada comprobar la cantidad de *clusters* adecuada se utilizaron además los métodos de *Silhouette* y *Elbow*. El número de *clusters* óptimo fue de dos.

La Tabla3 resume la cantidad de clusters optimo y la cantidad de elementos en cada uno de los *clusters* (Cluster1 y Cluster2). Podemos observar que para ambas instancias se obtiene un *cluster* (Cluster1) con muchos más elementos que los que contiene el segundo *cluster* (Cluster2). La distribución de los elementos en los *clusters* no es homogénea y casi el 80% de las soluciones pertenecen al primer *cluster*.

Result-1000	N° clusters	Elem-Cluster1	Elem-Cluster2
Málaga	2	828	172
París	2	799	201

Tabla 3: Clusters y elementos para Result-1000 en Málaga y París

En la Figura 8 se muestra el gráfico de coordenadas paralelas de los dos clusters para la instancia de Málaga (izquierda) y la instancia de París (derecha). Podemos observar que el

Cluster1 presenta valores (normalizados) pequeños y negativos para los centroides de los gases emitidos mientras que para el Cluster2 los valores son positivos y mayores.

En la Tabla 4 se muestra el promedio de emisiones para los clusters identificados. Se puede notar que el primer cluster (Cluster1) en ambas instancias presenta una reducción estadísticamente significativa en todos los gases efecto invernadero. Las diferencias pueden parecer menores, pero se debe tener en cuenta que son medidas en Kilogramos por segundos. Esto representa, por ejemplo, que las soluciones en el Cluster1 para París permiten reducir (en promedio) 4320 toneladas de CO₂ por día con respecto a las soluciones en el Cluster2. Para dar una referencia 4000 toneladas de CO₂ equivalen a las emisiones eléctricas usadas en 700 hogares durante un año de acuerdo a la Agencia de Protección Ambiental de Estados Unidos⁴. Estos resultados indican que, aunque la función de fitness (Ecuación 2) utilizada no incluye ningún aspecto de los gases emitidos, también es apropiada para obtener una reducción en los mismos.

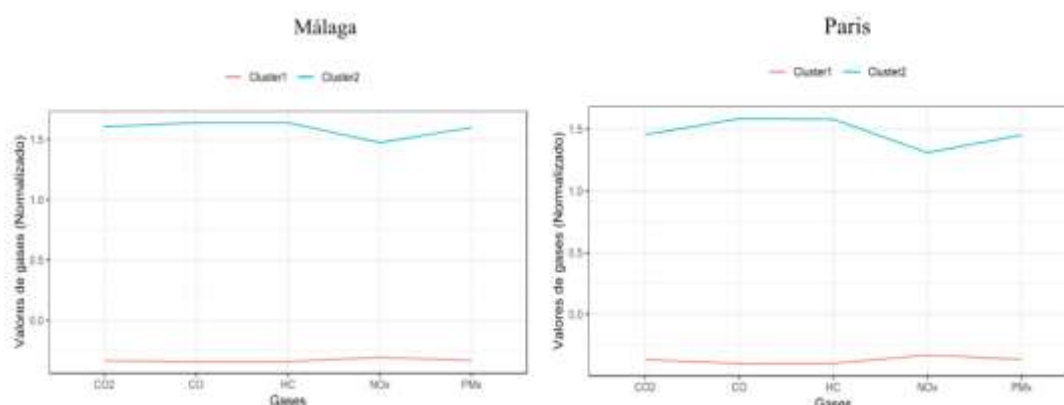


Figura 8: Clusters y centroides en cuanto a gases emitidos para instancias de Málaga y de París

En la Tabla 5 se muestra la correlación entre el fitness en Result-1000 y los gases emitidos. Se puede observar que la correlación del fitness con respecto a dos gases (CO y HC) es fuerte, y es significativa con respecto a PM_x (especialmente para la instancia de París), es moderado con CO₂ y débil con NO_x.

Instancia	Cluster	CO ₂	CO	HC	NO _x	PM _x
Málaga	Cluster1	719	6,19	0,97	4,67	0,10
	Cluster2	758	6,82	1,06	4,93	0,11
Paris	Cluster1	926	8,58	1,31	6,05	0,13
	Cluster2	976	9,74	1,47	6,38	0,15

Tabla 4: Promedio de gases emitidos (Kg/s) para cada cluster en Result-1000 en las instancias de Málaga y París

Instancia	CO ₂	CO	HC	NO _x	PM _x
Málaga	0,312	0,669	0,625	0,229	0,430
Paris	0,638	0,863	0,851	0,497	0,719

Tabla 5: Correlación entre el fitness y los gases emitidos

⁴ <https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator>

A continuación, se analizará la correlación entre los gases emitidos y cada una de las variables que están incluidas en la función de fitness (Ecuación 2): duración total de los viajes (TT_V) y tiempo de paradas y espera (TT_{EP}). No se analizan NV_D y NV_{ND} porque tienen valores constantes ya que todos los vehículos llegan a destino en la simulación. Tampoco se muestra el radio de colores (P) porque no existe una relación lineal entre estos valores y los valores de los gases emitidos. En la Tabla 6 y en la Tabla 7 se muestra la correlación entre las emisiones de los gases y la duración total de los viajes (TT_V) y los tiempos totales de paradas y espera (TT_{EP}), respectivamente. Se puede observar que en todos los casos la correlación es positiva al igual que la obtenida con la función de fitness completa. Esto es interesante desde el punto de vista práctico (los encargados de administrar la planificación de los tiempos de los semáforos) ya que esta relación provee un mecanismo indirecto para ajustar la planificación y reducir estos factores, para reducir las emisiones (ya que mecanismos directos como limitar el número de vehículos provoca disconformidad en los ciudadanos).

Instancia	CO ₂	CO	HC	NO _x	PM _x
Málaga	0,644	0,897	0,863	0,492	0,706
Paris	0,781	0,962	0,953	0,619	0,825

Tabla 6: Correlación entre la duración (TT_V) y los gases emitidos

Instancia	CO ₂	CO	HC	NO _x	PM _x
Málaga	0,526	0,826	0,783	0,396	0,621
Paris	0,755	0,953	0,939	0,585	0,800

Tabla 7: Correlación entre paradas (TT_{EP}) y los gases emitidos

8. CONCLUSIONES

Este trabajo se ha centrado en el problema de la planificación de las luces de tráfico, un tema importante en el marco de una ciudad inteligente. El primer objetivo fue utilizar para resolver el problema dos metaheurísticas: una poblacional (cGA) y otra de trayectoria (SA) aplicadas a dos escenarios cercanos a la realidad (instancias de Málaga y de París). El segundo objetivo fue realizar un análisis en profundidad y determinar la calidad de las soluciones obtenidas por los algoritmos en cuanto a la reducción de los gases emitidos. En cuanto a la calidad de las soluciones obtenidas por los algoritmos se obtuvo una excelente performance de cGA con respecto a SA. Se puede decir que atendiendo a la función objetivo utilizada cGA obtuvo una reducción del 30% de su valor para la instancia de Málaga y una reducción del 37% para la instancia de París. Además, se lograron mejoras con respecto a otro algoritmo de la literatura (DE), reduciendo en el valor de función objetivo en un 10% y un 7%, con las instancias de Málaga y de París, respectivamente. En cuanto a los resultados de los gases emitidos, también se ha logrado una mayor reducción en la emisión de gases por parte de cGA con respecto a SA, con diferencias estadísticamente significativas en todos los análisis realizados. El porcentaje de reducción de gases emitidos es del 8% al 14% en la instancia de Málaga. Para la instancia de París la reducción es mayor y está entre el 10% al 20%.

Se ha mostrado que la función de fitness utilizada es adecuada no solamente para el tráfico sino también para reducir las emisiones de gases. Desde el punto de vista de los que deben administrar el tráfico, se provee una herramienta adicional para reducir las emisiones de gases en una forma indirecta.

Trabajos futuros se centrarán en reducir el número de evaluaciones con el objetivo de reducir el esfuerzo computacional en términos de tiempo de procesamiento. Además, se realizará el estudio de diferentes funciones de fitness que involucren en su diseño a los gases analizados, con el fin de obtener una mayor reducción en las emisiones.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto 29/B225 “Soluciones inteligentes para el desarrollo urbano sostenible”. Los autores agradecen a la Universidad Nacional de la Patagonia Austral.

REFERENCIAS

- BASKAN, O., y HALDENBILEN, S. (2011). Ant colony optimization approach for optimizing traffic signal timings. In Ant colony optimization-methods and applications. IntechOpen. <https://doi.org/10.5772/13665>
- BASKAN, O., HALDENBILEN, S., CEYLAN, H., y CEYLAN, H. (2009). A new solution algorithm for improving performance of ant colony optimization. Applied Mathematics and Computation, 211(1), 75-84. <https://doi.org/10.1016/j.amc.2009.01.025>
- DORIGO, M. (1992). Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano.
- DORRONSORO, B., Y ALBA, E. (2008). Cellular Genetic Algorithms (Vol. 42). Springer-VerlGA US.
- EBERHART, R., Y KENNEDY, J. (1995). A new optimizer using particle swarm theory. In MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science (pp. 39-43). Ieee. <https://doi.org/10.1109/MHS.1995.494215>
- GARCÍA-NIETO, J., ALBA, E., y OLIVERA, A. C. (2012). Swarm intelligence for traffic light scheduling: Application to real urban areas. Engineering Applications of Artificial Intelligence, 25(2), 274-283. <https://doi.org/10.1016/j.engappai.2011.04.011>
- GARCIA-NIETO, J., OLIVERA, A. C., y ALBA, E. (2013). Optimal cycle program of traffic lights with particle swarm optimization. <https://doi.org/10.1109/TEVC.2013.2260755>
- HE, J., & HOU, Z. (2012). Ant colony algorithm for traffic signal timing optimization. Advances in Engineering Software, 43(1), 14-18. <https://doi.org/10.1016/j.advensoft.2011.09.002>
- HAUSBERGER, S., REXEIS, M., ZALLINGER, M., y LUZ, R. (2009). Emission Factors from the Model PHEM for the HBEFA Version 3. Report Nr. I-20/2009 Haus-Em 33/08/679 from 07.12.
- JABBARPOUR, M. R., MALAKOOTI, H., NOOR, R. M., ANUAR, N. B., y KHAMIS, N. (2014). Ant colony optimisation for vehicle traffic systems: applications and challenges. International Journal of Bio-Inspired Computation, 6(1), 32-56. <https://doi.org/10.1504/IJBIC.2014.059970>
- KIRKPATRICK, S., GELATT, C. D., y VECCHI, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680. <https://doi.org/10.1126/science.220.4598.671>
- KRAJZEWICZ, D., BONERT, M., & WGANER, P. (2006). The open source traffic simulation packGae SUMO. RoboCup 2006 Infrastructure Simulation Competition, 1, 1-5.

- KUN, C. H. E. N., y LEI, Y. U. (2007). Microscopic traffic-emission simulation and case study for evaluation of traffic control strategies. *Journal of Transportation Systems Engineering and Information Technology*, 7(1), 93-99. [https://doi.org/10.1016/S1570-6672\(07\)60011-7](https://doi.org/10.1016/S1570-6672(07)60011-7)
- METROPOLIS, N., ROSENBLUTH, A. W., ROSENBLUTH, M. N., TELLER, A. H., y TELLER, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6), 1087-1092. <https://doi.org/10.1063/1.1699114>
- NESMACHNOW, S., MASSOBRIO, R., ARRECHE, E., MUMFORD, C., OLIVERA, A. C., VIDAL, P. J., y TCHERNYKH, A. (2019). Traffic lights synchronization for Bus Rapid Transit using a parallel evolutionary algorithm. *International Journal of Transportation Science and Technology*, 8(1), 53-67. <https://doi.org/10.1016/j.ijtst.2018.10.002>
- PÉRES, M., RUIZ, G., NESMACHNOW, S., & OLIVERA, A. C. (2018). Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay. *Applied Soft Computing*, 70, 472-485. <https://doi.org/10.1016/j.asoc.2018.05.044>
- SÁNCHEZ, J. J., GALÁN, M., y RUBIO, E. (2004). Genetic algorithms and cellular automata: A new architecture for traffic light cycles optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No. 04TH8753) (Vol. 2, pp. 1668-1674)*. IEEE. <https://doi.org/10.1109/CEC.2004.1331096>
- SÁNCHEZ, J., GALÁN, M., y RUBIO, E. (2008). Applying a traffic lights evolutionary optimization technique to a real case: “Las Ramblas” area in santa cruz de tenerife. *IEEE Transactions on evolutionary computation*, 12(1), 25-40. <https://doi.org/10.1109/TEVC.2007.892765>
- SOURAVLIAS, D., LUQUE, G., ALBA, E., y PARSOPOULOS, K. E. (2016). Smart traffic lights: A first parallel computing approach. In *2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS) (pp. 229-236)*. IEEE. <https://doi.org/10.1109/INCoS.2016.72>
- STORN, R., & PRICE, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4), 341-359. <https://doi.org/10.1023/A:1008202821328>
- THAIN, D., TANNENBAUM, T., Y LIVNY, M. (2005). Distributed computing in practice: the Condor experience. *Concurrency and computation: practice and experience*, 17(2-4), 323-356. <https://doi.org/10.1002/cpe.938>
- TEKLU, F., SUMALEE, A., & WATLING, D. (2007). A genetic algorithm approach for optimizing traffic control signals considering routing. *Computer-Aided Civil and Infrastructure Engineering*, 22(1), 31-43. <https://doi.org/10.1111/j.1467-8667.2006.00468.x>
- VILLGARA, A., ALBA, E., Y LEGUIZAMÓN, G. (2016). A methodology for the hybridization based in active components: the case of cGA and scatter search. *Computational intelligence and neuroscience*, 2016. <https://doi.org/10.1155/2016/8289237>
- WHITLEY D. (1993). Cellular genetic algorithms, in: S. Forrest (Ed.), In *Proceedings of the Fifth International Conference on GAS*, Morgan Kaufmann, San Mateo, CA, 1993.
- YUAN, S., ZHAO, X., & AN, Y. (2014). Identification and optimization of traffic bottleneck with signal timing. *Journal of Traffic and Transportation Engineering (English Edition)*, 1(5), 353-361. [https://doi.org/10.1016/S2095-7564\(15\)30281-6](https://doi.org/10.1016/S2095-7564(15)30281-6)