

A Multi-Objective Swarm Approach with Pedestrians Spotlight in Traffic Urban Optimisation

Ana Carolina Olivera  and Pablo Javier Vidal 

Abstract—The way that people moves is changing. From a sustainability point of view is necessary put the focus on pedestrians. To reduce pollution and congestion in urban areas, it is necessary moves people with not necessary moves vehicles. This work introduces a particle swarm multi-objective approach that optimizes vehicles and pedestrians' traffic urban flow, considering the traffic lights timing. Traffic lights and their scheduling significantly impact vehicles and pedestrian flow in metropolitan cities. From the point of view of the scenario, a large-scale congested urban area is used to test the proposed methodology. The strategy is compared with five state-of-the-art algorithms with satisfactory results.

Index Terms—Swarm Intelligence, Multi-objective Optimization, Pedestrians, Traffic Flow

I. INTRODUCCIÓN

El interés por mejorar la experiencia de los peatones dentro de las ciudades forma parte de la necesidad de crear una ciudad sostenible capaz de priorizar el movimiento de las personas más que el de los vehículos que transitan por ella. Este cambio de paradigma implica no solo una mejora en el movimiento de las personas dentro de la red de tráfico, sino también su seguridad. En los países en desarrollo, los cruces peatonales y las pasarelas tienen menos prioridad que los vehículos. Además, los proyectos de renovación urbana no necesariamente dan más importancia a los peatones. Poner el foco en la implementación de políticas que consideren el nivel de servicio (LOS, *Level of Service*) de los peatones es crucial en las próximas décadas [1]–[3]. Existen dos objetivos claros que una planificación de la red de tráfico debe alcanzar, mantener el flujo de vehículos de tráfico y garantizar el nivel de servicio de los peatones sobre la red. Una forma en que los tomadores de decisiones pueden hacer esto es mejorando la programación de los ciclos de los semáforos en las intersecciones señalizadas [4], [5]. La optimización de la programación de los ciclos de los semáforos utilizando un método de control de tráfico inteligente puede mejorar el flujo de tráfico urbano [6]. Mejorar la experiencia de los peatones puede influir en la movilidad al reducir el uso de vehículos en las ciudades. Existen algunos trabajos donde se considera a los peatones dentro de la optimización de la programación de semáforos utilizando inteligencia computacional [1], [6], [7].

Ana Carolina Olivera, Pablo Javier Vidal. Instituto para las Tecnologías de la Información y las Comunicaciones, Consejo Nacional de Investigaciones Científicas y Técnicas, Universidad Nacional de Cuyo, Padre Jorge Contreras 1300, M5502JMA, Mendoza, Argentina.

Ana Carolina Olivera, Pablo Javier Vidal. Facultad de Ingeniería, Universidad Nacional de Cuyo, Centro Universitario, M5502JMA, Mendoza, Argentina
e-mails: acolivera@conicet.gov.ar, pjvidal@conicet.gov.ar

En este trabajo, se propone un enfoque de enjambre multi-objetivo para mejorar el flujo de vehículos y peatones en una área congestionada de una ciudad.

II. TRABAJOS RELACIONADOS

Si bien la literatura sobre inteligencia computacional para resolver la programación de semáforos es extensa, existen pocos trabajos orientados al nivel de servicio brindado a los peatones [1], [6]–[8].

Yu [8] proponen un método que usa programación cuadrática convexa en el que la programación de una intersección aislada con semáforos fijos se optimiza para mejorar los retrasos tanto de vehículos como de peatones.

En Li [1] los autores propusieron una programación de semáforos de dos niveles que considera los retrasos de los peatones que cruzan las intersecciones. El objetivo es minimizar la demora de los peatones manteniendo un buen funcionamiento de la red de tráfico. VisSim [9] es un simulador de tráfico microscópico y se utiliza para la evaluación de la posible solución.

Gao [7] propusieron la aproximación de semáforos bi-objetivos con impacto tanto en el retraso de vehículos como de peatones. Los autores utilizan la búsqueda de armonía [10] y la colonia de abejas artificiales [11] para resolver el problema proponiendo un operador de búsqueda local para mejorar la búsqueda de los algoritmos. Los resultados se comparan con el *Non-dominated Sorting Genetic Algorithm II* (NSGA-II). Las intersecciones señalizadas analizadas son de dos tipos: intersecciones de un solo sentido con dos fases e intersecciones simples con solo cuatro fases.

Li [6] propusieron una herramienta de ancho de banda de progresión para coordinar los semáforos teniendo en cuenta a los peatones. Vissim se utiliza para la simulación de la propuesta de programación de semáforos y el flujo de tráfico (peatones y vehículos). El principal inconveniente es que en el modelo solo se consideran dos intersecciones adyacentes.

Finalmente, Bai [12] exploraron la capacidad de optimización dinámica del tiempo de los semáforos considerando el retraso de los peatones y el costo del tiempo y el consumo de combustible de los vehículos. La estrategia se aplicó en una sola intersección en Beijing, China.

Los enfoques de *swarm intelligence* se utilizan para una variedad de problemas de optimización [13]–[16]. En particular, existen varios trabajos que utilizan enfoques de enjambre para la programación de los ciclos de los semáforos. Olivera [17] proponen un algoritmo mono-objetivo de enjambre que considera el consumo de combustible, los tiempos de demora

y la contaminación generada por los vehículos. El principal inconveniente es que no se considera el flujo de peatones. Similar ocurre con [18], [19]. Los autores se esfuerzan por mejorar el flujo de tráfico de vehículos sin tener en cuenta el impacto de la programación de los ciclos de los semáforos sobre el flujo de peatones. En Jia [20], los autores proponen un modelo de optimización de sincronización de semáforos multi-objetivo basado en la optimización de enjambres de partículas con objetivos que consideran el tiempo perdido, emisiones de vehículos y capacidad de intersección. Una vez más, los peatones no se consideran en el modelo. Además, el diseño de la intersección de la instancia no contempla a los peatones en la intersección (solo se optimiza una intersección) y solo el flujo de tráfico vehicular.

En resumen, todos estos trabajos, aunque algunos toman la incidencia de peatones y vehículos en el flujo vehicular, se limitan al considerar intersección aislada sin una mejora real del flujo vehicular y peatonal global de una ciudad o gran área urbana. Por esta razón, los resultados obtenidos por los diferentes enfoques son difícilmente aplicables en la realidad. Más aún, aquellos que toman áreas de mayor tamaño y utilizan *swarm intelligence* no tienen en cuenta a los peatones. En este trabajo, se propone un método que utiliza el algoritmo de enjambre de partículas multi-objetivo para optimizar simultáneamente 157 intersecciones señalizadas de un área urbana real con características de congestión para mejorar el flujo de tráfico de peatones y vehículos en la red de tráfico.

III. DESCRIPCIÓN DEL PROBLEMA

Dada una red vial que incluya intersecciones (señalizadas o no), calles, semáforos y flujo de tránsito tanto de vehículos como de peatones, los semáforos en las intersecciones semaforizadas deben programarse para coordinar el tránsito en la vía. Si en la red de tráfico se tienen I intersecciones semaforizadas, cada intersección señalizada $i \in I$ pasa por fases válidas f_{ij} dentro de su ciclo. Por lo tanto, f_{ij} representa la j -ésima fase que atraviesa la intersección i . Para señalar la fase que inicia el programa del ciclo, se configura un desfase u *offset* (en segundos). Entonces, para cada intersección i existe un desplazamiento o_i . Cada fase f_{ij} permanece activa en la intersección i durante varios segundos ($d_{f_{ij}}$) y luego cambia a la siguiente fase. El ciclo se repite perpetuamente.

El problema de la programación de los ciclos de los semáforos es encontrar las duraciones $d_{f_{ij}}$ para las fases (variables de decisión) de cada intersección i que minimice el tiempo de espera tanto de vehículos como de peatones considerando el flujo vehicular y peatonal. El modelo propuesto para la programación de los ciclos de los semáforos contempla el impacto en el flujo vehicular y peatonal en un periodo de tiempo (simulación en segundos). En (1) y (2) se definen los dos objetivos del análisis. Los objetivos contemplan las demoras vehiculares y peatonales en toda la red de tráfico en un periodo de tiempo (en segundos). En el caso de los vehículos este tiempo de demora contempla el tiempo promedio perdido por conducir más lento de lo deseado. La velocidad deseada tiene en cuenta el factor de velocidad del

vehículo (el tipo de vehículo, moto, auto, camión, etc.) según el *Handbook Emission Factors for Road Transport* (HBEFA) [21]. Para el caso de los peatones es el tiempo promedio de tiempo perdido por caminar por debajo de la velocidad máxima y el tiempo perdido promedio al detenerse en una esquina [22].

$$R_V = \frac{\sum_{v=0}^{\#V} v_r}{\#V} \quad (1)$$

$$R_P = \frac{\sum_{p=0}^{\#P} p_r}{\#P} \quad (2)$$

donde R_V es el retraso promedio de todos los vehículos en la red vial durante el tiempo de simulación. v_r es el tiempo de retraso del vehículo v , $\#V$ es el número total de vehículos durante la simulación. R_P denota el retraso promedio de todos los peatones durante la simulación, y p_r es el retraso o tiempo de demora de un peatón p . $\#P$ indica el número total de peatones durante el tiempo de simulación.

Es importante señalar que las fases están predefinidas en la topología de la red vial por lo que se optimiza el tiempo en segundos de cada fase para cada cruce considerando el tráfico en las vías. Además, las condiciones climáticas no son consideradas. De esta manera una solución al problema de la programación de los ciclos de los semáforos para una red de tráfico puede representarse como un vector de enteros (segundos),

$$[o_1, d_{f_{11}}, \dots, d_{f_{1k_1}}, o_2, d_{f_{21}}, \dots, d_{f_{2k_2}}, \dots, o_n, d_{f_{n1}}, \dots, d_{f_{nk_n}}]$$

donde o_1, o_2, \dots, o_n son los *offset* de cada intersección $i = 1, 2, \dots, n$ siendo $n = |I|$ el número de intersecciones semaforizadas y $d_{f_{ik_1}}, d_{f_{ik_2}}, \dots, d_{f_{ik_i}}$ son las duraciones asignadas a cada fase de la intersección i .

IV. ALGORITMO MULTI-OBJETIVO DE ENJAMBRE

En este trabajo se propone una estrategia de enjambre multi-objetivo que considera el tráfico de todos los agentes de la red vial, vehículos (motorizados o no) y peatones. La técnica propuesta se basa en el *Multi-Objective Particle Swarm Optimizer* (OMOPSO) diseñado por Sierra [23] con dos diferencias principales: i) los alelos cromosómicos utilizan una representación flotante que se redondea cuando se evalúa la solución, y ii) los operadores de enjambre se adaptan al problema de programación de los ciclos de los semáforos.

El optimizador de enjambre de partículas multi-objetivo considera dos tipos de dominancia: la dominancia de Pareto y la dominancia ϵ . La dominancia de Pareto utiliza un factor de hacinamiento o *crowding* [24] para evitar el elitismo mientras que el concepto de dominancia ϵ se utiliza para establecer qué soluciones (líderes) se mantendrán durante las generaciones cuando el tamaño de líderes se sobrepasa.

Después de cada generación, el conjunto de líderes se actualiza, así como los valores de *crowding* correspondientes. Si el tamaño del conjunto de líderes es mayor que el tamaño máximo permitido definido a través de ϵ , sólo se conservan los mejores líderes en función de su valor de *crowding*. El resto de los líderes son eliminados.

OMOPSO utiliza dos operadores de mutación, mutación uniforme (es decir, el rango de variabilidad permitido para cada variable de decisión se mantiene constante durante generaciones) y mutación no uniforme (es decir, el rango de variabilidad permitido para cada variable de decisión disminuye con el tiempo). Estos operadores modifican los valores de las variables de decisión de una partícula con cierta probabilidad. Adicionalmente, existe la posibilidad de no aplicar mutación alguna. Teniendo en cuenta el tipo de mutación, el enjambre se divide en tres partes (de igual tamaño). Cada subconjunto adopta un esquema de mutación diferente. Esto permite la explotación (mutación no uniforme) y la exploración (mutación uniforme) del espacio de búsqueda a lo largo de las iteraciones. La tasa de mutación se define como $1/codesize$ donde $codesize$ se refiere a la longitud total del vector que codifica todas las variables de decisión del problema de programación de los ciclos de los semáforos (el número total de fases de todas las intersecciones señalizadas).

Por último, OMOPSO adopta el concepto de ϵ -dominance [25] para fijar el tamaño del archivo externo que contiene las soluciones (no dominadas) que informará el algoritmo al final de su ejecución. Se dice que un vector de decisión x_1 domina ϵ a un vector de decisión x_2 para algún $\epsilon > 0$ si y solo si $R_i(x_1)/(1 + \epsilon) \leq R_i(x_2)$, $i = \{V, P\}$ y $R_i(x_1)/(1 + \epsilon) < R_i(x_2)$, para R_V o R_P . Para el problema de optimización de programación de semáforos, el valor de ϵ se establece en 0.0075 considerando trabajos anteriores que utilizan OMOPSO [23].

Considerando el Algoritmo 1, OMOPSO comienza inicializando el enjambre y calculando los líderes (líneas 1-2). Las partículas no dominadas (los líderes) se envían al archivo ϵ (línea 3). Se calcula el factor de hacinamiento de los líderes (línea 4). En cada generación (línea 6) y para cada partícula (línea 7), el líder se selecciona mediante un torneo binario basado en el valor de hacinamiento de los líderes (línea 8). En cada iteración, se aplica cada vuelo de partículas (línea 9) y el operador de mutación (línea 10). El vuelo de las partículas se calcula en base a la subdivisión del enjambre. La actualización de la velocidad de una partícula i se calcula mediante $v_i(t) = W \times v_i(t-1) + C_1 \times a_1(x_{pbest_i} - x_i(t)) + c_2 \times a_2(x_{gbest} - x_i(t))$, donde W es un número aleatorio entre 0.1 y 0.5, C_1 y C_2 son números aleatorios entre 1.5 y 2, y a_1 y a_2 son números aleatorios entre 0.0 y 1.0. Una vez que se evalúa la partícula (línea 11), se actualiza su valor p_{best} (línea 12). Una nueva partícula reemplaza su valor p_{best} si dicho valor está dominado por la nueva partícula o si ambos no están dominados entre sí. Una vez que se han actualizado todas las partículas, también se actualiza el conjunto de líderes (línea 14). Solo las partículas que superan su valor p_{best} intentarán ingresar al conjunto de líderes (según su valor de hacinamiento). Una vez que se ha actualizado el conjunto de líderes, se actualiza el archivo ϵ (línea 15). Las iteraciones se repiten hasta alcanzar el límite g_{max} .

Para el tratamiento de la programación de los ciclos de los semáforos, es necesario considerar que el tiempo de las fases de una intersección de semáforos está en segundos (es decir, números enteros). De esta forma, una partícula i se codifica como vector de números flotantes que cuando es

Algoritmo 1 Pseudocódigo de OMOPSO

```

1: Inicializa el enjambre
2: Inicializa los líderes
3: Envía los líderes al archivo  $\epsilon$ 
4: hacinamiento(líderes)
5:  $g = 0$ 
6: while  $g < g_{max}$  do
7:   for each partícula  $i$  do
8:     Seleccionar líder
9:     Volar( $i$ )
10:    Mutar( $i$ )
11:    Evaluar( $i$ )
12:    Actualizar( $p_{best_i}$ )
13:  end for
14:  Actualizar líderes
15:  Envía líderes al archivo  $\epsilon$ 
16:  hacinamiento(líderes)
17:   $g++$ 
18: end while
19: Reporta los resultados del archivo  $\epsilon$ 

```

evaluada deben ser redondeados. Cuando el valor de la parte decimal de un alelo es inferior a 0.5 se descarta y cuando el número decimal es superior a 0.5 suma uno a la parte entera. Entonces, todos los alelos de las partículas se cambian con estas consideraciones. La mutación resta uno al alelo considerando los límites máximos y mínimos definidos para los valores de las fases de las intersecciones.

De esta forma, dado un vector solución $[o_1, d_{f_{11}}, \dots, d_{f_{1k_1}}, o_2, d_{f_{21}}, \dots, d_{f_{2k_2}}, \dots, o_n, d_{f_{n1}}, \dots, d_{f_{nk_n}}]$ que representa las duraciones de las fases en el problema de la programación de los ciclos de semáforos, se utiliza una representación real para codificar la duración de cada fase f_{ij} y el desplazamiento correspondiente (o_i). Es importante destacar que el algoritmo nunca produce soluciones no factibles puesto que las operaciones sobre los vectores solución no modifican las posiciones de las fases sino su valor (sus duraciones en segundos). De esta manera el algoritmo no necesita ninguna estrategia de recuperación para soluciones no factibles. Los estados asociados a las luces amarillas tienen una duración constante de 4 segundos (según la normativa de tráfico), por lo que no se incluyen en la representación. Los estados verde y rojo tienen una duración $f_{ij} \in [5, 50] \subset N$. Los desplazamientos u *offsets* se definen de tal manera que $o_i \in [0, 50k_i + 5k_i] \subset N$, siendo $50k_i + 5k_i$ la duración total del programa [26]. En la Fig. 1 se observa una intersección semaforizada con cruce peatonal vista desde el visualizador de SUMO. Debajo de la imagen se puede ver el fragmento del archivo .net de SUMO que define internamente la topología de la red en particular para esta intersección. A partir de este archivo se obtienen las fases válidas que atraviesan las intersecciones en particular la intersección observada. Luego, se arma la representación de la solución donde cada valor del vector representa el offset y las duraciones de las fases que atraviesan las intersecciones, en particular, en la figura se presentan los valores en un vector solución de la intersección.

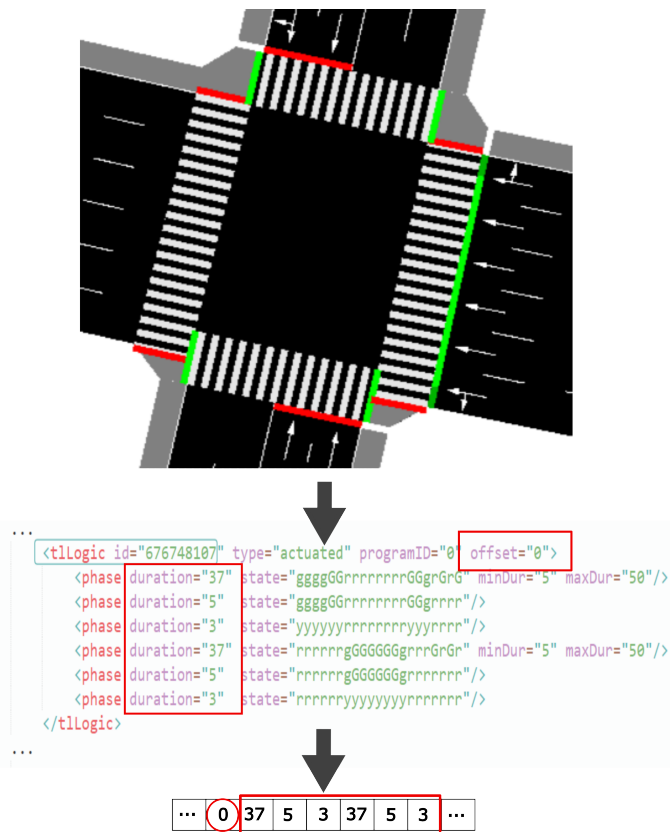


Fig. 1. Una intersección semaforizada con cruce peatonal.

V. CONFIGURACIÓN DE LOS EXPERIMENTOS

A continuación, se introduce el escenario al que se le optimizará la programación de los ciclos de los semáforos. Luego, se presentan los algoritmos utilizados a comparar con nuestra propuesta. Por último, se definen los indicadores multi-objetivo para evaluar el comportamiento de los algoritmos.

A. Escenario

En orden de demostrar el potencial de la nuestra propuesta se utiliza como escenario el centro y los alrededores de la ciudad de Mendoza en Argentina con 157 intersecciones señalizadas con topología heterogénea. El mínimo y máximo en segundos de cada fase de una intersección se establece en 4 segundos para fases amarillas y entre 5 a 50 segundos para las otras fases [5], [17]–[19], [27]. La simulación se establece en 900 segundos.

En la Fig. 2 se puede observar la distribución de las 157 intersecciones semaforizadas del escenario utilizado para el testeo de los algoritmos. Cada una de estas intersecciones tiene una estructura similar a la presentada en la Fig. 1, la cual es extraída del escenario y puede observarse con un recuadro azul la intersección de la Fig. 2. Cada intersección de la Fig. 2 (marcada con un ●) esta compuesta por una serie de fases con sus respectivos valores de duración y *offset*, los cuales son obtenidos como se muestra en la Fig. 1 para la construcción del vector solución teniendo en cuenta las 157 intersecciones.

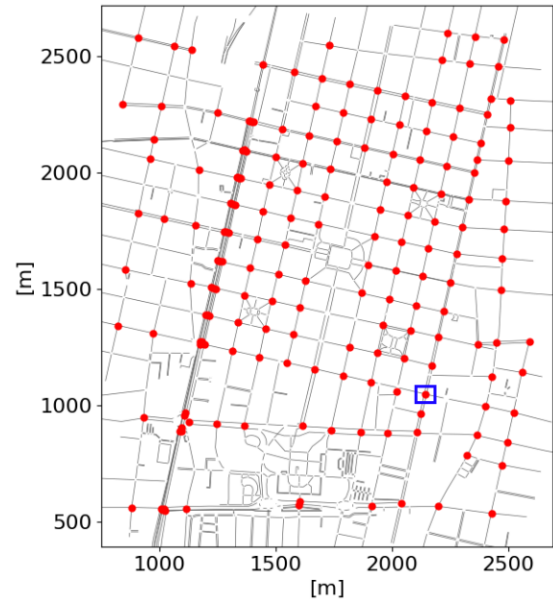


Fig. 2. Ubicación de las 157 intersecciones semaforizadas sobre el centro de la ciudad de Mendoza y sus alrededores.

B. Algoritmos

Cinco algoritmos multi-objetivo del estado del arte son utilizados para evaluar la técnica propuesta. NSGA-II (*Non-dominated Sorting Genetic Algorithm, version II*) [24], SPEA2 (*Strength Pareto Evolutionary Algorithm 2*) [28], MOEA/D (*Multi-objective Evolutionary Algorithm based on Decomposition*) [29], MOCeLL (*Cellular Genetic Algorithm for Multi-objective Optimization*) [30] y GDE3 (*Generalized Differential Evolution, version 3*) [31].

Non-dominated Sorting Genetic Algorithm, version II (NSGAII, por sus siglas en inglés) [24], es un algoritmo evolutivo multi-objetivo de última generación que se ha aplicado con éxito en áreas diversas [32]. El cálculo de la aptitud se basa en la dominancia de Pareto, construyendo *frentes* de soluciones. La búsqueda evolutiva en el NSGA-II mejora de la versión anterior (NSGA), utilizando: *i*) una clasificación elitista no dominada que reduce la complejidad de la verificación del dominio; *ii*) una técnica de hacinamiento (*crowding*) para la preservación de la diversidad; y *iii*) el valor de aptitud de una solución considera valores de distancia de hacinamiento.

Strength Pareto Evolutionary Algorithm 2 (SPEA2, por sus siglas en inglés) [28] es un algoritmo evolutivo multi-objetivo elitista popular en la literatura basado en el algoritmo evolutivo de dominancia Pareto. En SPEA2, la aptitud tomada para cada individuo tiene en cuenta la cantidad de individuos que domina y está dominada por el individuo. SPEA2 tiene una técnica de estimación de densidad vecina más cercana que permite una guía más precisa del proceso de búsqueda y un método de truncamiento de archivo que garantiza la preservación de soluciones en los límites de la frontera.

Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D, por sus siglas en inglés) descompone un problema de optimización multi-objetivo en una serie de subproblemas de optimización escalares y los optimiza

TABLA I
CONFIGURACIÓN UTILIZADA PARA CADA ALGORITMO

Algoritmo	Parámetro	Valor
OMOPSO	Tamaño del enjambre	100
	Mutación Uniforme	Bit-Flip con prob. ($1/codesize$)
	Mutación no Uniforme	0.5%
NSGA-II	Tamaño Población	100
	Cruzamiento	SPX con probabilidad 100%
	Mutación	Bit-Flip con prob. ($1/codesize$)
	Selección	Torneo Binario
SPEA2	Tamaño Población	100
	Cruzamiento	SPX con probabilidad 100%
	Mutación	Bit-Flip con prob. ($1/codesize$)
	Selección	Torneo Binario
MOCeII	Tamaño Población	100 (10×10)
	Vecindario	9 soluciones cercanas
	Cruzamiento	SPX con probabilidad 100%
	Mutación	Bit-Flip con prob. ($1/codesize$)
MOEA/D	Selección	Torneo Binario
	Tamaño Población	100 (10×10)
	Cruzamiento	prob. 0.1
	Factor de Escalado	f=0.5
GD3	Selección	DE/rand/1/bin
	Tamaño Población	100 (10×10)
	Cruzamiento	prob. 0.1
	Factor de Escalado	f=0.5
GD3	Selección	DE/rand/1/bin
	Tamaño Población	100 (10×10)
	Cruzamiento	prob. 0.1
	Factor de Escalado	f=0.5

simultáneamente. Cada uno está optimizado solo por usar información de sus subproblemas vecinos [29].

Cellular Genetic Algorithm for Multi-objective Optimization (MOCeII) [30] es un algoritmo genético y elitista basado en celular que utiliza un archivo externo para almacenar soluciones no dominadas y un mecanismo de retroalimentación en el que las soluciones de este archivo reemplazan al azar a las personas existentes en la población después de cada iteración.

Generalized Differential Evolution, version 3 (GDE3) [31] es un algoritmo que implementa la estrategia de evolución diferencial extendida para resolver problemas de optimización multi-objetivo. El operador de selección se basa en el dominio de Pareto, y la estrategia “DE/Rand/1/bin” [33] se aplica para resolver problemas con múltiples objetivos y restricciones.

Los operadores evolutivos comprenden *Simulated Binary Crossover* (SBX, Cruzamiento Binario Simulado) [34] el cual permite la emulación del operador tradicional de cruce de punto único (SPX, por sus siglas en inglés) para la codificación real y la mutación de Bit-Flip (cambio de bit) que en este trabajo es modificada para considerar qué tipo de fase se está mutando, por ejemplo, la fase que contiene amarillos no superará los 4 segundos. El torneo binario se utiliza como operador de selección. La Tabla I muestra la configuración utilizada para cada algoritmo. Para establecer estos valores se utilizaron pruebas empíricas no reportadas y valores utilizados en el estado del arte sobre programación de los ciclos de los semáforos [5], [27] y para algoritmos evolutivos multi-objetivo como los utilizados en el presente trabajo [35].

Para todos los algoritmos, el número máximo de evaluaciones se establece como 5000. Todos los experimentos se han realizado utilizando Toko Cluster [36] con un procesador

AMD Opteron / EPYC (64 núcleos y 128 GB de RAM). El sistema operativo es Ubuntu 18.04 LTS. La biblioteca Jmetalpy [37] [38] contiene todos los algoritmos, en sus versiones canónicas, utilizados para este trabajo. El simulador microscópico de tráfico utilizado para evaluar las soluciones fue SUMO [22] en su versión 1.4.0. La topología completa de la red junto con el flujo del tráfico se generó a través de la herramienta osmWebWizard.py [39] provista como parte de SUMO [22]. Se puede consultar la topología de la red mostrada en la Fig. 2 en http://github.com/acolivera/topologia_mendozaa.git.

C. Indicadores

Para analizar el comportamiento del algoritmo de optimización de enjambre de partículas multi-objetivo frente a las otras técnicas se utilizan métricas multi-objetivo de la literatura [40]. Los indicadores (métricas) utilizados en este trabajo comprenden: hipervolumen (HV, *Hipervolume*) [41], Propagación (Δ , *Spread*) [24], distancia generacional (GD, *Generational Distance*) [42], distancia generacional invertida (IGD, *Inverted Generational Distance*) [43], IGD mejorado (IDG+, *Improved Inverted Generational Distance*) [44] y el indicador ϵ (I_ϵ) [45]. GD mide la distancia entre el conjunto de soluciones evolucionadas y el verdadero frente Pareto; La métrica de HV considera tanto la cercanía como la diversidad al mismo tiempo; Δ mide el grado de propagación alcanzado entre las soluciones obtenidas; IGD mide el alcance de la convergencia a un conjunto conocido de soluciones del frente de Pareto y se utilizan tanto para el espacio de la solución como al espacio objetivo; IGD+ modifica el método de cálculo de distancia; I_ϵ Da la distancia mínima por la cual se debe traducir una aproximación de Pareto, se puede traducir o se puede traducir en cada dimensión en el espacio objetivo, de modo que otra aproximación se domina débilmente.

Se realiza una prueba de Kolmogorov-Smirnov para verificar si las muestras se distribuyen de acuerdo con una distribución normal o no para verificar si las diferencias entre los algoritmos son estadísticamente significativas o solo una cuestión de azar. Si es así, se realiza una prueba ANOVA I; De lo contrario, se realiza una prueba de Kruskal-Wallis. Dado que más de dos algoritmos están involucrados en el estudio, se ha realizado posterior a la prueba que una fase que permite que se realicen múltiples comparaciones de las muestras (múltiples comparativas). Todas las pruebas estadísticas se realizan con un nivel de confianza de 95% [46]–[48].

VI. ANÁLISIS Y RESULTADOS

En esta sección, se presenta el análisis de los resultados obtenidos por los algoritmos en comparación con OMOPSO.

La Tabla II ilustra los resultados de las métricas. Como se puede observar, OMOPSO obtuvo los mejores resultados para todas las métricas considerando el escenario presentado.

En particular, para I_ϵ OMOPSO obtiene casi tres veces en magnitud en comparación con el segundo mejor. Para Δ los resultados evidencian que las soluciones obtenidas por OMOPSO se encuentran en el Frente de Pareto. Para la programación de los ciclos de los semáforos, esto se traduce en

TABLA II
RESULTADOS OBTENIDOS POR LOS INDICADORES PARA LA MEDIA (M, MEAN) Y LA DESVIACIÓN ESTÁNDAR (SD, STANDARD DEVIATION) PARA CADA ALGORITMO

		MOCcell	NSGAI	SPEA2	MOEAD	OMOPSO	GDE3
I_{ϵ}	(M)	$7.85e-01$	$4.08e-01$	$4.19e-01$	$7.66e-01$	$1.20e-01$	$7.18e-01$
	(SD)	$7.0e-02$	$1.2e-01$	$1.1e-01$	$0.0e+00$	$0.0e+00$	$0.0e+00$
Δ	(M)	$8.14e-01$	$1.33e+00$	$8.83e-01$	$1.02e+00$	$6.91e-01$	$8.10e-01$
	(SD)	$1.5e-02$	$3.8e-02$	$8.8e-05$	$0.0e+00$	$0.0e+00$	$0.0e+00$
GD	(M)	$1.74e-01$	$4.51e-02$	$7.40e-0$	$5.62e-02$	$9.22e-03$	$2.52e-01$
	(SD)	$2.1e-02$	$1.3e-03$	$2.1e-02$	$0.0e+00$	$0.0e+00$	$0.0e+00$
HV	(M)	$7.34e-02$	$3.41e-01$	$4.07e-01$	$0.00e+00$	$7.04e-0$	$3.72e-03$
	(SD)	$2.0e-02$	$1.0e-01$	$1.3e-01$	$0.0e+00$	$0.0e+00$	$0.0e+00$
IGD	(M)	$1.43e-01$	$6.96e-02$	$6.73e-02$	$1.72e-01$	$1.66e-02$	$1.57e-01$
	(SD)	$2.3e-02$	$4.2e-04$	$2.6e-03$	$0.0e+00$	$0.0e+00$	$0.0e+00$
IGD+	(M)	$6.26e-01$	$3.03e-01$	$2.62e-01$	$8.36e-01$	$2.26e-02$	$7.69e-01$
	(SD)	$2.7e-02$	$3.5e-02$	$3.8e-02$	$0.0e+00$	$0.0e+00$	$0.0e+00$

diferentes opciones de programaciones que un tomador de decisiones puede considerar según diferentes políticas aplicadas para priorizar a los peatones o a los vehículos.

El segundo mejor fue logrado por NSGAI y SPEA2 en dos y tres métricas. GDE3 solo obtiene el segundo mejor con Δ . Estos resultados demuestran que OMOPSO obtiene soluciones que equilibran la exploración y la explotación del espacio de soluciones. Además, OMOPSO aporta generacionalmente a las soluciones del Frente de Pareto obtenido.

Con respecto a los tiempos perdidos de viaje tanto de los vehículos como de los peatones se analizaron los valores obtenidos por las programaciones de los ciclos de los semáforos obtenidas con OMOPSO y con NSGAI que fueron los que obtuvieron los mejores resultados con las métricas. En promedio, el tiempo de viaje de los vehículos utilizando las programaciones de los semáforos obtenidas con OMOPSO son 5% menores que con las programaciones de los ciclos de los semáforos construidas por el NSGAI. Esto se traduce en más tiempo de viaje por parte de los vehículos que se simularon sobre las programaciones obtenidas a través del NSGAI en contraposición con las programaciones obtenidas con OMOPSO. Con respecto a los peatones la diferencia a favor del OMOPSO alcanza el 7% en promedio. En resumen, para el centro y alrededores de Mendoza, OMOPSO consigue programaciones de los ciclos de los semáforos que permiten la fluidez tanto de los peatones como de los vehículos sobre la red de tráfico lo que significa que el peatón consigue una mayor facilidad para desplazarse en la ciudad aún en horas de alta intensidad de tráfico. Esto alienta a que las personas dejen sus vehículos alejados de los lugares típicos de congestión vehicular disminuyendo las emisiones vehiculares y el gasto de combustible promoviendo así el desplazamiento peatonal.

VII. CONCLUSIONES

En este trabajo se presentó una técnica que utiliza el algoritmo de optimización de enjambre de partículas multi-objetivo (OMOPSO) para la programación de los ciclos de los semáforos. El objetivo fue mejorar el flujo tanto de vehículos (motorizados o no) como de los peatones sobre una red de

tráfico considerando reducir sus demoras. Como caso de estudio se utilizó el centro y alrededores de la ciudad de Mendoza en Argentina que consta de 157 intersección semaforizadas. Se comparó la técnica con cinco algoritmos multi-objetivo populares en la literatura sobre optimización. OMOPSO obtuvo el mejor rendimiento en las métricas analizadas, por ejemplo para I_{ϵ} triplicó el valor con respecto al resultado alcanzado por el segundo mejor. En cuanto al flujo del tráfico OMOPSO consigue que los peatones puedan desplazarse sobre la red de tráfico incluso en áreas en áreas congestionadas por lo que la gente puede optar por este medio para desplazarse por el centro de la ciudad lo que se ve traducido en reducción del consumo de combustible y emisiones.

AGRADECIMIENTOS

Los fondos para realizar esta investigación provienen en parte de la Universidad Nacional de Cuyo (Proyecto 80020210100059UN) y en parte de la Agencia Nacional de Promoción de la Investigación, el Desarrollo Tecnológico y la Innovación (AGENCIA I+D+i) con el PICT-2020-SERIEA-00743, Argentina. Los autores también agradecen al Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina.

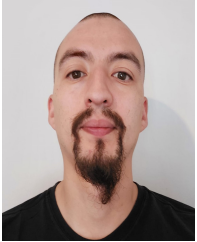
REFERENCES

- [1] K. Li, J. Xiang, X. Yu, and Y. Ni, *Analysis and Optimization of Pedestrian Traffic Signal at Intersections*, 2019, pp. 3183–3194.
- [2] A. Singh, J. Baalsrud Hauge, M. Wiktorsson, and U. Upadhyay, "Optimizing local and global objectives for sustainable mobility in urban areas," *Journal of Urban Mobility*, vol. 2, p. 100012, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2667091721000121>
- [3] G. Domenico, C. G. Carla, and M. Margherita, "Experimental models of pedestrian flows as support to design new sustainable paths in urban context," *Transportation Research Procedia*, vol. 60, no. 2021, pp. 188–195, 2022. [Online]. Available: <https://doi.org/10.1016/j.trpro.2021.12.025>
- [4] A. Olivera, J. García-Nieto, and E. Alba, "Reducing vehicle emissions and fuel consumption in the city by using particle swarm optimization," *Applied Intelligence*, vol. 42, no. 3, 2015.
- [5] P. Vidal and A. Olivera, "Management of urban traffic flow based on traffic lights scheduling optimization," *IEEE Latin America Transactions*, vol. 17, no. 1, 2019.

- [6] D. Li, Y. Song, and Q. Chen, "Bilevel Programming for Traffic Signal Coordinated Control considering Pedestrian Crossing," *Journal of Advanced Transportation*, vol. 2020, 2020.
- [7] K. Gao, Y. Zhang, Y. Zhang, R. Su, and P. N. Suganthan, "Meta-Heuristics for Bi-Objective Urban Traffic Light Scheduling Problems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2618–2629, 2019.
- [8] C. Yu, W. Ma, K. Han, and X. Yang, "Optimization of vehicle and pedestrian signals at isolated intersections," *Transportation Research Part B: Methodological*, vol. 98, no. January, pp. 135–153, 2017.
- [9] M. Fellendorf and P. Vortisch, *Microscopic Traffic Flow Simulator VISSIM*. New York, NY: Springer New York, 2010, pp. 63–93. [Online]. Available: https://doi.org/10.1007/978-1-4419-6142-6_2
- [10] Z. W. Geem, J. H. Kim, and G. Loganathan, "A new heuristic optimization algorithm: Harmony search," *SIMULATION*, vol. 76, no. 2, pp. 60–68, 2001.
- [11] D. Karaboga, B. Akay, and C. Ozturk, "Artificial bee colony (abc) optimization algorithm for training feed-forward neural networks," in *Modeling Decisions for Artificial Intelligence*, V. Torra, Y. Narukawa, and Y. Yoshida, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 318–329.
- [12] K. Bai, E. Yao, L. Pan, L. Li, and W. Chen, "Dynamic crosswalk signal timing optimization model considering vehicle and pedestrian delays and fuel consumption cost," *Sustainability (Switzerland)*, vol. 12, no. 2, 2020.
- [13] K. Jha and S. Saha, "Incorporation of multimodal multiobjective optimization in designing a filter based feature selection technique," *Applied Soft Computing*, vol. 98, p. 106823, 2021.
- [14] S. Kanwal, I. Younas, and M. Bashir, "Evolving convolutional autoencoders using multi-objective Particle Swarm Optimization," *Computers & Electrical Engineering*, vol. 91, p. 107108, 2021.
- [15] P. Ocloń, M. Rerak, R. V. Rao, P. Cisek, A. Vallati, D. Jakubek, and B. Rozegnał, "Multiobjective optimization of underground power cable systems," *Energy*, vol. 215, p. 119089, 2021.
- [16] M. Sedighkia, B. Datta, and A. Abdoli, "Minimizing physical habitat impacts at downstream of diversion dams by a multiobjective optimization of environmental flow regime," *Environmental Modelling & Software*, vol. 140, p. 105029, 2021.
- [17] A. C. Olivera, J. M. García-Nieto, and E. Alba, "Reducing vehicle emissions and fuel consumption in the city by using particle swarm optimization," *Applied Intelligence*, vol. 42, no. 3, pp. 389–405, 2015. [Online]. Available: <http://dx.doi.org/10.1007/s10489-014-0604-3>
- [18] J. García-Nieto, E. Alba, and A. Carolina Olivera, "Swarm intelligence for traffic light scheduling: Application to real urban areas," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, 2012.
- [19] J. García-Nieto, A. Olivera, and E. Alba, "Optimal cycle program of traffic lights with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 6, 2013.
- [20] H. Jia, Y. Lin, Q. Luo, Y. Li, and H. Miao, "Multi-objective optimization of urban road intersection signal timing based on particle swarm optimization algorithm," *Advances in Mechanical Engineering*, vol. 11, no. 4, pp. 1–9, 2019.
- [21] B. Graz, H. Lyon, B. Notter, B. C. Infrás, S. Hausberger, C. Matzer, K. Weller, M. Dippold, and N. Po, "HBEFA 4.2 Documentation of updates," Tech. Rep. Iv1, 2022.
- [22] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, "Microscopic traffic simulation using sumo," in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, November 2018, pp. 2575–2582. [Online]. Available: <https://elib.dlr.de/127994/>
- [23] M. R. Sierra and C. A. Coello Coello, "Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance," in *Evolutionary Multi-Criterion Optimization*, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 505–519.
- [24] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [25] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multiobjective Optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, sep 2002. [Online]. Available: <https://direct.mit.edu/evco/article/10/3/263-282/1129>
- [26] S. T. Manual, *REPORT 812 Signal Timing Manual*, 2015.
- [27] M. Péres, G. Ruiz, S. Nesmachnow, and A. Olivera, "Multiobjective evolutionary optimization of traffic flow and pollution in Montevideo, Uruguay," *Applied Soft Computing Journal*, vol. 70, 2018.
- [28] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, Tech. Rep. 103, 2001.
- [29] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
- [30] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "Design issues in a multiobjective cellular genetic algorithm," in *Evolutionary Multi-Criterion Optimization. 4th International Conference, EMO 2007*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds., vol. 4403. Springer, 2007, pp. 126–140.
- [31] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *Proceedings of the Congress on Evolutionary Computation*, 2006, pp. 1179–1186.
- [32] E. Baquela and A. Olivera, "A novel hybrid multi-objective memodel-based evolutionary optimization algorithm," *Operations Research Perspectives*, vol. 6, 2019.
- [33] R. Storn and K. Price, "Differential Evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, Tech. Rep. 11, 1995.
- [34] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 3, pp. 1–15, 1994.
- [35] K. Sekine and T. Tatsukawa, "A Parametric Study of Crossover Operators in Multi-objective Evolutionary Algorithm," *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, pp. 1196–1203, 2019.
- [36] <https://toko.uncu.edu.ar/>.
- [37] A. Benítez-Hidalgo, A. J. Nebro, J. García-Nieto, I. Oregi, and J. D. Ser, "jmetalpy: A python framework for multi-objective optimization with metaheuristics," *Swarm and Evolutionary Computation*, p. 100598, 2019.
- [38] <https://github.com/jMetal/jMetalPy>.
- [39] <http://sumo.dlr.de/docs/Tools/Import/OSM.html#osmwebwizardpy>.
- [40] Y. Yang, J. Luo, L. Huang, and Q. Liu, "A many-objective evolutionary algorithm with epsilon-indicator direction vector," *Applied Soft Computing Journal*, vol. 76, pp. 326–355, 2019. [Online]. Available: <https://doi.org/10.1016/j.asoc.2018.11.041>
- [41] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms — a comparative case study," in *Parallel Problem Solving from Nature — PPSN V*, A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 292–301.
- [42] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Ph.D. dissertation, Wright Patterson AFB, OH, USA, 1999, aA19928483.
- [43] P. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 174–188, 2003.
- [44] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," 2015, pp. 170–177.
- [45] E. Zitzler, L. Thiele, M. Laumanns, C. Fonseca, and V. D. Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 117–132, 2003.
- [46] T. Bartz-Beielstein and M. Preuss, "Experimental research in evolutionary computation," in *9th annual conference companion on Genetic and evolutionary computation*. ACM, 2007, pp. 3001–3020.
- [47] J. N. Hooker, "Testing heuristics: We have it all wrong," *Journal of heuristics*, vol. 1, no. 1, pp. 33–42, 1995.
- [48] J. Higgins, *Introduction to modern nonparametric statistics*. Pacific Grove, CA: Duxbury Press, 2003.



Ana Carolina Olivera is an Adjunct Researcher at National Council of Scientifics and Technological Researches from the MINCyT, Argentine. Dr. in Computer Science from Universidad Nacional del Sur. She is an Associate Professor at the Facultad de Ingeniería from Universidad Nacional de Cuyo. Her research focuses on metaheuristics and optimization in complex problems. She has published several book chapters, articles in indexed journals and proceedings of refereed international conferences. ORCID: 0000-0001-7825-1959



Pablo Javier Vidal is an Adjunct Professor at the Universidad Nacional de Cuyo, Argentine. Dr. in Software Engineering and Artificial Intelligence, from Universidad de Málaga, Spain. He is an Assistant Researcher at National Council of Scientifics and Technological Researches from the Ministerio de Ciencia y Tecnología de la Nación, Argentine. His main research topics are: parallel and distributed computing, bioinformatics and metaheuristics. ORCID: 0000-0001-6502-8010