

Planificación multiobjetivo de viajes compartidos en taxis utilizando un micro algoritmo evolutivo paralelo

Renzo Massobrio, Gabriel Fagúndez y Sergio Nesmachnow

Resumen— Este trabajo presenta la aplicación de un micro algoritmo evolutivo paralelo para resolver una variante multiobjetivo del problema de planificación de viajes compartidos en taxis. Los objetivos considerados en la variante abordada del problema son el costo total del viaje y la demora en llegar a destino por parte de cada pasajero. Esta versión del problema de viajes compartidos en taxis contempla situaciones realistas con alta aplicabilidad en la práctica, y toma en cuenta los criterios más considerados por los usuarios. El problema se resuelve con un micro algoritmo evolutivo paralelo que utiliza el modelo de subpoblaciones distribuidas y una estrategia de descomposición de dominio para contemplar las diferentes ponderaciones de los objetivos. El análisis experimental realizado sobre un conjunto de instancias reales del problema muestra que el algoritmo propuesto es capaz de encontrar significativas mejoras en ambos objetivos cuando se lo compara con estrategias ávidas intuitivas para resolver el problema.

Palabras clave— planificación multiobjetivo, viajes compartidos, algoritmos evolutivos paralelos

I. INTRODUCCIÓN

Los viajes compartidos en automóvil, conocidos bajo el término anglosajón *car pooling*, han captado el interés del público en los últimos años [1]. Compartir vehículos con personas que se dirigen a destinos cercanos, genera beneficios no solo en el plano económico, sino también en el ecológico, a niveles individuales y colectivos. Por otra parte, el *car pooling* permite minimizar los costos de traslado y la cantidad de automóviles en las rutas, reduciendo así la polución y contribuyendo a minimizar el impacto ambiental del transporte, un problema capital en las últimas décadas en las grandes ciudades [2, 3].

Los automóviles con taxímetro (taxis) constituyen un medio de transporte rápido y confortable. Sin embargo, es frecuente que los taxis no se utilicen a capacidad completa. Por este motivo, se puede aplicar el concepto de viaje compartido a los taxis de forma de obtener similares beneficios. Desde el punto de vista de los usuarios, minimizar los costos es un claro objetivo al compartir vehículos con otras personas, aunque puede no ser el único. En ciertas situaciones, minimizar la demora puede ser más gravitante que únicamente minimizar el costo de traslado. Costo y demora son objetivos contrapuestos y por tal motivo resulta importante atacar el problema multiobjetivo, hallando soluciones con diferentes valores de compromiso entre costo y demora.

En la actualidad existen servicios web que proporcionan soluciones al problema de planificación de viajes compartidos utilizando heurísticas simples (sin contemplar la inclusión de técnicas de inteligencia computacional), y algunos trabajos de investigación que han resuelto variantes del problema de planificar viajes en taxis utilizando métodos específicos. El problema de planificación de viajes compartidos es NP-difícil [4]. Para resolver instancias con dimensiones realistas es necesario utilizar heurísticas o metaheurísticas [5] que permitan hallar soluciones de calidad aceptable en tiempos razonables, especialmente si se contempla la posibilidad de calcular resultados en tiempo real para ofrecer el servicio a los usuarios.

En esta línea de trabajo, este artículo presenta un micro algoritmo evolutivo paralelo aplicado a resolver la variante multiobjetivo del problema de planificación de viajes compartidos en taxis que considera la minimización simultánea del costo del viaje y la demora en llegar a destino por parte de cada pasajero. Las principales contribuciones del trabajo consisten en: i) la implementación de un micro algoritmo evolutivo paralelo que utiliza el modelo de subpoblaciones distribuidas y una estrategia de descomposición de dominio para contemplar las diferentes ponderaciones de los objetivos, pensado para resolver eficientemente el problema y ii) el análisis experimental realizado sobre un conjunto de instancias realistas del problema, que demuestra que el algoritmo propuesto es capaz de encontrar planificaciones con significativas mejoras en ambos objetivos cuando se lo compara con estrategias ávidas intuitivas para resolver el problema.

El artículo se organiza del modo que se describe a continuación. La Sección II presenta la versión multiobjetivo del problema de planificación de viajes compartidos en taxis y su formulación matemática. El micro algoritmo evolutivo paralelo ideado para resolver el problema se introduce en la Sección III. Una revisión de los principales trabajos relacionados se ofrece en la Sección IV. Los detalles de diseño e implementación del algoritmo propuesto se describen en la Sección V. La Sección VI reporta el análisis experimental realizado sobre escenarios realistas del problema y una discusión sobre los principales resultados obtenidos. Finalmente, la Sección VII presenta las conclusiones del trabajo y formula las principales líneas de trabajo actual y futuro.

II. EL PROBLEMA DE PLANIFICACIÓN DE VIAJES COMPARTIDOS EN TAXIS (VERSIÓN MULTIOBJETIVO)

Esta sección presenta una descripción del problema de viajes compartidos en taxis, y su formulación matemática como problema de optimización multi-objetivo.

A. Descripción del problema

El problema a resolver (*taxi pooling*) modela una realidad donde un determinado número de personas, ubicadas en un mismo lugar de origen, deciden viajar hacia diferentes destinos utilizando taxis. El problema de optimización relacionado consiste en determinar el número apropiado de taxis y la distribución adecuada de pasajeros, buscando minimizar dos objetivos relevantes para los usuarios: el *costo del viaje* y la *demora percibida*. La distribución de pasajeros está restringida a la cantidad máxima de personas que pueden viajar en un mismo taxi. Se contemplan diferentes tamaños de vehículos, y una cantidad fija de vehículos de cada una de las capacidades.

Aunque el problema fue originalmente concebido para modelar el transporte en taxis en la ciudad de Montevideo, Uruguay, ni su formulación ni el algoritmo propuesto para su resolución restringen su aplicabilidad a otros escenarios, tal como se demuestra en el análisis experimental reportado en la Sección VI. Los costos asociados al transporte están modelados con una función realista que considera la tarificación basada en distancias recorridas por cada taxi. Este modelo corresponde a considerar escenarios con tráfico fluido, sin grandes embotellamientos. La incorporación de datos de tráfico en tiempo real está propuesta como una de las principales líneas de trabajo actual y futuro.

Respecto al problema, las siguientes consideraciones deben ser tenidas en cuenta:

- Cada taxi puede trasladar a un número limitado de pasajeros dependiendo de su capacidad. La cantidad de taxis disponibles de cada tipo es dada como entrada del algoritmo.
- El número máximo de taxis a utilizar para un escenario con N pasajeros es N , en el caso particular de que cada pasajero viaje en un vehículo distinto.
- El costo de un taxi está dado por la suma del costo inicial (conocido popularmente como “bajada de bandera”), más el costo determinado por la distancia recorrida desde el origen al destino. No se consideran otros posibles costos tales como esperas, propinas o cargos extras por equipaje.
- Cada pasajero tiene asociada una tolerancia que indica el tiempo que éste está dispuesto a demorar por sobre el tiempo que le demandaría un viaje directo desde el origen a su destino. La tolerancia de cada pasajero también es dada como entrada al algoritmo.

B. Formulación matemática

La formulación matemática del problema de viajes compartidos en taxis se presenta a continuación.

Dados los siguientes elementos:

- Un conjunto de pasajeros $P = \{p_1, p_2, \dots, p_N\}$; que parten de un punto geográfico común O y desean trasladarse hacia un conjunto de destinos potencialmente diferentes $D = \{d_1, d_2, \dots, d_N\}$.
- Un conjunto de taxis $T = \{t_1, t_2, \dots, t_M\}$; con $M \leq N$, con capacidades (pasajeros que pueden transportar) $K = \{K_1, K_2, \dots, K_M\}$.
- Una función C que determina cuántos pasajeros hacen uso de un taxi en un determinado viaje, siendo $C_i \leq K_i$.
- Una matriz MC de dimensión $(N+1) \times (N+1)$ que especifica el costo entre cada uno de los puntos geográficos involucrados en el problema (un punto origen y N puntos destino).
- Una matriz MT de dimensión $(N+1) \times (N+1)$ que especifica los tiempos de demora para un viaje entre cada uno de los puntos geográficos involucrados en el problema.
- Un vector T de largo N donde T_i es el tiempo extra que el usuario i tolera por encima del tiempo que tarda en ir directamente del origen a su destino.
- Una función de *costo total*, determinada por el valor B constante (costo inicial, “bajada de bandera”) y la suma para todos los taxis de los costos entre el origen, los destinos intermedios y el destino final:

$$CT = \sum_{t_i} (B + \sum_{j=1}^{C(t_i)} MC[d_{j-1}, d_j]) \quad (1)$$

- Una función de *demora total*, determinada por la diferencia entre el tiempo tolerado y el tiempo real que cada pasajero tarda en llegar a su destino:

$$DT = \sum_{t_i} \left(\sum_{j=1}^{C(t_i)} \left(T_i + MT[0, d_j] - \left(\sum_{h=1}^j (MT[h-1, h]) \right) \right) \right) \quad (2)$$

El problema consiste en hallar una *planificación de viajes*, es decir una función $f : P \rightarrow T$ para transportar los N pasajeros en L taxis ($L \leq M$) que determine cómo asignar pasajeros a taxis y el orden en que serán trasladados a los respectivos destinos, minimizando la función objetivo definida por la agregación lineal ponderada de costo y demora $FO = W_C \times CT + W_D \times DT$, siendo $W_C = 1 - W_D$.

III. COMPUTACIÓN EVOLUTIVA

A. Algoritmos evolutivos

Los Algoritmos Evolutivos (AE) son técnicas estocásticas que emulan el proceso de evolución natural de las especies para resolver problemas de optimización, búsqueda y aprendizaje [6]. En los últimos

30 años, los AE han sido exitosamente aplicados para resolver problemas de optimización subyacentes a problemas complejos del mundo real en múltiples áreas de aplicación [5].

Un AE es una técnica iterativa (cada iteración se denomina *generación*) que aplica operadores estocásticos sobre un conjunto de individuos (la *población*). Inicialmente, la población se genera aleatoriamente, o aplicando una heurística específica para el problema a resolver. Cada individuo en la población codifica una solución tentativa al problema estudiado, y tiene un valor de *fitness*, dado por una función de evaluación que determina la adecuación del individuo para resolver el problema. El AE busca mejorar el fitness de los individuos en la población, mediante la aplicación de *operadores evolutivos*, como la *recombinación* de partes de dos individuos y la *mutación* aleatoria de su codificación, guiando al AE a soluciones de mayor calidad.

El criterio de parada usualmente involucra un número determinado de generaciones, una cota de calidad sobre el mejor valor de fitness hallado, o la detección de una situación de convergencia. Políticas específicas se utilizan para seleccionar los individuos que participan en la recombinación (la *selección*) y para determinar cuáles nuevos individuos serán insertados en la población en cada nueva generación (el *reemplazo*). El AE retorna la mejor solución hallada en el proceso iterativo, tomando en cuenta la función de fitness considerada para el problema.

B. Algoritmos evolutivos paralelos

Las implementaciones paralelas se han popularizado como un mecanismo para mejorar el desempeño de los AE. Dividiendo la población o el cálculo de fitness entre varios elementos de procesamiento, los AE paralelos permiten abordar problemas de grandes dimensiones y difíciles de resolver, hallando resultados de buena calidad en tiempos razonables.

Los modelos paralelos de los AE utilizados en este estudio corresponden a la categoría de subpoblaciones distribuidas [7]. La población original se divide en varias subpoblaciones (islas), cada una de las cuales ejecuta un AE secuencial donde los individuos solo son capaces de interactuar con otros en su misma isla. Se define un operador adicional de *Migración* que permite el intercambio ocasional de individuos entre islas, introduciendo una nueva fuente de diversidad. El Algoritmo 1 presenta un esquema paralelo para un AE, donde *emigrantes* denota al conjunto de individuos a intercambiar con otra isla, seleccionados de acuerdo a una política determinada por *seleccMigracion*. El operador de migración intercambia los individuos entre islas de acuerdo a una topología de interconexión, que generalmente es un anillo unidireccional. La *CondicionMigracion* determina cuándo se lleva a cabo el intercambio de individuos.

Algorithm 1 Esquema de un AE Paralelo.

```

1: inicializar( $P(0)$ )
2:  $t \leftarrow 0$  {contador de generación}
3: evaluar( $P(0)$ )
4: mientras no se cumple CriterioParada hacer
5:   padres  $\leftarrow$  selección( $P(t)$ )
6:   hijos  $\leftarrow$  operadores de variación(padres)
7:   nueva población  $\leftarrow$  reemplazo(hijos,  $P(t)$ )
8:   evaluar( $P(t)$ )
9:    $t++$ 
10:   $P(t) \leftarrow$  nueva población
11:  si CondicionMigración entonces
12:    emigrantes  $\leftarrow$  seleccMigracion( $P(t)$ )
13:    inmigrantes  $\leftarrow$  migración(emigrantes)
14:    insertar(inmigrantes,  $P(t)$ )
15:  fin si
16: fin mientras
17: retornar mejor solución hallada

```

C. El algoritmo $p\mu$ -MOEA/D

Los AE paralelos mejoran su eficiencia computacional al trabajar con subpoblaciones que limitan las interacciones entre individuos. Sin embargo, los AE suelen perder diversidad cuando trabajan con poblaciones pequeñas, convergiendo prematuramente a soluciones lejanas al óptimo del problema.

Varias alternativas se han propuesto para mitigar el efecto de la pérdida de diversidad en los AE. En este trabajo se aplica un micro algoritmo paralelo ($p\mu$ -MOEA/D) para la resolución del problema de viajes compartidos en taxis. La idea de un micro algoritmo genético fue presentada por Coello y Pulido [8], basándose en los estudios teóricos de Goldberg sobre la convergencia al óptimo de un problema de optimización cuando se utiliza un AE con tan solo tres individuos en la población, en caso de contar con un operador apropiado para introducir diversidad ante convergencia prematura.

El algoritmo $p\mu$ -MOEA/D propuesto en este trabajo combina dos características de diseño: (1) un modelo paralelo de subpoblaciones distribuidas utilizando micropoblaciones y migración, que se utiliza como operador para proveer diversidad a la búsqueda y (2) un enfoque de descomposición de dominio similar al aplicado en el algoritmo MOEA/D [9] para resolver un problema de optimización con dos objetivos. Cada subpoblación en $p\mu$ -MOEA/D se enfoca en resolver un problema de optimización específico, aplicando un esquema de agregación lineal para los objetivos del problema, pero utilizando diferentes pares de pesos para ponderar las funciones costo y demora. Este enfoque permite muestrear el frente de Pareto del problema, a pesar de que el algoritmo no utilice un esquema explícito de asignación de fitness basado en dominancia de Pareto. El enfoque ha sido aplicado previamente al algoritmo CHC [10].

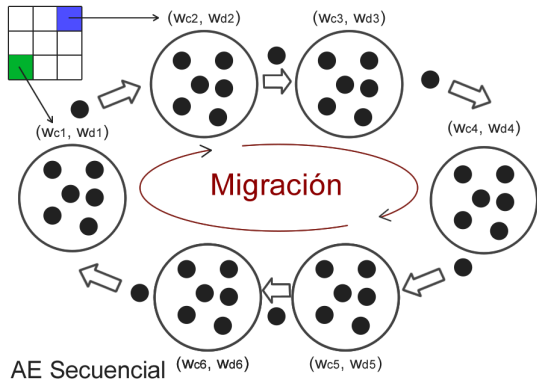


Fig. 1. Esquema del algoritmo $p\mu$ -MOEA/D

IV. TRABAJOS RELACIONADOS

Diferentes variantes del *taxi-pooling* han sido estudiadas en la literatura. Ma et al. [11] diseñaron un sistema dinámico para compartir taxis, combinando un método de búsqueda para encontrar taxis candidatos a satisfacer un pedido, y un algoritmo de planificación que inserta el viaje en el itinerario del taxi que incurra en la menor distancia adicional al atender ese pedido. Usando una base de datos de trayectorias GPS generadas por 33.000 taxis durante 3 meses en la ciudad de Beijing, el sistema alcanzó un ahorro en la distancia de un 13 %.

Contemplando el objetivo de costo, Tao et al. [12] propusieron dos heurísticas ávidas para optimizar los costos en viajes compartidos de taxis. Una prueba piloto realizada en Taipei con 10 taxis y 798 pasajeros mostró resultados promedio de 60.3 % en el porcentaje de éxito en los emparejamientos. Los resultados sobre la mejora en el consumo de combustible se reportan en términos absolutos, siendo difícil utilizarlos como base para una comparación.

Abordando el problema multiobjetivo, Yeqian et al. [13] propusieron un algoritmo de recocido simulado enfocado en minimizar el costo operativo de los taxis y maximizar la satisfacción de los usuarios. La formulación del problema tiene en cuenta la distancia real recorrida, el tiempo de recorrido extra de los taxis y el tiempo de espera de los clientes. El algoritmo propuesto redujo la distancia recorrida en 19 % y aumentó 66 % la cantidad de taxis disponibles respecto a una situación en que no se comparten taxis.

En nuestro grupo de trabajo se ha abordado la resolución de la variante mono-objetivo del problema de planificación de viajes compartidos en taxis, utilizando AE secuenciales [14] y AE paralelos [15].

El análisis de trabajos relacionados muestra que el problema de viajes compartidos en taxis es de interés en la actualidad, debido a su impacto en el medioambiente y en la economía. Sin embargo, las soluciones centradas en el usuario son escasas en la literatura, en especial aquellas que buscan optimizar más de un objetivo simultáneamente. Por lo tanto, es posible contribuir desarrollando aplicaciones que apliquen inteligencia computacional enfocadas en maximizar

el ahorro y la satisfacción de los clientes que hacen uso de viajes compartidos en taxis.

V. DISEÑO E IMPLEMENTACIÓN DE $p\mu$ -MOEA/D

Esta sección presenta los detalles de diseño e implementación de $p\mu$ -MOEA/D aplicado al problema.

A. Codificación y función de fitness

Las soluciones se representan como tuplas que contienen a los enteros entre 1 y N , representando a los pasajeros, y $N - 1$ ceros utilizados para separar pasajeros asignados a diferentes taxis. El orden en que se visitan los destinos es el especificado en la secuencia. La Figura 2 muestra un ejemplo de codificación de la solución para una instancia con 5 pasajeros.

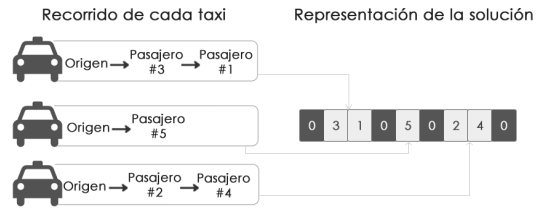


Fig. 2. Ejemplo de la codificación utilizada.

El esquema de codificación tiene las siguientes restricciones: i) cada entero debe aparecer una única vez en la tupla, ii) los ceros consecutivos tienen el mismo significado que un único cero, iii) las secuencias de dígitos distintos de cero deben ser de largo menor o igual a la máxima capacidad de vehículo disponible hasta el momento, de acuerdo al vector de capacidades de la flota de vehículos.

El fitness queda definido por la función objetivo descrita en II-B, utilizando $Wc = [0: \frac{1}{\#islas}: 1]$ y $Wd = 1 - Wc$.

B. Operadores evolutivos

Dado que la codificación utilizada tiene características y restricciones particulares, se utilizan operadores evolutivos adaptados al problema.

Inicialización de la población: Se utilizaron dos métodos de inicialización. La *inicialización aleatoria* usa un algoritmo constructivo que ubica aleatoriamente los números del 1 al N en una tupla creada con $2N - 1$ ceros. La *inicialización ávida* inserta en la población dos individuos generados con las soluciones halladas por algoritmos ávidos para costo y demora (ver Sección VI), y completa la población con individuos generados aplicando un número aleatorio de perturbaciones (i.e. intercambios de dos elementos), a las soluciones obtenidas por ambos algoritmos ávidos. Ambas estrategias de inicialización se estudian en la evaluación experimental.

Proceso de corrección: Ambas estrategias de inicialización pueden violar alguna de las restricciones definidas para la codificación de las soluciones, por lo cual se aplica una función correctiva para garantizar que las soluciones generadas sean válidas. El

algoritmo de corrección busca secuencias de dígitos distintos de cero de largo mayor a la máxima capacidad de vehículo disponible hasta el momento. Al encontrar una, se busca la primer pareja de ceros consecutivos y se mueve el primer cero hacia una posición aleatoria dentro de la secuencia encontrada, de forma de romper con el grupo inválido de dígitos distintos de cero. Este proceso continúa hasta que se recorre la solución por completo, garantizando que la misma cumple con todas las restricciones.

Selección: se utilizó el operador de *selección por torneo* para otorgar una apropiada presión selectiva a las micropoblaciones. Experimentos iniciales mostraron que la selección proporcional no es capaz de ofrecer suficiente diversidad, conduciendo a convergencia prematura.

Recombinación: se aplicó una variante del operador *Position Based Crossover* (PBX), explicado en el Algoritmo 2.

Algorithm 2 Esquema de la recombinación *PBX*

- 1: Seleccionar aleatoriamente varias posiciones del padre 1.
 - 2: generar parcialmente el hijo 1, copiando los valores de las posiciones elegidas del padre 1.
 - 3: Marcar las posiciones del padre 2 que ya fueron seleccionados en el padre 1.
 - 4: Desde el inicio del padre 2, seleccionar secuencialmente el siguiente valor no marcado, e insertarlo en la primer posición libre del hijo.
-

De forma de cumplir con las restricciones impuestas por la codificación de la solución, se aplica la misma función correctiva utilizada en la inicialización sobre el hijo resultante del cruzamiento PBX. La Figura 3 muestra un ejemplo del cruzamiento PBX para una instancia con 5 pasajeros.

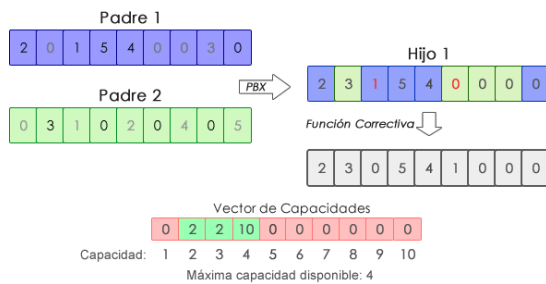


Fig. 3. Ejemplo del operador PBX aplicado.

Mutación: Se aplicó el operador de mutación por intercambio (*Exchange Mutation*, EM), que selecciona e intercambia dos posiciones de la solución y luego aplica la función correctiva para corregir las potenciales soluciones inválidas.

Migración: El operador de migración se aplica asincrónicamente, considerando a las islas conectadas en una topología de anillo unidireccional. Los emigrantes se seleccionan mediante torneo y reemplazan a los peores individuos de la isla destino. La

migración contribuye a mejorar la diversidad, evitando la convergencia prematura del algoritmo.

VI. ANÁLISIS EXPERIMENTAL

Esta sección reporta el análisis experimental de $p\mu$ -MOEA/D sobre un conjunto de instancias realistas del problema.

A. Plataforma de desarrollo y ejecución

El algoritmo $p\mu$ -MOEA/D se desarrolló en C++, utilizando Malva [16]. La evaluación experimental se realizó en un HP Proliant DL585, AMD Opteron 6272 a 2.09GHz, 48GB de RAM y Gigabit Ethernet, en la plataforma de cómputo de alto desempeño Cluster FING (Universidad de la República, Uruguay, <http://www.fing.edu.uy/cluster>) [17].

B. Instancias del problema

Para evaluar el algoritmo propuesto, se generó un conjunto de instancias realistas del problema de planificación de viajes compartidos en taxis. Se utilizó un enfoque metodológico específico para generar los escenarios, siguiendo los lineamientos presentados en los trabajos relacionados y utilizando servicios disponibles para recabar información sobre pedidos de taxi, mapas, y tarifas, incluyendo:

- *Generador de Pedidos de Taxis (Taxi Query Generator, TQG)*, una herramienta que utiliza información de una base de datos de trayectorias de taxis, obtenidas con dispositivos GPS en 10.357 taxis de la ciudad de Beijing, para generar pedidos de taxis realistas. Los datos son una muestra de los utilizados por Ma *et al.* [11]. Para obtener instancias aplicables al problema estudiado en este artículo, se implementó un script que agrupa viajes de TQG que comienzan en orígenes cercanos y devuelve instancias del problema de un origen a muchos destinos.

- La interfaz para aplicaciones de *TaxiFareFinder* [18], que se utilizó para obtener la matriz de costos de cada instancia del problema generada, junto a los precios correspondientes a la bajada de bandera. Cada par de coordenadas de una determinada instancia del problema es enviado a la interfaz de TaxiFareFinder para obtener el costo entre cada punto.

Además, se generó un conjunto de instancias para la ciudad de Montevideo, Uruguay, considerando un origen ubicado en un punto de interés de la ciudad y un conjunto de destinos aleatorios en la ciudad. Los orígenes se corresponden con cuatro situaciones particulares: un local bailable, un evento deportivo, un centro de estudios terciario y un centro comercial.

A partir del mapa de cada instancia, se generaron distintas variantes al combinar diferentes valores para el nivel de tolerancia en la demora de cada pasajero, y las capacidades de vehículos disponibles.

C. Ajuste paramétrico

Debido a la cualidad estocástica de los AEs, es necesario ajustar sus parámetros previamente al análisis experimental. Para la configuración paramétrica

se utilizaron cuatro instancias del problema (distintas de las utilizadas en la evaluación experimental para evitar sesgo). Luego de una evaluación inicial, el tamaño de la micro población se fijó en 15 individuos. El operador de migración se fijó de forma de seleccionar 2 individuos de cada isla mediante una selección por torneo, y enviarlos a la isla vecina, donde reemplazan a los peores individuos. El operador de migración se aplica cada 1000 generaciones.

El ajuste paramétrico se focalizó en estudiar las probabilidades de recombinación (p_C , valores candidato 0.6, 0.75 y 0.95) y de mutación (p_M , valores candidato 0.001, 0.01 y 0.1). Se estudiaron todas las combinaciones de valores sobre las cuatro instancias de calibración, realizando 30 ejecuciones independientes de 20000 generaciones en cada caso.

La Figura 4 muestra los valores promedio para las funciones de costo y demora obtenidos en la instancia #2, para cada combinación de parámetros.

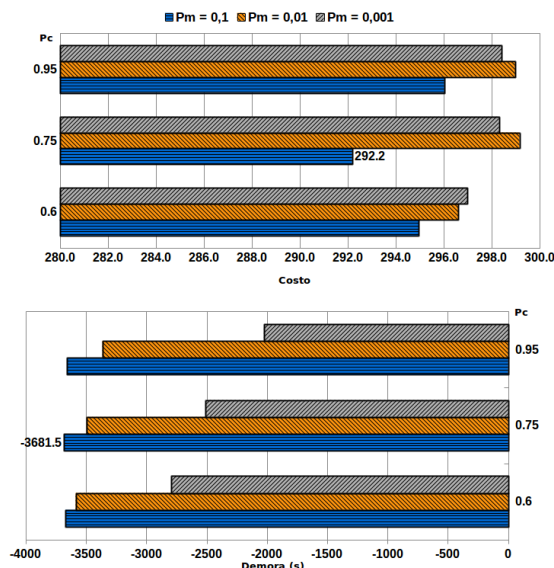


Fig. 4. Configuración paramétrica (instancia #2)

Los resultados del análisis paramétrico indican que los mejores resultados para las instancias estudiadas se obtienen utilizando $p_C=0.75$ y $p_M=0.1$.

D. Evaluación experimental

El análisis experimental se enfocó en evaluar el desempeño de $p\mu$ -MOEA/D y la calidad de las soluciones alcanzadas. Se utilizaron 39 instancias del problema (27 ubicadas en Beijing y 12 en Montevideo) que corresponden a escenarios de entre 9 y 50 pasajeros. Estas dimensiones son razonablemente mayores a lo que se puede esperar en instancias generadas por usuarios finales de un sistema de planificación en la vida real. Se realizaron 30 ejecuciones independientes de cada instancia, con un criterio de parada de 20000 generaciones.

D.1 Comparación contra dos algoritmos ávidos que resuelven el problema

Un algoritmo ávido es un método de construcción de soluciones que toma decisiones localmente óptimas en cada paso. Es relevante comparar el desempeño de $p\mu$ -MOEA/D frente a un algoritmo ávido en términos de la calidad de las soluciones alcanzadas y de la eficiencia computacional. Con este propósito, se implementaron dos algoritmos ávidos para resolver el problema de planificación de viajes compartidos en taxis, contemplando los dos objetivos considerados en la función de fitness.

El algoritmo ávido que minimiza el costo se basa en aplicar una técnica de agrupamiento simple e intuitiva. Toma un taxi, y agrega el destino más cercano al taxi actual hasta completarlo (en cuyo caso forma un nuevo taxi), o hasta que todos los pasajeros hayan sido asignados. Una excepción ocurre cuando el costo de añadir un nuevo destino al taxi actual es mayor al costo de asignar un nuevo taxi que transporte únicamente a ese pasajero. En este caso, se forma un nuevo taxi y se “cierra” el anterior. El algoritmo emula una estrategia intuitiva para resolver el problema con un enfoque de agregación de estrategias de decisión localmente óptimas, y se aproxima a lo que un grupo de usuarios humanos pueden idear de manera simple para resolver el problema.

El algoritmo ávido que minimiza la demora de los usuarios aplica una técnica simple e intuitiva que toma los pasajeros más apurados y los ubica en un nuevo taxi, en la primera ubicación. Luego, toma los pasajeros en orden según su urgencia, selecciona el taxi que minimice su demora tomando en cuenta el último pasajero agregado, y añade al pasajero en la última ubicación. Si el taxi alcanza el máximo de pasajeros, el mismo se considera como “cerrado”. Al cerrar un taxi, existe la chance de que alguno de los ya formados pero no cerrados, supere la nueva cota superior de pasajeros según el vector de capacidades. Se utiliza una corrección que en caso de detectar que un taxi supera la nueva cota máxima de pasajeros, lo elimina y crea dos nuevos taxis de capacidad uno con sus dos primeros pasajeros.

E. Resultados y discusión

La Tabla I reporta los valores de costo y demora alcanzados en cada conjunto de instancias por el $p\mu$ -MOEA/D con las dos inicializaciones estudiadas y por los algoritmos ávidos. Los resultados muestran que el $p\mu$ -MOEA/D es poco sensible a la estrategia de inicialización elegida, lo que sugiere robustez en la búsqueda. Para el resto de la evaluación experimental se elige la inicialización ávida por alcanzar mejores resultados en instancias de dimensiones grandes.

La Tabla II reporta las mejoras en costo y demora obtenidas por el $p\mu$ -MOEA/D con inicialización ávida, sobre los algoritmos ávidos para costo y demora. Adicionalmente, se indica entre paréntesis el tiempo necesario en segundos para alcanzar dicha mejora.

Se observa que en el mejor caso se alcanzan mejoras de hasta un 69.0 % en el costo y 135.9 % en la demora. Adicionalmente, las mejoras se alcanzan en tiempos muy cortos, de apenas un par de segundos en la mayoría de los casos, lo que permite que $p\mu$ -MOEA/D sea útil para resolver instancias reales del problema en línea, como en el caso de la aplicación web instalada en mepaseaste.uy.

La Figura 5 muestra las soluciones no dominadas alcanzadas a las 0, 10000, y 20000 generaciones por $p\mu$ -MOEA/D, en 30 ejecuciones independientes, para una instancia representativa de cada grupo de instancias de prueba. Como referencia, se incluyen las soluciones calculadas por los algoritmos ávidos.

Se puede apreciar que $p\mu$ -MOEA/D permite aproximar correctamente un hipotético frente de Pareto real para las distintas instancias de prueba. Es de destacar que, además de alcanzar soluciones de mejor calidad con el correr de las generaciones, $p\mu$ -MOEA/D mantiene bien muestreado tanto los extremos del frente como las soluciones cercanas al vector ideal. Los resultados alcanzados muestran que el modelo de descomposición de datos elegido permite mantener un buen nivel de diversidad a lo largo de la ejecución del algoritmo.

VII. CONCLUSIONES

Este trabajo presentó el diseño e implementación de un micro AE paralelo para resolver el problema multiobjetivo de planificación de viajes compartidos en taxis, desde un origen a múltiples destinos. Los objetivos considerados son el costo total del viaje y la demora percibida por los pasajeros al compartir su viaje (con respecto a viajar en un único taxi).

El micro algoritmo evolutivo paralelo propuesto conforma una precisa solución para el problema abordado. El análisis experimental realizado sobre un conjunto de 39 instancias del problema construidas en base a datos reales tomados de GPS de taxis de la ciudad de Beijing y adaptadas a la ciudad de Montevideo, mostró que los resultados del algoritmo propuesto permiten mejorar significativamente los resultados—hasta un 69.0 % en costo y hasta un 135.9 % en demora—sobre los calculados empleando algoritmos ávidos que simulan el comportamiento intuitivo de un grupo de usuarios humanos al enfrentarse al problema. Los resultados se obtienen en tiempos de ejecución reducidos, permitiendo su aplicación para la resolución en línea del problema.

Las principales líneas de trabajo futuro se orientan a incorporar datos del tráfico en tiempo real, que permitan estimar con mayor precisión las tarifas y elegir rutas alternativas según corresponda, así como también datos acerca de la disponibilidad de vehículos y sus distintas capacidades. Asimismo, se propone estudiar AE que utilicen un enfoque multiobjetivo explícito para la resolución del problema.

En la página web del proyecto www.fing.edu.uy/inco/grupos/cecal/hpc/AG-Taxi se encuen-

tran publicados los siguientes recursos: i) código fuente de $p\mu$ -MOEA/D y de los algoritmos ávidos; ii) generador de instancias del problema; iii) instancias del problema utilizadas y resultados numéricos detallados de los experimentos realizados.

AGRADECIMIENTOS

Esta investigación ha sido parcialmente financiada por ANII y PEDECIBA, Uruguay.

REFERENCIAS

- [1] N. Fellows and D. Pitfield, "An economic and operational evaluation of urban car-sharing," *Transportation Research Part D: Transport and Environment*, vol. 5, no. 1, pp. 1–10, 2000.
- [2] E. Ferrari, R. Manzini, A. Pareschi, A. Persona, and A. Regattieri, "The car pooling problem: Heuristic algorithms based on savings functions," *Journal of Advanced Transportation*, vol. 37, pp. 243–272, 2003.
- [3] R. Katzev, "Car sharing: A new approach to urban transportation problems," *Analyses of Social Issues and Public Policy*, vol. 3, no. 1, pp. 65–86, 2003.
- [4] A. Letchford, R. Eglese, and J. Lysgaard, "Multistars, partial multistars and the capacitated vehicle routing problem," *Mathematical Programming*, vol. 94, no. 1, pp. 21–40, 2002.
- [5] S. Nesmachnow, "Metaheuristics as soft computing techniques for efficient optimization," in *Encyclopedia of Information Science and Technology*, M. Khosrow-Pour, Ed., pp. 1–10. IGI Global, 2014.
- [6] T. Bäck, D. Fogel, and Z. Michalewicz, Eds., *Handbook of evolutionary computation*, Oxford Univ. Press, 1997.
- [7] E. Alba, G. Luque, and S. Nesmachnow, "Parallel metaheuristics: Recent advances and new trends," *International Transactions in Operational Research*, vol. 20, no. 1, pp. 1–48, 2013.
- [8] C. Coello and G. Pulido, "A micro-genetic algorithm for multiobjective optimization," in *Proc. of the 1st Int. Conf. on Evolutionary Multi-Criterion Optimization*, London, UK, 2001, pp. 126–140.
- [9] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, 2007.
- [10] S. Nesmachnow and S. Iturriaga, "Multiobjective grid scheduling using a domain decomposition based parallel micro evolutionary algorithm," *Int. Journal of Grid and Utility Computing*, vol. 4, pp. 70–84, 2013.
- [11] Y. Zheng S. Ma and O. Wolfson, "T-share: A large-scale dynamic taxi ridesharing service," in *IEEE 29th Int. Conf. on Data Engineering*, 2013, pp. 410–421.
- [12] C. Tao and C. Chen, "Heuristic algorithms for the dynamic taxipooling problem based on intelligent transportation system technologies," in *4th Int. Conf. on Fuzzy Systems and Knowledge Discovery*, 2007, pp. 590–595.
- [13] Li F. Qiu H. Xu Y. Lin, W., "Research on optimization of vehicle routing problem for ride-sharing taxi," in *8th Int. Conf. on Traffic and Transportation Studies Changsha, China*, 2012.
- [14] R. Massobrio, G. Fagúndez, and S. Nesmachnow, "A parallel micro evolutionary algorithm for taxi sharing optimization," in *VII ALIO/EURO Workshop on Applied Combinatorial Optimization*, Montevideo, Uruguay, 2014.
- [15] G. Fagúndez, R. Massobrio, and S. Nesmachnow, "Resolución en línea del problema de viajes compartidos en taxis usando algoritmos evolutivos," in *Conf. Latinoamericana de Informática*, Montevideo, Uruguay, 2014.
- [16] "The Malva Project: A framework for computational intelligence in C++," [Online 11-2014] <https://github.com/themalvaproject>.
- [17] S. Nesmachnow, "Computación científica de alto desempeño en la Facultad de Ingeniería, Universidad de la República," *Revista de la Asociación de Ingenieros del Uruguay*, vol. 61, pp. 12–15, 2010.
- [18] "TaxiFareFinder API," [Online 11-2014] <http://www.taxifarefinder.com>.

TABLA I
RESULTADOS ALCANZADOS CON CADA TÉCNICA

instancias	métrica		p μ -MOEA/D (R)	p μ -MOEA/D (G)	ávido
chicas	costo	min	105.42	105.42	131.35
		avg	312.34	244.18	215.08
	demora	min	-2877.00	-2877.00	6039.30
		avg	2115.48	5884.63	12210.34
medianas	costo	min	283.22	296.80	428.11
		avg	840.46	612.49	449.68
	demora	min	-12406.20	-12406.20	11613.10
		avg	5286.01	17666.75	23586.89
grandes	costo	min	369.51	328.88	384.20
		avg	768.76	586.53	500.45
	demora	min	-3562.70	-3581.70	5886.05
		avg	5545.52	24198.86	22310.47
Montevideo	costo	min	146.31	149.33	152.31
		avg	361.06	300.72	260.03
	demora	min	-2028.85	-2028.85	1501.70
		avg	1307.22	5288.68	6609.86

TABLA II
MEJORA ALCANZADA SOBRE EL ALGORITMO ÁVIDO

instancias	métrica		<i>generaciones</i>		
			10000	15000	20000
chicas	max. mejora ávido_c	max	36.0 % (0.6s)	36.0 % (0.9s)	36.0 % (1.2s)
		avg	21.0 % (1.0s)	21.0 % (1.4s)	21.0 % (1.8s)
		std	12.2	12.2	12.2
	max. mejora ávido_d	max	122.4 % (1.2s)	122.4 % (1.8s)	122.4 % (2.4s)
		avg	111.1 % (0.8s)	111.1 % (1.2s)	111.1 % (1.6s)
		std	5.0	5.0	5.0
medianas	max. mejora ávido_c	max	31.6 % (1.3s)	32.5 % (2.0s)	32.5 % (2.6s)
		avg	19.2 % (1.9s)	19.7 % (2.8s)	19.9 % (3.7s)
		std	9.5	9.4	9.3
	max. mejora ávido_d	max	135.9 % (1.8s)	135.9 % (2.7s)	135.9 % (3.6s)
		avg	120.4 % (1.8s)	120.4 % (2.7s)	120.4 % (3.6s)
		std	8.8	8.8	8.8
grandes	max. mejora ávido_c	max	67.3 % (2.5s)	68.1 % (3.7s)	69.0 % (5.0s)
		avg	15.5 % (2.9s)	16.8 % (4.3s)	17.2 % (5.7s)
		std	19.3	19.3	19.5
	max. mejora ávido_d	max	128.5 % (3.1s)	130.4 % (4.6s)	131.3 % (6.1s)
		avg	115.1 % (2.8s)	115.7 % (4.1s)	116.0 % (5.5s)
		std	6.6	6.9	7.1
Montevideo	max. mejora ávido_c	max	33.9 % (0.9s)	33.9 % (1.3s)	33.9 % (1.8s)
		avg	14.3 % (1.1s)	14.4 % (1.6s)	14.4 % (2.2s)
		std	8.9	8.9	8.9
	max. mejora ávido_d	max	123.5 % (0.5s)	123.5 % (0.7s)	123.5 % (0.9s)
		avg	116.3 % (1.0s)	116.3 % (1.5s)	116.3 % (2.0s)
		std	4.7	4.7	4.7

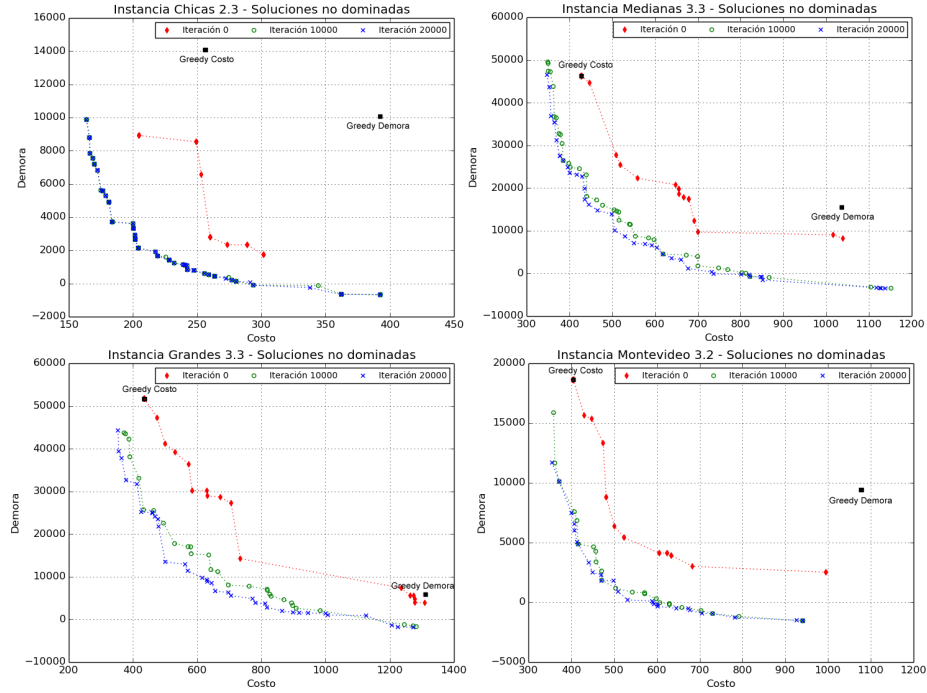


Fig. 5. Frentes de Pareto para una instancia de cada conjunto.