

# Aplicación del algoritmo genético a problemas de transporte

Alejandro Hermosillo García, Mauricio Andrés Perez Flores, Tom Bruno Arc

A01634552, A01639917, A01562405

Dr. Rajesh Roshan Biswal, Prof. Salvador Hinojosa

**Abstract**—Gracias al crecimiento exponencial de la tecnología en los últimos años, comprar y vender en línea se ha convertido en una actividad rutinaria en la vida de muchas personas, lo cual implica la entrega de dichos artículos. Para garantizar un buen servicio y mantener satisfecho al cliente se debe tener una buena logística de ruteo. Para poder solucionar un problema de transporte se suele buscar minimizar las distancias recorridas, pero también es de suma importancia optimizar el uso de la flota. Es por esto, que en este documento se implementa un algoritmo genético para resolver dicha clase de problemas. Los resultados del algoritmo NSGA-II demuestran ser buenos ya que optimizaron de manera correcta el problema para todas las fitness functions.

**Index Terms**—Algoritmos Genéticos, NSGA-II, Frente de Pareto, MOTSP

## I. INTRODUCCIÓN

El paso más crítico en el proceso de entrega es el último, cuando un artículo se traslada de un centro de distribución a su destino final y es el que las empresas quieren garantizar que sea tan rápido y eficiente como sea posible.

Gracias al crecimiento exponencial de la tecnología en los últimos años, comprar y vender en línea se ha convertido en una actividad rutinaria en la vida de muchas personas. Casi cualquier cosa se puede comprar a través del comercio electrónico. Poder pedir cualquier cosa sin esperar más de tres o cuatro días a la llegada del producto y sin tener que ir personalmente a una tienda es un hecho que ha revolucionado el mundo.

Desde el año 2020, debido a la pandemia, el comercio electrónico ha mostrado un crecimiento grande y un mayor impacto en la economía mundial. Solamente en México, según un estudio realizado por la MX Internet Association en 2018, el valor de esta industria pasó de 2,450 millones de pesos mexicanos en 2009 a 233,660 millones de pesos mexicanos. Según BBVA, en 2020 el 60 por ciento de sus transacciones se realizaron a través de comercio electrónico. Datos extraídos de la Asociación Mexicana de Ventas en Línea (AMVO) señalan que el comercio electrónico se había incrementado en 20 mil millones de pesos, lo que representa un 8 por ciento más con respecto al 2019.

## II. TSP

El Traveling Salesman Problem (TSP) o Problema del Agente Viajero, hace referencia a la problemática de encontrar la ruta más corta y al mismo tiempo más eficiente para ir de un punto a otro. Sabemos que pueden existir múltiples formas

de llegar al mismo lugar, pero elegir la más eficiente está directamente relacionada con reducir los costos de traslado, pero no significa que este sea el único punto de entrega que se tenga, la primera variación del método TSP es el MTSP.

- **MTSP**(Problema del viajero múltiple): Esta variación del TSP tradicional tiene varios vendedores involucrados para visitar un número determinado de ubicaciones exactamente una vez y regresar a la posición inicial con el mínimo costo de viaje.
- **VRP** (Problema de enrutamiento de vehículos): El VRP es una optimización combinatoria que implica encontrar un diseño óptimo de rutas recorridas por una flota de múltiples vehículos para atender a un conjunto de clientes.
- **MOTSP** (multiobjective optimization TSP): Este método se relaciona con problemas de optimización matemática que involucran más de una función objetivo que deben ser optimizadas simultáneamente, lo cual lleva a un serie de soluciones que se acercan a el frente de Pareto.

Gracias a la complejidad del problema y sus diversas implementaciones, se han mezclado los diferentes métodos de TSP para encontrar la solución óptima. Es por ello que esta metodología es utilizada por diversas empresas, especialmente las de paquetería, para generar rutas con menor tiempo y distancia con el fin de ahorrar dinero y poder realizar la mayor cantidad de entregas posible.

## III. ALGORITMOS GENÉTICOS (GA)

El algoritmo genético es un método para resolver problemas de optimización restringidos y no restringidos que se basa en la selección natural, el proceso que impulsa la evolución biológica. El algoritmo genético modifica repetidamente una población de soluciones individuales. En cada paso, el algoritmo genético selecciona individuos de la población actual para que sean padres y los utiliza para producir hijos para la próxima generación. Durante generaciones sucesivas, la población "evoluciona" hacia una solución óptima. La solución de un problema de optimización utilizando GA implica una búsqueda estocástica del espacio de solución utilizando cadenas de números enteros, conocidos como cromosomas, que representan los parámetros que se están optimizando.

## IV. FUNCIONES

Para la solución de este problema se implemento la estructura de programación orientada a objetos, dentro de la cual se separo el problema en las distintas funciones del GA:

#### A. *initialize()*:

Se genera aleatoriamente una población inicial constituida por posibles soluciones del problema llamados cromosomas.

#### B. *selection()*:

Este es un conjunto de reglas que sirven para elegir a los progenitores de la siguiente generación.

#### C. *crossover()*:

Es un operador genético utilizado para generar variación en la programación de un cromosoma o cromosomas de una generación a la siguiente. Los progenitores escogidos en la selección se reproducirán a través del cruzamiento genético y generarán hijos como su descendencia.

#### D. *mutation()*:

La mutación en un algoritmo genético es una operación cuyo objetivo es generar nueva información dentro de la población de soluciones para obtener una mejor exploración del espacio de búsqueda.

#### E. *objective functions()*:

Estas funciones son los elementos utilizados para decidir los valores adecuados de los cromosomas de decisión que resuelven el problema de optimización. También son llamadas fitness functions.

#### F. *pareto front()*:

Aquí se calcula cuáles cromosomas tienen dominancia sobre otros con ayuda de las funciones objetivo, y se guarda las mejores soluciones no dominadas.

#### G. *get best()*:

Esta función retorna al final de ejecución las soluciones del frente de pareto.

## V. METODOLOGÍA

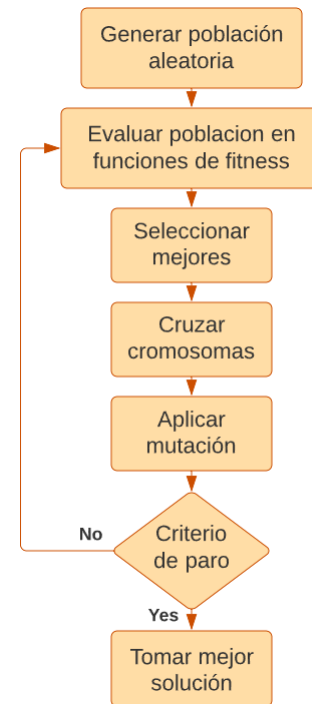


Fig. 1: Diagrama de la metodología

## VI. RESULTADOS

Para observar si el funcionamiento del algoritmo era el deseado, se hicieron varias pruebas, en las cuales se jugó con los valores de los hiperparámetros que recibe el algoritmo. A continuación se presentan los resultados de la corrida que devolvió los valores más prometedores:

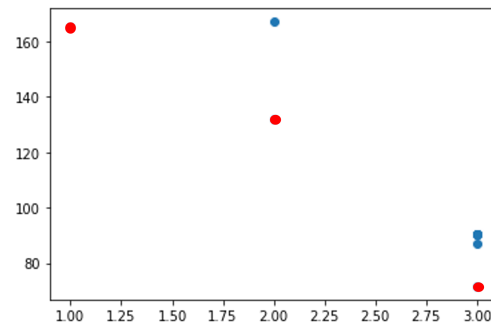


Fig. 2: Frente final

Se observa en el frente final que hay 3 soluciones no dominadas para ese problema en particular. En la primera y segunda solución se utilizan 3 vehículos distintos y se puede observar que entre estas dos soluciones no hay mucha variación. Lo único que cambia fue el orden de entrega para el vehículo cuya ruta se representa con la línea naranja. No obstante, la tercera solución utiliza un solo vehículo y la ruta graficada se asemeja al resultado que se obtendría con utilizar un algoritmo ya establecido para la resolución de problemas de tipo TSP y VRP. El valor de los hiperparámetros para esta

corrida en particular fue tomar 10 puntos (ciudades), iterar 20 veces, generar 10 cromosomas y una tasa de mutación de 0.99. Se observa que la tasa de mutación fue un valor muy alto. Esto se debe a que debido a la manera en la que se definió el cromosoma y el crossover, hay mucha explotación en las rutas de cada camión, pero no hay mucha exploración de que pasaría si alguna ciudad perteneciera a alguna otra ruta. Con la alta tasa de mutación se pretende proveer al algoritmo la oportunidad de tener mayor variación en las rutas y una mayor exploración. Es precisamente por esta técnica de crossover, por la cual el algoritmo converge de manera rápida y eficiente.

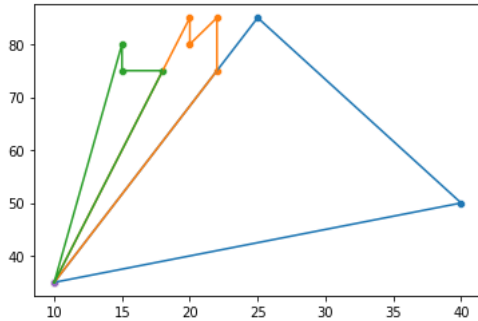


Fig. 3: Solución 1

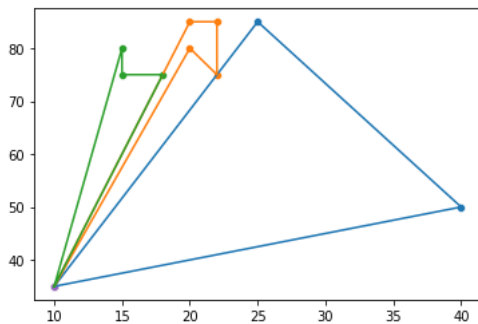


Fig. 4: Solución 2

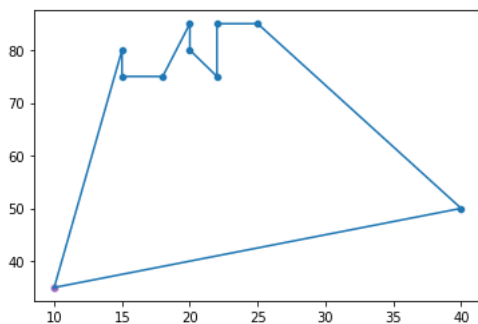


Fig. 5: Solución 3

Se intentó correr el algoritmo tomando en cuenta una mayor cantidad de ciudades, menores tasas de mutación y mayor número de cromosomas y una gran variedad de iteraciones. Encontrar la combinación de hiperparámetros correcta generaba soluciones que no se veían tan óptimas como la mostrada anteriormente, pero si generaba soluciones factibles.

## VII. CONCLUSIONES

La mayor dificultad al realizar el algoritmo capaz de resolver la presente situación es como se definirá el cromosoma y cómo se definirán las operaciones de crossover y de mutación. De eso depende si el algoritmo es capaz de encontrar buenas soluciones o no. También se concluye que uno de los desafíos de resolver el problema con una flota heterogénea es como realizar un crossover que logre mantener la optimalidad de los padres, pero que a la vez ayude a explorar más el entorno. De la misma forma es complicado obtener un operador de mutación que logre la heterogeneidad de los autos y ayuda explorar más el entorno.

## REFERENCES

- [1] E. Reyes, "Estos cuatro retos acechan el crecimiento del ecommerce mexicano," 2020. [Online]. Available: <https://expansion.mx/tecnologia/2020/07/14/estos-cuatro-retos-acechan-el-crecimiento-del-ecommerce-mexicano>
- [2] G. Chávez, "El e-commerce crecerá 60% en 2020 impulsado por covid-19," 2020. [Online]. Available: <https://expansion.mx/tecnologia/2020/04/09/el-e-commerce-crecera-60-en-2020-impulsado-por-covid-19>
- [3] A. de Internet MX, "Tendencias de crecimiento del e-commerce en México 2020," 2020. [Online]. Available: <https://www.asociaciondeinternet.mx/tendencias-de-crecimiento-del-e-commerce-en-mexico-2020>
- [4] S. S. Özark, L. P. Veelenurf, T. V. Woensel, and G. Laporte, "Optimizing e-commerce last-mile vehicle routing and scheduling under uncertain customer presence," *Transportation Research Part E: Logistics and Transportation Review*, vol. 148, p. 102263, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1366554521000399>
- [5] Y. Luo, B. Golden, S. Poikonen, and R. Zhang, "A fresh look at the traveling salesman problem with a center," *Computers Operations Research*, vol. 143, p. 105748, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054822000454>
- [6] Z. Zhang and J. Yang, "A discrete cuckoo search algorithm for traveling salesman problem and its application in cutting path optimization," *Computers Industrial Engineering*, vol. 169, p. 108157, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835222002273>
- [7] B. S. López, "Problema del agente viajero - tsp," 2019. [Online]. Available: <https://www.ingenieriaindustrialonline.com/investigacion-de-operaciones/problema-del-agente-viajero-tsp/>