

Diseño de rutas para estaciones de cargas eléctricas de vehículos eléctricos con algoritmos Metaheurísticos

Design of Routes for Electric Vehicle Charging Stations Using Metaheuristic Algorithms

Ortiz-Aguilar, L.^{1*} , Palacios-Ortega, M.² , Xoca Orozco, L. A.³  (*lucero.oa@purisima.tecnm.mx)

Recibido: 31 de mayo de 2024 Aceptado: 21 de junio de 2024

RESUMEN

En este trabajo se plantea un diseño de rutas para estaciones de carga eléctrica de vehículos eléctricos (VE) en el Estado de Guanajuato, utilizando algoritmos metaheurísticos. El crecimiento de la movilidad eléctrica plantea desafíos significativos en la planificación y optimización de infraestructuras de carga. En respuesta, este artículo propone una metodología basada en algoritmos metaheurísticos, específicamente algoritmos de búsqueda local iterada y la metodología de diseño propuesta por Ortiz-Aguilar et al. (2015) [1], para optimizar la ubicación y distribución de estaciones de carga, maximizando la cobertura de

servicio. Se presenta un estudio de caso detallado, utilizando datos geoespaciales de las estaciones de la República Mexicana, para validar la eficacia y viabilidad de la propuesta. Los resultados muestran que los enfoques heurísticos dentro de un algoritmo de búsqueda local iterada que ofrecen soluciones competitivas en términos de cobertura de servicio, lo que demuestra su utilidad en la planificación de infraestructuras de carga para VE.

Palabras clave: Problema del agente viajero, Aplicaciones de la IA, Metaheurísticas, Vehículos eléctricos, Electromovilidad

¹División de Sistemas Automotrices, Tecnológico Nacional de México / ITS de Purísima del Rincón, Blvd. del Valle 2301, Guardarrayas, 36418 Purísima de Bustos, Gto.

²División de Ingeniería Industrial, Tecnológico Nacional de México / ITS de Purísima del Rincón, Blvd. del Valle 2301, Guardarrayas, 36418 Purísima de Bustos, Gto.

³División de Ingeniería Bioquímica, Tecnológico Nacional de México / ITS de Purísima del Rincón, Blvd. del Valle 2301, Guardarrayas, 36418 Purísima de Bustos, Gto.



ABSTRACT

In this work, a route design for electric vehicle (EV) charging stations in the State of Guanajuato is proposed using metaheuristic algorithms. The growth of electric mobility poses significant challenges in the planning and optimization of charging infrastructures. In response, this article proposes a methodology based on metaheuristic algorithms, specifically Iterated Local Search algorithms and the design methodology proposed by Ortiz-Aguilar [1], to optimize the location and distribution of charging stations, maximizing service coverage. A detailed case study is presented, using geospatial data from stations in the Mexican Republic, to validate the efficacy and feasibility of the proposal. The results show that metaheuristic approaches offer competitive solutions in terms of service coverage, demonstrating their utility in the planning of EV charging infrastructures.

Keywords: *TSP, Metaheuristics, Electric vehicles, Electromobility*

INTRODUCCIÓN

En el contexto actual de los vehículos se está optando por cambiar de los automóviles de combustión interna a los eléctricos. En México se busca reducir las emisiones de gases de efecto invernadero, ya que de acuerdo con el inventario Nacional de Emisiones (1990-2019) [2], menciona que el 24.5 % de las emisiones de CO_2 vienen del sector transporte. El nuevo reto será el generar la infraestructura necesaria para ubicar estaciones de carga para vehículos eléctricos. Al igual que las gasolineras tradicionales, las estaciones de carga pueden tener diferentes capacidades y opciones de precios y el costo de recarga de cada opción puede variar de una estación a otra.

Una estrategia de carga inteligente puede contribuir de manera importante a la gestión eficiente de los recursos disponibles en las redes de los Vehículos eléctricos (EV). Al reconocer la importancia de la carga inteligente en el contexto de las redes EV, algunas investigaciones han presentado varias estrategias de carga inteligente en los últimos años, cuyo objetivo es aumentar la autonomía y el tiempo de carga [3].

Dado el contexto anterior, se puede considerar para calcular el costo de energía el camino hasta el destino para reducir el tiempo y el coste de la carga. Esto proporcionó la motivación para esta investigación.

La propuesta de este trabajo es considerar las estaciones de gasolina existentes como posibles puntos de recarga para los EV, dado que la demanda de combustibles disminuirá a medida que aumente el número de vehículos eléctricos. Para resolver este problema, se utilizó una técnica del estado del arte bien conocida como el algoritmo de búsqueda local iterada para aproximar y ofrecer una solución factible.

El artículo se organiza de la siguiente manera, en la sección Estado del Arte se describe el estado

del arte de la investigación. En la sección Marco Teórico, se describe el marco teórico y en la sección Metodología la metodología. En la sección Resultados se reportan los resultados de esta investigación y finalmente en la última sección se dan conclusiones del tema.

ESTADO DEL ARTE EN ELECTROMOVILIDAD Y SUS APLICACIONES CON IA

En esta sección se da una visión general de estado del arte en las investigaciones relacionadas a la electromovilidad y aplicaciones de la inteligencia artificial a vehículos eléctricos.

En la investigación realizada por Rajawat et al. (2024) [4] estudian el problema de coordinación de carga, donde un conjunto de camiones tiene como objetivo planificar dónde y durante cuánto tiempo cargar a lo largo de rutas previamente planificadas para su propio beneficio, mientras que los recursos de carga son limitados. Para facilitar la planificación de la carga de camiones y mitigar la congestión en las estaciones de carga, dicho trabajo propone un marco de coordinación distribuida de dos capas. Los autores calculan los planes de carga de los camiones resolviendo problemas de optimización de enteros mixtos, y las estaciones programan las órdenes de carga de acuerdo con la regla de "*primero en llegar, primero en ser atendido*". Su esquema solo requiere un simple intercambio de información entre camiones y estaciones, esto permite cálculos completamente distribuidos y la replanificación por parte de los camiones al acercarse a cada estación. Además, realizaron estudios numéricos basados en la red de carreteras sueca y compararon los resultados con planes de carga calculados fuera de línea. Se demostró que el marco trajo grandes beneficios al reducir los costos colectivos de todas las partes involucradas. Además, proponen integrar la estimación de multitudes en tiempo real en las estaciones de carga.

En el trabajo realizado por Silber et al. (2024) [5], se demostró el potencial para reducir los factores de simultaneidad de los sitios de carga rápida. Adoptarlo ayudaría en la expansión de la infraestructura de carga de vehículos eléctricos, ya que, por un lado, significa que los transformadores pueden ser más pequeños y, consecuentemente, la demanda de energía de los operadores de red en los puntos de conexión puede ser menor. Esto ayuda en la transmisión de energía y no reserva una cantidad máxima de energía raramente necesaria, lo que ralentizaría la construcción de otros sitios de carga. Por otro lado, los operadores de puntos de carga pueden ahorrar recursos al diseñar sitios de carga rápida y, por lo tanto, pueden construir más. Esto aumentaría las oportunidades de carga para los conductores y ayudaría a la aceptación de la electromovilidad en general. Sin embargo, el umbral de aceptación α se estableció en 0.01, lo que influye en los resultados de *OPT*. En investigaciones futuras, se deberían investigar los valores aceptables para α para posiblemente aumentar el umbral y disminuir aún más *OPT*. Además, los avances tecnológicos en el diseño de estaciones de carga podrían integrarse, así como mejoras en las curvas de carga de vehículos eléctricos. Esto revelaría qué tan estables resultan ser los resultados de este trabajo ante la tecnología cambiante. Por último, el análisis sobre la integración de sistemas de almacenamiento de energía eléctrica (BESS) puede llevarse a cabo con más detalle dadas los resultados

negativos de esta publicación. Cambiar las tasas de carga y las capacidades, así como reducir los precios, por ejemplo, para el uso en segunda vida, puede llevar a soluciones económicamente viables.

Por otra parte, en la investigación realizada por Chen, et al. (2023) [6], se establece un marco de coordinación P2P jerárquico para estaciones de carga de vehículos eléctricos (EVCS) equipadas con tecnología de respuesta a la demanda (DRE) y sistemas de almacenamiento de energía eléctrica (BESS), donde la transacción P2P entre múltiples EVCSs se coordina con la operación de redes de transporte urbano (UTN) y la operación de distribución de energía (PDN). Se desarrolla un esquema de transacción P2P completamente descentralizado para múltiples EVCSs, que también incorpora la protección de la privacidad y el costo comercial de las transacciones de energía (TE). Los resultados muestran que, primero, los resultados de asignación de carga de los vehículos eléctricos (EVs) pueden estar fácilmente disponibles para cada agente EVCS después de resolver el problema de asignación de transporte de energía múltiple (MTAP) por el operador del sistema de transporte (TSO). La plataforma P2P permite transacciones de TE entre EVCSs, donde la compra de energía de la red se reduce en casi un 25.44 %, de 6161.7 kWh a 4,594 kWh. La energía intercambiada entre EVCSs en el proceso de negociación P2P es de hasta 2,583.7 kWh, lo que permite una reducción del 16.61 % en el costo de operación de EVCS en comparación con el funcionamiento independiente. Finalmente, se señala que algunos casos valiosos indican que la reconfiguración de la PDN puede mejorar el espacio factible de transacciones P2P entre EVCSs bajo las restricciones de operación del DSO. Además, se investiga el impacto de la congestión del tráfico de UTN y se valida la escalabilidad del marco propuesto. En el sistema de 123 buses de IEEE, donde hay más entidades comerciales y mayores cantidades de transacción, se logra una reducción del 39.4 % en el costo de operación de EVCS.

En el trabajo realizado por Moghaddam et al. (2017) [7], proponen una estrategia de carga inteligente para una red de vehículos eléctricos enchufables, la cual ofrece múltiples opciones de carga en las estaciones de carga. Consideran que las diferentes restricciones presentadas como lo pueden ser la distancia, tiempo, etc., son importantes de considerar para que el usuario del vehículo pueda recargar en un tiempo mínimo y llegar a su destino sin retrasos significativos. Como método de solución utilizan un método de colas para estimar el retraso en varias estaciones de carga y para evitar los tiempos de espera largos y la posible superposición entre periodos pico de carga introdujeron el concepto de carga parcial. Los resultados reportados muestran que en la simulación reducen significativamente el retraso medio en la carga (hasta un 25 %) y el costo (hasta un 15 %). Como trabajo futuro proponen investigar con un modelo de precios dinámico óptimo para minimizar la superposición entre el PEV pico y el residencial.

En el contexto del diseño de rutas para estaciones de carga de vehículos eléctricos (EV), la planificación y optimización de infraestructuras de carga es un desafío complejo que implica múltiples variables y restricciones. Las metaheurísticas ofrecen una serie de ventajas que las hacen particularmente adecuadas

para abordar este problema.

MARCO TEÓRICO

Electric Vehicle Charging Scheduling

El problema de la programación de carga de vehículos eléctricos o Electric Vehicle Charging Scheduling (EVCS), que tiene como objetivo determinar es encontrar una estación de carga a lo largo del camino de modo que el tiempo total de viaje (incluido el tiempo de conducción desde la ubicación actual hasta el destino, el tiempo de espera en una estación de carga y el tiempo de carga) y el costo de carga sean minimizado [7]. De Acuerdo con Moghaddam et al. (2017) [7], este problema se formula de acuerdo con Ec. (1):

$$T^{\{drive\}}(s, dst) = \frac{D(s, j)}{v(s, j)} + \frac{D(j, dst)}{v(j, dst)} \quad (1)$$

donde:

- $D(s, j)$ y $D(j, dst)$ indican la distancia de la ruta desde el origen hasta la estación de carga y desde la estación de carga hasta el destino, respectivamente.
- $v(s, j)$ y $v(j, dst)$ indican la velocidad de movimiento promedio del PEV desde su origen hasta la estación de carga y desde la estación de carga hasta el destino, respectivamente.

El costo de carga de cada vehículo eléctrico enchufable (PEV) depende de su opción de carga asociada y de la tarifa de carga asociada en la estación j . Para un PEV, el costo de carga se calcula mediante la Ec. (2):

$$C_j^{recharge}(o) = (d_{charging} Pr_k) j \text{ en } j \quad (2)$$

donde $d_{charging}$ es la demanda de carga de cada PEV, que depende del SoC restante y objetivo para cada PEV (Ec. (3)):

$$d_{charging} = \sigma E - SoC \quad (3)$$

Aquí, σ es el coeficiente de carga parcial, que puede variar desde el SoC actual hasta el SoC máximo (por ejemplo, 90 % de la capacidad de la batería). En circunstancias normales, la mayoría de los usuarios preferirían una carga completa (es decir, $\sigma = 0,9$).

Heurísticas de perturbación

Una definición simple dada por Singh y Choudhary (2021) [8] es "*El enfoque heurístico se basa en prueba y error para encontrar o descubrir soluciones a los problemas*". En metaheurística se utilizan con mayor frecuencia dos tipos de heurísticas: construcción y perturbación. De acuerdo con lo mencionado anteriormente, se definen las heurísticas constructivas y perturbativas utilizadas en este trabajo. Según Laporte y F. Semet [9] dos técnicas principales para construir soluciones al problema del viajante (TSP) son la heurística codiciosa o aleatoria y la definición formal de TSP se puede encontrar en lo publicado por Jünger et al. (1995) [10].

En esta investigación se eligió la heurística aleatoria para realizar las soluciones iniciales (ver Algoritmo 1) y cinco heurísticas perturbativas descritas en los Algoritmos 1, 2, 3, 4, y 5.

Algoritmo 1: Algoritmo de construcción de solución de TSP

Require: $n \geq 0$, $FF \leftarrow \text{Fitness Function}$

1: $S \leftarrow \text{empty solution}$

2: $\text{route} \leftarrow \text{empty list of size } n$

3: **Hacer desde** $i \leftarrow 0$ **hasta** $n - 1$ **hacer**

4: $\text{route}[i] \leftarrow \text{random value between } 0 \text{ and } n - 1$

5: **Fin**

6: $S \leftarrow \text{set}(\text{route})$

7: $S \leftarrow \text{evaluateSolution}(S, FF)$

8: **Regresar** S

Algoritmo 2: Heurística Two Swap (h_1)

Requiere: $n \geq 1$, $SI \leftarrow \text{Initial Solution}$, $FF \leftarrow \text{Fitness Function}$

1: $\text{solutionB} \leftarrow \text{copy}(SI)$

2: $\text{index} \leftarrow \text{random}(0, n - 1)$

3: $\text{index2} \leftarrow \text{random}(0, n - 1)$

4: $\text{value} \leftarrow \text{solutionB}[\text{index2}]$

5: $\text{solutionB}[\text{index2}] \leftarrow \text{solutionB}[\text{index}]$

6: $\text{solutionB}[\text{index}] \leftarrow \text{value}$

7: $\text{solutionB} \leftarrow \text{evaluateSolution}(\text{solutionB}, FF)$

8: **Regresar** $\text{criteria.toReplace}(SI, \text{solutionB})$

Algoritmo 3: heurística Crossover (h_2)

Requiere: $n \geq 1$, $solutionA \leftarrow Initial\ Solution$, $FF \leftarrow Fitness\ Function$

```
1:  $solutionB \leftarrow copy(solutionA)$ 
2:  $parent1 \leftarrow solutionA.body$ 
3:  $n \leftarrow len(parent1)$ 
4:  $parent2 \leftarrow sample(range(0, n), n)$ 
5:  $k \leftarrow random(0, n - 1)$ 
6:  $Index \leftarrow []$ 
7: For{ $i$  in  $range(n)$ }
8: If{ $i \neq k$ }
9:  $Index.append(i)$ 
10: EndIf
11: EndFor
12:  $son \leftarrow []$ 
13: For{ $i$  in  $range(n)$ }
14: If{ $i \geq n - len(Index)$ }
15:  $son.append(parent1[Index[i]])$ 
16: Else
17:  $son.append(-1)$ 
18: EndIf
19: EndFor
20:  $if \leftarrow []$ 
21: For{ $elem$  in  $parent2$ }
22: If{ $elem$  not in  $son$ }
23:  $f.append(elem)$ 
24: EndIf
25: EndFor
26: For{ $i$  in  $range(n)$ }
27: If{ $son[i] == -1$ }
28:  $son[i] \leftarrow f.pop(0)$ 
29: EndIf
30: EndFor
31:  $solutionB \leftarrow set(son)$ 
32:  $solutionB \leftarrow evaluateSolution(solutionB, FF)$ 
33: Regresar  $criteria.toReplace(SI, solutionB)$ 
```

Algoritmo 4: Heurística Nearest Neighbor (h_3)

Requiere: $n \geq 1$, $solutionA \leftarrow Initial\ Solution$, $FF \leftarrow Fitness\ Function$ $matrix \leftarrow cost\ Matrix$

```

1:  $solutionB \leftarrow copy.deepcopy(solutionA)$ 
2:  $parent1 \leftarrow solutionA.body$ 
3:  $c \leftarrow random.randint(0, len(parent1) - 1)$ 
4:  $ic \leftarrow parent1.index(c)$ 
5:  $rowt \leftarrow matrix[c]$ 
6:  $rowOrderedWithIndex \leftarrow []$ 
7: For{ $i$ ,  $value$  in  $enumerate(rowt)$ }
8:  $rowOrderedWithIndex.append((i, value))$ 
9: EndFor
10:  $rowOrderedWithIndex.sort(rowt)$ 
11:  $orderedIndex \leftarrow []$ 
12: For{ $index$ , in  $rowOrderedWithIndex$ }
13:  $orderedIndex.append(index)$ 
14: EndFor
15:  $nc \leftarrow orderedIndex[1]$ 
16: For{ $i$ , in  $enumerate(parent1)$ }
17: If{ $parent1[i] == nc$ }
18:  $inc \leftarrow i$ 
19: break
20: EndIf
21: EndFor
22:  $c1 \leftarrow parent1[ic + 1]$ 
23:  $parent1[ic + 1] \leftarrow nc$ 
24:  $parent1[inc] \leftarrow c1$ 
25:  $solutionB \leftarrow set(son)$ 
26:  $solutionB \leftarrow evaluateSolution(solutionB, FF)$ 
27: Regresar  $criteria.toReplace(SI, solutionB)$ 

```

Algoritmo 5: Heurística K-OPT (h_4 , h_5)

Requiere: $n \geq 1$, $solutionA \leftarrow Initial\ Solution$, $FF \leftarrow Fitness\ Function$

```
1:  $solutionB \leftarrow copy(solutionA)$ 
2:  $son \leftarrow solutionB.body$ 
3:  $ind \leftarrow random.sample(len(son), n)$ 
4:  $random.shuffle(ind)$ 
5:  $sonCopy \leftarrow son[:]$ 
6: For{ $i$  in  $range(self.n)$ }
7:  $son[ind[i]] \leftarrow sonCopy[ind[(i + 1) \% n]]$ 
8: EndFor
9:  $solutionB \leftarrow set(son)$ 
10:  $solutionB \leftarrow evaluateSolution(solutionB, FF)$ 
11: Regresar  $criteria.toReplace(SI, solutionB)$ 
```

Metaheurística de Búsqueda Local Iterada

En la experimentación se utilizó el algoritmo de búsqueda local iterada (ILS). Esta metaheurística fue propuesta por Lourenco et al. (2023) [11], que construyen o modifican una solución mediante una heurística incorporada. Las soluciones generadas son mejores que las generadas aleatoriamente o alteradas. Este algoritmo tiene como objetivo intensificar la solución inicial explorando sus vecinos. El algoritmo ILS se muestra en el Algoritmo 6, tomado de Talbi (1965) [12].

METODOLOGÍA

La metodología de este trabajo se basa en la propuesta por Ortiz-Aguilar et al. [13] y se puede resumir en los siguientes pasos o etapas, cada uno de los cuales se describe detalladamente a continuación:

I. Diseño del insumo

■ Modelado de Restricciones y Variables:

Matriz de Distancias (MMA): Se recopiló la ubicación geográfica de las estaciones de gasolina existentes en el estado de Guanajuato. Utilizando Python y Google Maps, se obtuvo la longitud y la latitud de cada estación para generar una matriz de distancias entre todas las estaciones. Esta matriz es fundamental para calcular las rutas óptimas.

Parámetros de Carga: Se consideraron variables como la capacidad de carga de las estaciones, las tarifas de carga, y las restricciones de tiempo. Estas variables se integraron en el modelo para asegurar que las soluciones sean realistas y aplicables en un contexto práctico.

Algoritmo 6: Iterated Local Search (ILS)

Requiere: $inSol \leftarrow Initial\ Solution$, $maxI \leftarrow MaxIterations$

```

1:  $bestSol \leftarrow inSol$ 
2:  $currentSol \leftarrow inSol$ 
3:  $iteration \leftarrow 0$ 
4: While{ $iteration < maxI$ }
5:  $perturbedS \leftarrow perturb(currentSol)$ 
6:  $localOptima \leftarrow localSearch(perturbedSol)$ 
7: If{ $isImprovement(localOptima, bestSol)$ }
8:  $bestSol \leftarrow localOptima$ 
9: EndIf
10:  $currentSol \leftarrow diversify(localOptima)$ 
11:  $iteration \leftarrow iteration + 1$ 
12: EndWhile
13: Regresar  $bestSol$ 

```

II. Diseño Metodológico de los Experimentos

Selección de técnicas metaheurísticas y parámetros

- **Metaheurísticas Utilizadas:** Se seleccionó la búsqueda local iterada (ILS) para resolver el problema. La elección de esta técnica se basa en su capacidad comprobada para manejar problemas de optimización complejos y su flexibilidad para adaptarse a diferentes condiciones.

- **Heurísticas Constructivas y Perturbativas:** Para generar soluciones iniciales y mejorar las existentes, se emplearon varias heurísticas, como:

Heurística Aleatoria: Para generar soluciones iniciales aleatorias.

Heurística Two-Swap : Intercambio de dos elementos en la solución.

- **Heurística Crossover:** Combinación de soluciones parentales para generar nuevas soluciones.

- Heurística Nearest Neighbor: Selección del vecino más cercano para construir rutas.
- Heurística K-OPT: Reordenamiento de sub-rutas para mejorar la solución.

III. Experimentación

Configuración y ejecución de experimentos

La implementación se realizó utilizando Python y las condiciones de experimento. Cada algoritmo fue probado bajo condiciones controladas con un mismo límite de llamadas a función por instancia. Los experimentos se repitieron 31 veces para asegurar la consistencia y fiabilidad de los resultados.

Estadísticas Utilizadas: Para el análisis estadístico de los resultados, se aplicaron pruebas como Shapiro-Wilks para verificar la normalidad de los datos. Dado que los datos no seguían una distribución normal, se utilizó la mediana como medida representativa. Además, se emplearon pruebas de Friedman, Friedman Alineado y Quade para comparar el rendimiento de las heurísticas.

IV. Evaluación de resultados

Análisis de Resultados y Comparación

- Métricas de Desempeño: Los resultados se evaluaron en términos de distancia total recorrida y la eficiencia de las rutas generadas. Se compararon las heurísticas utilizando métricas de desempeño específicas para cada técnica metaheurística.
- Selección de Mejores Soluciones: Se identificaron las dos mejores heurísticas basadas en su desempeño.

RESULTADOS

Para este conjunto de casos, el enfoque se inclinó hacia el empleo de las estructuras *MMA* y *LPH*. Estos fueron seleccionados en función de su relevancia y aplicabilidad en el marco del problema, como se detalla en el trabajo Ortiz-Aguilar et al. (2023) [14].

A continuación, se describe la configuración de parámetros para cada Metaheurística y se explica las pruebas estadísticas utilizadas para comparar los experimentos. La experimentación implicó probar cada una de las instancias en una computadora portátil con las siguientes especificaciones: una CPU Intel® Core i3-1005G1 con frecuencia de 1,20 GHz, 8 GB de RAM y un sistema operativo Windows 11 de 64 bits. La metodología se implementó utilizando Python y los experimentos se repitieron 31 veces. Cada experimento heurístico tenía un límite estipulado de 100.000 llamadas a funciones por instancia.

Para el análisis estadístico, primero se aplicó la prueba estadística de Shapiro-Wilks para comprobar si los datos tenían una distribución normal, y los resultados indicaron que los datos no seguían una distribución normal; por lo tanto, se eligió la mediana como estadísticamente representativa.

Instancias

Como punto de partida para hacer una propuesta de solución de dónde ubicar posibles puntos de carga de vehículos eléctricos fue tomar la información de donde se encuentran actualmente las gasolineras en México. Por lo cual se consultó el informe de estaciones de servicio y estaciones de autoconsumo franquiciadas por Petróleos Mexicanos (PEMEX) obtenida del Directorio de Estaciones de Servicio [15], en la Tabla 1 se muestra el resumen de gasolineras por estado que están activas, siendo un total de 6754.

De las instancias para este trabajo se decidió tomar primero las del estado de Guanajuato para hacer el análisis. La información obtenida contenía solamente la dirección de cada estación, por lo que era necesario sacar una coordenada en términos de longitud y latitud para poder procesarla y generar una matriz de costos (MMA).

Tabla 1. Cantidad de estaciones de gasolina por Estado en México.

Estado	No	Estado	No
Campeche	77	Oaxaca	111
Chiapas	155	Puebla	286
Chihuahua	215	Querétaro	196
Ciudad De México	297	Quintana Roo	79
Coahuila	237	San Luis Potosí	148
Colima	45	Sinaloa	220

Tabla 1. Cantidad de estaciones de gasolina por Estado en México (...continuada).

Estado	No	Estado	No
Durango	93	Sonora	248
Estado De México	670	Tabasco	169
Guanajuato	434	Tamaulipas	218
Guerrero	90	Tlaxcala	54
Hidalgo	148	Veracruz	448
Jalisco	461	Yucatán	177
Michoacán	187	Zacatecas	85

En una primera aproximación se supone que se tienen que pasar por todas las estaciones para tener un tiempo de recarga estándar en cada una de ellas, por lo cual a este punto es solamente importante la distancia total recorrida. En la Tabla 2 se muestran los resultados en términos de distancia de las heurísticas y se tomarán a las dos mejores para ser puestas en el algoritmo de búsqueda local con las diferentes heurísticas.

Tabla 2. Resultados de las heurísticas.

Estado	h_1	h_2	h_3	h_4	h_5
AGUASCALIENTES	2.89	6.61	4.53	2.91	3.14
BAJA CALIFORNIA	58.83	240.63	154.65	54.79	67.92
BAJA CALIFORNIA SUR	30.29	82.95	50.69	24.55	23.14
CAMPECHE	16.08	58.20	32.41	18.38	16.00
CHIAPAS	34.37	146.32	93.64	38.51	44.64
CHIHUAHUA	73.09	306.76	205.33	87.18	95.65
COAHUILA	110.34	403.59	228.26	107.18	108.10
COLIMA	4.10	8.05	4.85	4.01	1.91
DURANGO	19.25	75.47	47.73	22.22	24.01
ESTADO DE MEXICO	78.67	293.35	196.33	99.81	119.00
GUERRERO	27.93	76.76	41.14	22.86	19.61
HIDALGO	26.26	71.62	45.93	23.63	24.47
JALISCO	93.74	372.71	242.09	117.48	143.48
MICHOACAN	45.78	170.11	105.03	55.40	55.63

Tabla 2. Resultados de las heurísticas (...continuada).

Estado	h_1	h_2	h_3	h_4	h_5
MORELOS	6.45	16.81	9.81	5.60	7.40
NAYARIT	45.53	68.78	56.93	36.06	35.17
NUEVO LEON	63.16	170.00	117.33	71.00	78.19
OAXACA	39.72	134.82	78.45	40.42	40.00
PUEBLA	39.54	144.52	91.72	48.21	54.09
QUERETARO	15.24	52.76	37.15	20.01	21.43

Por último, se aplicó una prueba de Friedman, Friedman Alineado y Quade, las cuales se utilizan frecuentemente para comparar algoritmos en Joaquín Derrac et al. [16], de dichas pruebas se muestran los resultados de la Tabla 3, de la cual se puede deducir lo siguiente:

- h_1 : Tiene valores de 1.75 en Friedman y Quade, y 596 en Friedman Alineado, sugiriendo un rendimiento relativamente bajo en comparación con otras heurísticas.
- h_2 : Con valores de 5, 1732, y 5, respectivamente, esta heurística muestra un rendimiento significativamente diferente, probablemente peor o mejor que las otras dependiendo del contexto del experimento.
- h_3 : Valores de 4, 1488, y 4, respectivamente, indicando un rendimiento intermedio.
- h_4 : Con valores de 1.9, 600, y 1.9, muestra un rendimiento similar al de h_1 .
- h_5 : Valores de 2.35, 634, y 2.35, también sugieren un rendimiento cercano al de h_1 y h_4 .

Tabla 3. Estadísticos de las pruebas de Friedman, Friedman Alineado y Quade.

Prueba	h_1	h_2	h_3	h_4	h_5	Estadístico
Friedman	1.75	5	4	1.9	2.35	65.56
Friedman Alineado	596	1732	1488	600	634	62.1871415
Quade	1.75	5	4	1.9	2.35	30.7795386

La mejor heurística de acuerdo con las pruebas, el iterado con los valores más bajos es la heurística de exploración h_1 y, por lo tanto, es el que tiene mejor rendimiento.

CONCLUSIONES

Este estudio presenta una comparativa de diversas heurísticas aplicadas al diseño de rutas para estaciones de carga de vehículos eléctricos. Los resultados obtenidos demuestran que los enfoques metaheurísticos, como la búsqueda local iterada, ofrece una solución inicial al dicho problema. Sin embargo, es importante destacar que esta investigación se enfoca exclusivamente en la comparación de diferentes heurísticas y sus aplicaciones, sin pretender establecer una solución definitiva al problema planteado.

En resumen, aunque esta investigación proporciona una base sólida para el diseño de rutas para estaciones de carga de vehículos eléctricos utilizando algoritmos metaheurísticos, es evidente que se necesita un enfoque multidisciplinario y colaborativo para abordar las limitaciones y desafíos identificados. Futuros trabajos deben enfocarse en mejorar la precisión de los modelos, integrar tecnologías avanzadas, y considerar más profundamente los factores humanos y económicos para optimizar la infraestructura de carga de EV de manera holística.

REFERENCIAS

- [1] L. de M. O. Aguilar, J. M. C. Valadez, H. J. P. Soberanes, C. L. D. González, C. L. Ramirez, and J. Soria-Alcaraz. Comparativa de algoritmos bioinspirados aplicados al problema de calendarización de horarios. *Research in Computing Science*, (94), pp. 33–43, 2015.
- [2] Inventario Nacional de Emisiones de Gases y Compuestos de Efecto Invernadero, 1990-2019.
- [3] N. Rauh, T. Franke, and J. F. Krems. Understanding the impact of electric vehicle driving experience on range anxiety. *Hum Factors*, vol. 57, no. 1, pp. 177–187, 2015.
- [4] S. P. Rajawat, A. Gautam, S. J. Janarthanan, L. Soni, and P. Khandekar. Adaptive EV Range Estimation and Optimization Based on Rider Demand and Terrain Requirements, 2024.
- [5] F. Silber, S. Scheubner, and A. März. Analysis of the simultaneity factor of fast-charging sites using Monte-Carlo simulation. *International Journal of Electrical Power & Energy Systems*, vol. 155, p. 109540, 2024.
- [6] X. Chen, X. Wang, M. Shahidehpour, L. Affolabi, Z. Lu, and K. Li. Distributed Peer-to-Peer Coordination of Hierarchical Three-Phase Energy Transactions Among Electric Vehicle Charging Stations in Constrained Power Distribution and Urban Transportation Networks. *IEEE Transactions on Transportation Electrification*, 2023.

- [7] Z. Moghaddam, I. Ahmad, D. Habibi, and Q. V. Phung. Smart charging strategy for electric vehicle charging stations. *IEEE Transactions on Transportation Electrification*, vol. 4, no. 1, pp. 76–88, 2017.
- [8] S. K. Singh Padam and Choudhary. Introduction: Optimization and Metaheuristics Algorithms. In *Metaheuristic and Evolutionary Computation: Algorithms and Applications*, A. and J. P. and A. S. and B. F. I. Malik Hasmat and Iqbal, Ed., Singapore: Springer Singapore, 2021, pp. 3–33.
- [9] G. Laporte and F. Semet. Classical heuristics for the capacitated VRP. In *The vehicle routing problem*, SIAM, 2002, pp. 109–128.
- [10] M. Jünger, G. Reinelt, and G. Rinaldi. The traveling salesman problem. *Handbooks in operations research and management science*, vol. 7, pp. 225–330, 1995.
- [11] L. Helena, M. Olivier, and S. Thomas. Iterated Local Search. In *Handbook of Metaheuristics*, vol. 57, F. Glover, G. Kochenberger, and F. S. Hillier, Eds., International Series in Operations Research & Management Science, vol. 57. Springer New York, 2003, pp. 320–353.
- [12] E. G. Talbi. *Metaheuristic from Design to Implementation*. Wiley, 1965.
- [13] L. de M. Ortiz-Aguilar, M. Carpio, H. Puga, J. A. Soria-Alcaraz, M. Ornelas-Rodríguez, and C. Lino. Increase methodology of design of course timetabling problem for students, classrooms, and teachers. In *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 713–728, 2017.
- [14] L. Ortiz-Aguilar, H.-A. Yeovanna, M. Benitez, S. Rodriguez-Miranda, and F. Mendoza-Vazquez. A Comparison Between Selection Operators Heuristics of Perturbation in CSP. In *Hybrid Intelligent Systems Based on Extensions of Fuzzy Logic, Neural Networks and Metaheuristics*, Springer Nature Switzerland Cham, 2023, pp. 365–377.
- [15] Directorio de Estaciones de Servicio. Consultado: el 19 de junio de 2024. [En línea]. Disponible en: https://491231.fs1.hubspotusercontent-na1.net/hubfs/491231/site_edenred_mx/wp/Directorio-Estaciones-Servicio-TC-0722.pdf
- [16] S. G. Joaquín Derrac. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput*, 2011.