



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

**TOMA DE DECISIONES PARA FLOTAS DE VEHÍCULOS ELÉCTRICOS
SUJETOS A INCERTEZA, BASADA EN ESCENARIOS Y ALGORITMOS
GENÉTICOS**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERA CIVIL ELÉCTRICA

DANIELA ALEJANDRA ARAVENA CABRERA

PROFESOR GUÍA:
DIEGO MUÑOZ CARPINTERO

PROFESOR CO-GUÍA:
JUAN-PABLO FUTALEF GALLARDO

COMISIÓN:
FRANCISCO RIVERA SERRANO

SANTIAGO DE CHILE
2023

TOMA DE DECISIONES PARA FLOTAS DE VEHÍCULOS ELÉCTRICOS SUJETOS A INCERTEZA, BASADA EN ESCENARIOS Y ALGORITMOS GENÉTICOS

El uso de vehículos eléctricos (EV) se ha masificado, pues son una alternativa sustentable a los vehículos convencionales. Sin embargo, su despliegue generalizado es aún difícil debido a restricciones impuestas por sus baterías, la escasa infraestructura de recarga aún presente en muchas ciudades y la incerteza intrínseca del tráfico. En esta memoria, se desarrolla una estrategia para resolver el problema de ruteo de vehículos eléctricos (E-VRP), considerando la incerteza asociada a la velocidad, impactando en los tiempos de viaje y consumos energéticos. El E-VRP se formula como un problema de optimización estocástica no lineal. La estocasticidad se considera presente en la velocidad y se aborda usando escenarios, los que se obtienen a partir del muestreo de distribuciones conocidas. El E-VRP es resuelto mediante algoritmos genéticos (GA) con una heurística que reduce el costo computacional asociado a la propagación de incertezas. La estrategia es comparada con una variante determinística del E-VRP. Los resultados muestran que la estrategia estocástica permite manejar la probabilidad con la que se satisfacen las restricciones probabilísticas del E-VRP, no así en el caso determinístico. Finalmente, el método de solución basado en GA permite resolver el E-VRP estocástico y aumentar la robustez de la solución.

*A mi hermana Ignacia,
el amor de mi vida.*

*A mi prima Ayleen, mi
compañera de infancia,
adolescencia y adultez.*

Tabla de Contenido

1. Introducción	1
1.1. Contexto y Motivación	1
1.2. Objetivos	3
1.2.1. Objetivo General	3
1.2.2. Objetivos Específicos	3
2. Marco teórico y Estado del Arte	4
2.1. Introducción	4
2.2. Marco Teórico	4
2.2.1. Algoritmos Genéticos	4
2.2.1.1. Métodos de Codificación de Individuos	5
2.2.1.2. Estructura	5
2.2.1.3. GA genérico	6
2.2.1.4. Aplicaciones de GA al E-VRP	7
2.2.2. Optimización Estocástica	8
2.2.2.1. Propagación de Incertidumbres mediante el Método de Escenarios	9
2.3. Estado del Arte	10
2.4. Discusión	12
3. Formulación de la Optimización basada en Escenarios para el Ruteo de Vehículos Eléctricos	14
3.1. Introducción	14
3.2. Formulación del problema	14
3.3. Modelación de la dinámica de los vehículos	16
3.4. Formulación de la toma de decisiones estocástica para el ruteo de EV	20
3.4.1. Obtención de escenarios	20
3.4.2. Codificación de Individuos	21
3.4.3. Población Inicial	22
3.4.4. Evaluación de Fitness y Manejo de Restricciones	23
3.4.4.1. Evaluación Determinística y Estocástica del Fitness	23
3.4.4.2. Penalizaciones	24
3.4.4.3. Costos operacionales	26
3.4.5. Algoritmo Genético Implementado	27
4. Estudios por Simulación del Modelo Desarrollado	29
4.1. Configuración Experimental	29
4.1.1. Pesos de la Función de Costos	30

4.2. Estudios realizados	30
4.2.1. Cumplimiento de Restricciones Estocásticas	32
4.2.2. Cantidad de Individuos Evaluados Estocásticamente	35
4.2.3. Comparación con la Formulación Determinística	38
5. Conclusiones y Trabajo Futuro	41
Bibliografía	43
Anexos	46
A. Ejemplo de codificación	46
B. Operaciones de crossover y mutación	47

Índice de Tablas

3.1.	Descripción y valor de constantes utilizadas para calcular el consumo de energía entre cada par de nodos.	17
4.1.	Constantes utilizadas en las restricciones del problema [3].	30
4.2.	Pesos de la función de costos.	30
4.3.	Valores hiperparámetros del problema	31
4.4.	Métricas al utilizar distintos métodos para calcular la cantidad de escenarios a evaluar estocásticamente.	36

Índice de Ilustraciones

2.1.	Modelo gráfico de un E-VRP genérico	11
3.1.	Ejemplo de grafo dirigido.	15
3.2.	Codificación de individuos. Elaboración propia	22
4.1.	Instancia utilizada para ejecutar el GA.	29
4.2.	Comparación de la velocidad de viaje de la instancia y de los escenarios, entre los nodos 0 y 17, a distintas horas del día.	31
4.3.	Ejemplo de ruta factible.	32
4.4.	Vista en detalle de algunas ventanas de tiempo de clientes.	33
4.5.	Vista en detalle del SOC de la batería al terminar el recorrido.	34
4.6.	Ejemplos de soluciones factibles, al aumentar el número de escenarios muestreados	34
4.7.	Comparación de costo operacional y tiempo computacional entre los distintos métodos utilizados.	36
4.8.	Porcentaje de individuos evaluados estocásticamente a través de las generaciones del GA, con los métodos mencionados, para $H = 6$	37
4.9.	Aplicación de escenarios a la solución determinística obtenida.	39

Capítulo 1

Introducción

1.1. Contexto y Motivación

El uso de vehículos eléctricos (EV, por sus siglas en inglés) se ha visto masificado en los últimos años, pues han surgido como una alternativa ambientalmente amigable al compararlos con aquellos vehículos de combustión interna. Según datos del Ministerio de Energía, el sector del transporte en Chile es responsable del 24 % de las emisiones totales de dióxido de carbono [1], por lo que es imperioso transitar hacia el uso de tecnologías que disminuyan este porcentaje de emisiones en la industria del transporte. Entre las ventajas de los EV, se tiene que estos no producen emisiones de gases ni partículas al estar en movimiento, impactando directamente en mejorar la calidad de aire en las ciudades y contribuyendo a combatir el cambio climático. Aún más, los EV gastan en torno a un quinto de la energía que uno de combustión interna para una misma distancia y una misma condición de carga, lo que se traduce en un ahorro económico en la operación [2].

Por otra parte, el problema de ruteo de vehículos (VRP, por sus siglas en inglés) pretende encontrar aquellas rutas con un menor costo, para que una flota de vehículos atienda a un conjunto determinado de clientes, bajo ciertas restricciones operacionales. En resumen, un conjunto de nodos representa las posiciones geográficas de los clientes y un depósito. Los arcos que conectan estos nodos tienen costos que representan el gasto de un vehículo cuando viaja de un nodo a otro y cada cliente tiene una demanda que debe ser satisfecha por una flota de vehículos [3]. Los primeros estudios de este modelamiento datan de 1959, cuando Dantzig y Ramser abordaron por primera vez el problema de despacho de camiones homogéneos para atender estaciones de servicio. Posteriormente en 1964, Clarke y Wright, abordaron el problema que consistía en atender un número de clientes geográficamente dispersos, por medio de una flota de vehículos con capacidades heterogéneas [4]. Algunos de los entornos reales para los que el estudio del VRP es de gran importancia, corresponden a problemas de logística reversa en recolección y entrega de contenedores para la distribución de sangre del American Red Cross [5], en la recolección y reciclaje de desperdicios de aceites lubricantes [6], en el reparto de leche [7], entre otros.

El problema de ruteo de vehículos eléctricos (E-VRP, por sus siglas en inglés) es una variante del VRP en donde todos los vehículos pertenecientes a la flota son eléctricos. La principal diferencia entre el VRP clásico y el E-VRP, es que este último debería permitir que los EV se desvíen hacia las estaciones de carga para recargar su batería si no hay suficiente

energía disponible. A diferencia de un proceso de recarga de combustible, recargar batería afecta mayormente la operación debido a su larga duración. Como resultado, la mayoría de las variantes VRP clásicas y sus métodos de solución correspondientes no son adecuados en un contexto EV [3].

Si bien, decenas de trabajos han intentado resolver el E-VRP [8], aún es necesario explorar nuevos métodos de solución y mejorar las formulaciones para abordar elementos más realistas, que consideren: condiciones dinámicas de tráfico, incertidumbre en los tiempos de viaje, integración de funciones de carga no lineales, estimación precisa del consumo de energía, técnicas de enrutamiento adaptativo y capacidades de las estaciones de carga [3]. En este contexto, se han desarrollado trabajos que consideran las no linealidades en la formulación del problema, sin embargo, son pocos autores los que reflexionan sobre la estocasticidad de las variables en él, debido a que de por sí el E-VRP ya es un problema complicado (en efecto, es de clase de complejidad NP-hard) [8].

El presente trabajo corresponde a una extensión de la formulación planteada en [3] -en la que se considera funciones de carga no lineal, degradación de baterías y capacidad limitada en las estaciones de carga- incorporando la estocasticidad del tráfico en la formulación. Por lo tanto, la complejidad de la solución aumenta, pues se hace necesaria la utilización de alguna metodología para propagar la incertidumbre de estas variables. Uno de los enfoques que se pueden utilizar corresponde al método de escenarios. Este se basa en encontrar la solución óptima bajo un número finito de resultados muestreados de la incertidumbre, a los que se les denomina “escenarios” [9]. Así, este método obtiene muestras aleatorias independientes e idénticamente distribuidas (i.i.d.) de la variable estocástica, para luego trabajar el problema de optimización original con cada una de estas muestras por separado. Con el objetivo de abordar el costo computacional que este método implica, se puede utilizar alguna heurística que no aumente de manera considerable la complejidad del algoritmo.

1.2. Objetivos

1.2.1. Objetivo General

El objetivo general de este trabajo de memoria consiste en implementar una estrategia para optimizar rutas para flotas de vehículos eléctricos, que considere la estocasticidad del tráfico y consumo energético de los vehículos y que sea capaz de manejar la incerteza del problema, sin aumentar en exceso el costo computacional.

1.2.2. Objetivos Específicos

Para lograr el objetivo general, se espera abordar los siguientes objetivos específicos:

1. Estudiar e implementar la propagación de incertidumbre de variables dinámicas en la ruta de un vehículo eléctrico utilizando el método de escenarios.
2. Diseñar e implementar algoritmos genéticos con una heurística que permita manejar el elevado costo computacional asociado a la propagación de la incerteza.
3. Analizar el desempeño del método implementado para el problema de ruteo de vehículos eléctricos y verificar que esta estrategia permite aumentar la robustez de la solución.

Capítulo 2

Marco teórico y Estado del Arte

2.1. Introducción

Este capítulo proporciona una revisión de los elementos claves relacionados con el método de solución E-VRP planteado. En primer lugar, se hace un estudio del marco teórico de conceptos como los algoritmos genéticos, su importancia en problemas de optimización, su estructura y algunas aplicaciones al E-VRP; y de la propagación de incertidumbres mediante el método de escenarios. Además, se hace una revisión en cuánto al estado del arte del E-VRP, considerando formulaciones determinísticas y estocásticas, junto con los métodos de solución utilizados.

2.2. Marco Teórico

2.2.1. Algoritmos Genéticos

Los algoritmos genéticos (GA, por sus siglas en inglés) son métodos de búsqueda heurística, aplicables a una amplia gama de problemas de optimización, es decir, que se utilizan para encontrar soluciones óptimas a problemas que minimizan o maximizan una función en particular [10]. El GA clásico se basa en un conjunto de soluciones candidatas que representan una solución al problema de optimización que se quiere resolver. Así, una solución es un candidato potencial para un óptimo del problema de optimización [11].

Estos algoritmos, pertenecen a una clase en particular de los Algoritmos Evolutivos (EA, por sus siglas en inglés), que usan técnicas inspiradas en la evolución biológica de los seres vivos, imitando así procesos biológicos de reproducción y selección natural para encontrar las soluciones más aptas [12]. Los EA se usan para resolver problemas que no tienen una solución eficiente bien definida. Además, estos algoritmos son mucho más poderosos y eficientes que los algoritmos de búsqueda aleatoria y de búsqueda exhaustiva, pues no requieren información adicional sobre el problema dado [10].

La evolución es la base del GA. Las especies son capaces de adaptarse a su entorno, pues han desarrollado estructuras complejas que les permiten la supervivencia en diferentes tipos de ambientes. El apareamiento y lograr la evolución de la descendencia son los principios fundamentales del éxito de la evolución. Estas son buenas razones para adaptar los principios evolutivos para resolver problemas de optimización [11].

2.2.1.1. Métodos de Codificación de Individuos

En el contexto GA, una población es una colección de varios individuos. Un individuo codifica una solución candidata al problema de optimización, es decir, el individuo es una representación indirecta de la solución, no la solución misma [3]. A continuación se presentan algunos de los métodos mas usados para codificar los individuos de la población de un GA [13].

Codificación Binaria

En esta codificación, cada individuo se representa mediante un vector binario que contiene solo ceros y unos. Generalmente, esta codificación se usa en el problema de la Mochila, en donde con un uno ("1") se representa la presencia de elementos, mientras que con un cero ("0") la ausencia de estos. De esta forma, un individuo I se representa como:

$$I = [0, 0, 1, 0, 1, 1]$$

Codificación de Permutación

En este caso, cada individuo se representa mediante un vector en donde cada elemento corresponde a la posición en una secuencia de elementos conocidos. Esta codificación se utiliza en problemas de ordenación. Por ejemplo, en el Problema del Vendedor Ambulante, la cadena de números representa la secuencia de ciudades visitadas por el vendedor. De esta forma, un individuo I se puede representar como:

$$I = [1, 5, 3, 2, 4]$$

Codificación de Valores

En la codificación de valores, cada individuo se representa como la cadena de algún valor. Los valores pueden ser cualquier elemento que esté relacionado con el problema: números de forma, números reales o caracteres para algunos objetos complicados. Esta codificación se usa generalmente en aplicaciones de redes neuronales, para encontrar los pesos de las redes. De esta forma, un individuo I se puede representar como:

$$I = [1.23, -2.12, 3.14, -0.34, 4.62]$$

2.2.1.2. Estructura

Un elemento esencial en la implementación de un GA, corresponde al *fitness*. Esta es una función que mide la calidad de las soluciones que ha generado el GA, es decir, aquellas soluciones que tienen mejor *fitness* corresponderán a los individuos con mayor probabilidad de ser elegidos. Por otro lado, el *fitness* de las soluciones no factibles se ve deteriorado debido a penalizaciones agregadas por no cumplir ciertas restricciones. Muchas veces hay casos en los que existen múltiples objetivos que deben optimizarse al mismo tiempo, por lo que los valores del *fitness* de cada objetivo individual se pueden agregar, por ejemplo, calculando la suma ponderada [11]. Tanto la elección de las penalizaciones, como la elección de los pesos apropiados en la optimización multiobjetivo, son propósitos de diseño del GA.

A continuación se presentan los tres operaciones principales que generan la nueva descendencia en el GA.

Selección

El procedimiento de selección se utiliza para determinar qué individuos son los que se reproducirán para generar la descendencia [10]. La probabilidad de que un individuo sea seleccionado depende de su *fitness* y, en general, es más probable que se elija a los individuos con mayor *fitness*. Por lo tanto, el procedimiento de selección tiene como objetivo permitir la convergencia hacia individuos más aptos, para mejorar la aptitud física promedio de la población en cada generación [3].

Uno de los operadores de selección más utilizados es el de selección de torneos, donde se selecciona aleatoriamente un conjunto de soluciones y dentro de este subconjunto de competencia, las mejores soluciones finalmente se escogen como nuevos padres. Este tipo de selección, ofrece una probabilidad positiva para que cada solución sobreviva, incluso si tiene peores valores de *fitness* que otras soluciones [11].

Crossover

La operación de *crossover* permite la combinación de material genético de dos o más soluciones padres para generar una o más soluciones hijas. Algunas investigaciones sugieren que es mejor usar más de dos padres para reproducir individuos de buena calidad [12]. Este operador se fundamenta en que ambos padres pueden representar partes exitosas de soluciones hijas que, cuando se combinan, incluso superan a sus padres. Para elegir qué soluciones padres participan en la generación de los hijos, muchos GA eligen aleatoriamente los padres a partir de distribuciones uniformes [11].

Mutación

El proceso de mutación altera aleatoriamente la estructura de los individuos [3], permitiendo así que se pueda explorar la zona alrededor de los individuos que se mutan. El diseño de este operador, debe cumplir con tres requisitos principalmente [11]: accesibilidad, lo que significa que debe haber una posibilidad mínima de llegar a cada parte del espacio de la solución, de modo que la probabilidad de que se pueda encontrar el óptimo sea positiva; imparcialidad, es decir, que el operador no debe inducir una desviación de la búsqueda hacia una dirección en particular; y escalabilidad, que suelen utilizarse en operadores de mutación que se basan en distribuciones de probabilidad.

El operador de mutación es un elemento fundamental en un GA, ya que los procesos de selección y *crossover* mantienen la información de aquellos individuos más aptos, pero estos solo son más aptos en relación con la generación actual, lo que puede hacer que el algoritmo converja demasiado rápido, atascándose en un óptimo local antes de encontrar el óptimo global. La mutación ayuda a mantener la diversidad en la población, pero también puede hacer que el algoritmo converja más lentamente [10].

2.2.1.3. GA genérico

En el Algoritmo 1 se presenta un pseudo-código que describe un GA genérico. Como se puede observar, al principio se inicializa un set de soluciones, denotando la población inicial, que preferiblemente debería cubrir todo el espacio de soluciones [11]. A continuación, esta población es evaluada, configurando un *fitness* para cada individuo. Luego, a través de las generaciones y utilizando los operadores de selección, *crossover* y mutación ya mencionados,

se genera una nueva descendencia de soluciones candidatas, las que nuevamente son evaluadas. Finalmente, dependiendo del *fitness* de los individuos, generados a partir de la evaluación de estos, se establece la población de la siguiente generación. Este procedimiento se realiza para todas las generaciones definidas, para luego retornar el mejor individuo de la población final, en base al valor del *fitness* de cada individuo.

Algoritmo 1 *Algoritmo Genético genérico* [14]

```

1:  $k \leftarrow 0$ ;
2:  $P(k) \leftarrow$  Población inicial;
3: evaluación( $P(k)$ )
4: while noTermine() do
5:    $P_0 \leftarrow$  selección( $P(k)$ );
6:    $P_1 \leftarrow$  crossover( $P(k)$ );
7:   mutación( $P_1$ );
8:   evaluación( $P_1$ );
9:    $P(k + 1) \leftarrow$  nuevaPoblación( $P_1, P(k)$ );
10:   $k \leftarrow k + 1$ ;
return mejorIndividuo( $P(k)$ )

```

Finalmente, el rendimiento de un algoritmo genético depende en gran medida de: el método utilizado para codificar las soluciones candidatas en los individuos, lo que la función de optimización está midiendo realmente, la probabilidad de cruce, la probabilidad de mutación, el tamaño de la población, el número de iteraciones, entre otros [10].

2.2.1.4. Aplicaciones de GA al E-VRP

Los GA se han implementado ampliamente en los VRP complejos y en sus diversas variantes, especialmente en redes de carreteras grandes y realistas [15]. La mayoría de los trabajos concluyen que se debe realizar un esfuerzo significativo en el diseño de las operaciones genéticas adecuadas. A continuación, se presenta una revisión de algunas de estas representaciones.

Como ya se ha mencionado, un GA consiste en una población que atraviesa un proceso de evolución, que incluye: la evaluación de los individuos, la selección de los padres, los procesos de *crossover* y mutación y el reemplazo de la antigua población por una nueva. Para acercar este algoritmo a la resolución de un VRP, se debe tener en cuenta que los aspectos importantes a considerar, corresponden a la gestión eficiente de la diversidad de la población y la función de *fitness* para comparar los individuos de la población [16].

En primer lugar, en [17] se busca optimizar una función multiobjetivo que considera los tiempos de viaje, el consumo de energía y los costos de cargar la batería. En este trabajo, se aplica un enfoque de programación difusa para obtener coeficientes de ponderación para cada objetivo, transformando así la función multiobjetivo en una de un solo objetivo. Entre los parámetros utilizados se destaca el uso de una brecha generacional de 0.9, probabilidad de *crossover* de 0.9 y un 0.5 de probabilidad de mutación. En una primera instancia se utilizan 100 generaciones, pero una evaluación posterior demostró que con 60 generaciones se obtienen mejores resultados. Finalmente, se observó que las compensaciones de los conductores tienen un efecto sobre los tiempos de viaje y casi ningún efecto sobre el consumo de energía

y los costos de carga.

Otro trabajo que implementa GA para desarrollar el E-VRP, corresponde al realizado por [15], en donde se resuelve el E-VRP con tiempos de carga y tiempos de viaje variable (EVRP-CTVTT, por sus siglas en inglés). Algunos de los parámetros considerados son: la utilización de 50 clientes y 20 estaciones de carga (CS, por sus siglas en inglés); el tiempo de recarga está fijado en 30 minutos y las baterías siempre se cargan al máximo de su capacidad. Los autores discretizaron un día en intervalos de dos minutos y aplican el algoritmo dinámico de Dijkstra para encontrar la ruta de tiempo de viaje más corta. Los autores presentaron un problema de la vida real y lo resolvieron aplicando un GA con un tiempo de ejecución promedio de tres horas. Los resultados indican que el GA produce un rendimiento aceptable en tiempo computacional, convergencia y calidad de solución.

En [3] el objetivo es encontrar y actualizar sus rutas de mínimo costo, planes de recarga, y tiempos de partida para visitar todos los destinos. La estrategia que usan consiste en dividir la operación en dos etapas, en donde cada una utiliza una GA distinto: pre-operación y operación online. En pre-operación, se calculan las rutas iniciales resolviendo un E-VRP *offline* (Off-E-VRP). En operación online, se actualizan las rutas según realizaciones del tráfico y mediciones del estado de los EV resolviendo un E-VRP *online* (On-E-VRP). Con esta codificación, se obtuvo que resolver el Off-E-VRP permite obtener buenas rutas iniciales, mientras que resolver el On-E-VRP mejora la operación y calidad del servicio.

Por otro lado, existen variaciones del GA para la optimización de rutas. Una de ellas son los GA híbridos (HGA). Esta variante combina una búsqueda local (LS) para la intensificación de soluciones, técnicas de gestión de diversidad de población de última generación y una búsqueda de vecindario grande (LNS) que actúa como operador de mutación. En [18] utilizan esta variante para resolver un VRP que combina vehículos convencionales, híbridos enchufables y eléctricos. Aquí, utilizan un *fitness* sesgado que combina el valor objetivo de un individuo con su contribución a la diversidad de la población para promover una buena exploración del espacio de búsqueda. Los resultados muestran que el HGA encuentra mejores resultados en promedio para todas las combinaciones de tipos de instancias.

2.2.2. Optimización Estocástica

La estocasticidad o incertidumbre aparece en todos los sistemas reales. Entre las diferentes fuentes de incertidumbre, se encuentran la incertidumbre aleatoria relacionada con la aleatoriedad inherente de los procesos naturales; y la incertidumbre epistémica, que se refiere al conocimiento limitado, es decir, a la carencia de datos o ausencia de información [19]. En términos generales, el modelamiento de un problema de optimización estocástica corresponde a [20]:

$$\begin{aligned} \min_{x \in \mathbb{R}} \quad & J(x_1, x_2, \dots, x_n) \\ \text{s.a} \quad & \mathbb{P}\{f(x_i, \alpha) > 0, \quad i = 1, \dots, n\} \geq (1 - \epsilon) \end{aligned}$$

donde J es la función objetivo que se busca optimizar, α es la incertidumbre y las n restricciones a las que está sujeta J , presentan probabilidades de cumplimiento.

Al igual que para el caso determinístico, una optimización estocástica debe considerar: variables o decisiones que hay que tomar; restricciones que limitan esas decisiones; una función objetivo o un criterio a optimizar [20]. Sin embargo, mientras que en la optimización determinista se supone que los parámetros del problema son conocidos con certeza, en la optimización estocástica se conocen solo las distribuciones de probabilidad de estos parámetros.

Así mismo, la diferencia entre una optimización determinística y una estocástica, radica en que para la primera una solución es factible cuando satisface todas las restricciones. Sin embargo, para la segunda, como las restricciones consideran la aleatoriedad de las variables no se puede hablar estrictamente de factibilidad, sino de probabilidad de que una cierta solución sea factible. Luego, las optimizaciones estocásticas son formulaciones conceptuales, que no se pueden resolver con un método convencional de optimización que solo puede manejar valores conocidos, sino que requiere una reformulación para tener aquellos valores conocidos.

Entonces, con el fin de caracterizar estas probabilidades de factibilidad, se hace necesario el uso de métodos que sean capaces de obtener o estimar las distribuciones de probabilidad, ya sea mediante caracterización o propagación de las incertidumbres asociadas a las variables del problema.

2.2.2.1. Propagación de Incertidumbres mediante el Método de Escenarios

La mayoría de las predicciones necesarias para la toma de decisiones, se basan en modelos computacionales que se aproximan a la realidad, pero que siempre tienen incertidumbres en sus predicciones debido a fuentes de variabilidad, aleatoriedad y estocasticidad [21]. Es por esto que considerar estas incertidumbres es esencial para obtener resultados realistas.

Uno de los métodos utilizados para incorporar incertidumbre a la formulación, es el enfoque de escenarios. Este proporciona una forma intuitiva de aproximar la solución a los problemas de optimización con restricciones de probabilidad, basada en encontrar la solución óptima bajo un número finito de resultados muestreados de la incertidumbre (llamados “escenarios”) [9]. Este enfoque abarca problemas de optimización de la forma:

$$\begin{aligned} \min_{x \in \mathbb{X}} \quad & J(x) = c^T x \\ \text{s.a.} \quad & \mathbb{P}[f(x, \delta) \leq 0] \geq (1 - \epsilon) \end{aligned} \tag{2.1}$$

donde $\mathbb{X} \subset \mathbb{R}^d$ es un conjunto compacto y convexo, c^T denota la transpuesta de un vector $c \in \mathbb{R}^d$, $\mathbb{P}[\cdot]$ es la medida de probabilidad sobre el conjunto de incertidumbre Δ , $f : \mathbb{R}^d \times \Delta \rightarrow \mathbb{R}$ es una función convexa en su primera argumento $x \in \mathbb{R}^d$ para la incertidumbre $\delta \in \Delta$ y ϵ es algún valor en el intervalo real abierto $(0, 1)$.

La restricción de probabilidad en (2.1) indica que para cualquier $x \in \mathbb{R}^d$ dado, el lado izquierdo representa la probabilidad de que x pertenezca al conjunto factible. Es decir:

$$\mathbb{P}[f(x, \delta) \leq 0] := \mathbb{P}\{\delta \in \Delta | f(x, \delta) \leq 0\}$$

Así, se considera que x es un punto factible de la restricción de probabilidad en (2.1) si

esta probabilidad es al menos $(1 - \epsilon)$.

La idea básica del enfoque de escenarios, es extraer un número específico $K \in N$ de muestras (escenarios) de la incertidumbre δ , y tomar la solución óptima que sea factible bajo todos estos escenarios (solución de escenarios) como un solución aproximada. El cálculo de la solución del escenario implica un programa de optimización determinista, que se obtiene reemplazando la restricción de probabilidad de (2.1) con las K muestras de restricciones deterministas.

Entre las ventajas de este enfoque, se tiene que no requiere del conocimiento de la función de distribución de la incertidumbre ni del conjunto de incertidumbres Δ (como generalmente se requiere en optimización estocástica), solo asume la disponibilidad de un número suficiente de muestras aleatorias independientes e idénticamente distribuídas (i.i.d.) de δ . Además, es computacionalmente eficiente, ya que, su solución se basa en un problema de optimización determinista que es canónicamente convexo, incluso cuando el problema original con restricciones de probabilidad no lo sea.

Si bien, el problema de optimización descrito en la ecuación (2.1) presenta una única restricción de probabilidad, el enfoque de escenarios también se puede utilizar en problemas con restricciones de probabilidad múltiple. Por ejemplo, en problemas de decisión estocástica de múltiples etapas en donde normalmente se debe aplicar una restricción de probabilidad por paso de tiempo, mientras que el problema de decisión óptima comprende varios pasos de tiempo.

A modo de ilustración, en [22] utilizan este enfoque para resolver un VRP con ventanas de tiempo inciertas. Aquí, definen un escenario como un tiempo de viaje aleatorio de cada arco de la red que se realiza en un solo rango, es decir, que en un escenario específico, cada arco tiene un rango de intervalo de posibles tiempos de viaje. Entonces, para cada realización de un escenario, primero encuentran la ruta que protege la solución contra el peor de los casos dentro de los rangos dados, para luego determinar la ruta óptima con respecto al costo mínimo esperado en el peor de los casos.

2.3. Estado del Arte

El E-VRP (ver Figura 2.1) tiene como objetivo el diseño de rutas de EV para servir a un conjunto de clientes, utilizando restricciones comunes en VRP, como la capacidad de carga del vehículo y las ventanas de tiempo de los clientes pero, además, se deben considerar restricciones propias de los EV, como la capacidad de carga limitada de las baterías, la infraestructura de carga disponible en las ciudades, operaciones de recarga de batería en la ruta y el estado de salud de las baterías para prevenir su degradación temprana [3]. La formulación del E-VRP es reciente, de hecho las primeras aproximaciones al VRP utilizando flotas eléctricas datan del 2011, cuando Gonçalves, Cardoso y Relvas resolvieron el problema de ruteo utilizando una flota mixta compuesta por EV y de combustión interna [8].

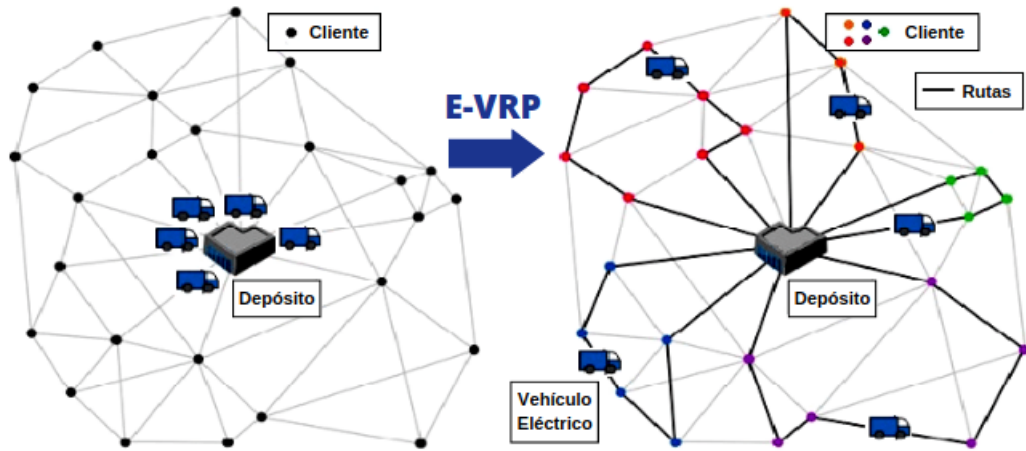


Figura 2.1: Modelo gráfico de un E-VRP genérico

Existen múltiples variantes del E-VRP y todas dependen del enfoque que el autor quiera dar al problema. A continuación, se realiza una revisión de las distintas variantes estudiadas hasta el momento.

Con el objetivo de obtener una primera aproximación al problema real de ruteo, muchos autores utilizan representaciones deterministas de las variables presentes en el E-VRP. Además, cada trabajo realizado emplea distintas funciones objetivo y restricciones por lo que surgen muchas variantes del E-VRP.

Uno de los primeros problemas de E-VRP en abordarse fue el Green VRP (GVRP), en donde la flota es propulsada por vehículos de combustibles alternativos (AFV): biodiesel, etanol, hidrógeno, metanol, gas natural, electricidad, etc [8]. La variante mas estudiada de este problema, considera las ventanas de tiempo de los clientes (E-VRPTW) [23][24][25], en donde los tiempos de recarga dependen del SOC de la batería al llegar a una CS y que, dada la limitada infraestructura de las CS, se utilizan flotas mixtas de vehículos eléctricos y de combustión interna. Así mismo, hay autores que estudian el E-VRPTW, agregando instancias de recargas parciales de batería (E-VRPTWPR), en donde se descubrió que una mejor solución implica que los vehículos carguen solo la cantidad de energía suficiente para terminar el segmento de la ruta hasta la siguiente estación de carga [8], eliminando del problema restricciones que se basan en la estrategia de carga completa del EV. Por otra parte, otros autores estudian el E-VRPTW con carga rápida (E-VRPTWFC), en donde se modelan variables binarias como variables de decisión para elegir qué tecnología de carga utilizar.

Hasta ahora, todos los E-VRP nombrados consideran variables determinísticas en su formulación, con el objetivo de reducir el costo computacional del problema y de dar un primer acercamiento al problema de ruteo. Sin embargo, aquellas formulaciones más realistas pueden incluir funciones de carga no lineal o la estocasticidad de las variables, entre otras. Así, en [26] se formula el E-VRP con funciones de carga no lineal (E-VRP-NL), en donde el tratamiento de la no linealidad se aborda por estimación del tiempo de carga mediante un enfoque basado en datos. Por otra parte, una forma de abordar la estocasticidad en el E-VRP, es considerando que el tráfico, dependiente de la hora día, es estocástico. Así, el problema se puede modelar como un ruteo dinámico de EV (D-EVRP)[27], que tiene por objetivo minimizar

la duración total del servicio. Otra manera consiste en considerar que los tiempos de viaje de los vehículos son las variables estocásticas del problema (E-VRPST) [28][29] o en [30] y [31] que abordan el E-VRP considerando que las demandas de los clientes son las variables estocásticas (E-VRPSD). Es importante destacar que, dado que el E-VRP es un problema reciente, el estudio de la variante estocástica del problema es escasa, principalmente debido a la complejidad computacional del algoritmo.

Entre las técnicas más utilizadas para resolver el E-VRP, se encuentran: las heurísticas de Búsqueda Adaptativa de Vecindarios Grandes (ALNS), Búsqueda de Vecindarios Grandes (LNS) y Algoritmos Genéticos (GA), siendo esta última la más utilizada y con la que los autores encontraron que la mayoría de las implementaciones no son eficientes [3]. Así mismo, los resultados más importantes obtenidos en estos estudios, muestran principalmente dos puntos: que las ventanas de tiempo son variables críticas cuando los EV tienen un rango de conducción máximo y tiempos de recarga elevados; y que el consumo de energía de los EV depende en gran medida de los perfiles de tráfico. Entre las técnicas utilizadas para resolver el E-VRP con variables estocásticas, se puede encontrar la proposición de un algoritmo de implementación híbrido (HRA)[27], muestreo de Monte Carlo [29], Búsqueda de Tabú (TS) [31], Aprendizaje por Refuerzo Seguro [30], entre otras soluciones heurísticas.

Una de las últimas implementaciones de este problema es la presentada en [3]. En este caso, la estrategia consistió en dividir la operación en dos etapas: pre-operación y operación *Online*. En pre-operación, se calculan las rutas iniciales resolviendo un E-VRP *Offline* (Off-EVRP). En operación online, se actualizan las rutas según realizaciones del tráfico y mediciones del estado de los EV resolviendo un E-VRP *Online* (On-EVRP). Ambas variantes del E-VRP se resuelven utilizando GA. Los resultados obtenidos muestran que resolver el Off-E-VRP permite obtener buenas rutas iniciales, mientras que resolver el On-E-VRP mejora la operación y calidad del servicio. Este trabajo de memoria pretende ser una continuación de la modelación presentada en [3], incorporando la estocasticidad de variables a la etapa de pre-operación.

2.4. Discusión

Dados todos los antecedentes presentados, se concluye que existen múltiples formas de abordar el E-VRP (estableciendo distintas funciones de costo a minimizar, con diversas restricciones), que generan diferentes variantes del problema. Aquella que toma en consideración las ventanas de tiempo de los clientes (E-VRPTW), es la variante más estudiada y es a partir de la cuál se generan nuevas variaciones en el E-VRP.

Entre los métodos de solución implementados para resolver el E-VRP, el GA es uno de los más estudiados. Así mismo, los resultados más importantes muestran principalmente dos puntos: que las ventanas de tiempo son variables críticas cuando los EV tienen un rango de conducción máximo y tiempos de recarga elevados; y que el consumo de energía de los EV depende en gran medida de los perfiles de tráfico.

Finalmente, la importancia de incorporar la estocasticidad de las variables al problema, reside en que pretende dar un cierto grado de realismo a las soluciones encontradas. Sin em-

bargo, debido al elevado costo computacional que suponen, la consideración de estocasticidad de variables es una de las variantes menos investigadas dentro del E-VRP.

Capítulo 3

Formulación de la Optimización basada en Escenarios para el Ruteo de Vehículos Eléctricos

3.1. Introducción

En [3] se diseña una estrategia para resolver el E-VRP, que divide la operación en dos etapas: pre-operación y operación *online*. En la pre-operación, se pretende que el sistema genere las rutas iniciales para cada vehículo de la flota. Para esto, se resuelve una variante del E-VRP llamada *Off-E-VRP*. A continuación, se formula una nueva estrategia para resolver el *Off-E-VRP*, utilizando el método de escenarios. En esta nueva estrategia se modifica la formulación para incluir la estocasticidad de las variables y restricciones probabilistas en el problema de optimización, lo que implica la modificación del GA a nivel de *fitness*.

3.2. Formulación del problema

Se tiene un grafo dirigido [32] $D = (\{0\} \cup N \cup F, A)$, en donde [3]:

- $\{0\}$ corresponde al depósito. Las rutas de cada vehículo eléctrico deben comenzar y terminar en él.
- $N = \{1, \dots, n\}$ contiene a los n nodos clientes que deben ser atendidos.
- $F = \{n + 1, \dots, n + s\}$ contiene a los s nodos correspondientes a las CS, en donde los vehículos pueden recargar su batería.
- A es un conjunto que contiene todos los arcos que conectan los nodos del grafo, i.e $A = \{(i, j) : \forall i, j \in V, i \neq j\}$, con $V = \{0\} \cup N \cup F$ el conjunto de todos los nodos.

En la Figura 3.1 se observa un ejemplo de grafo dirigido con los elementos ya mencionados.

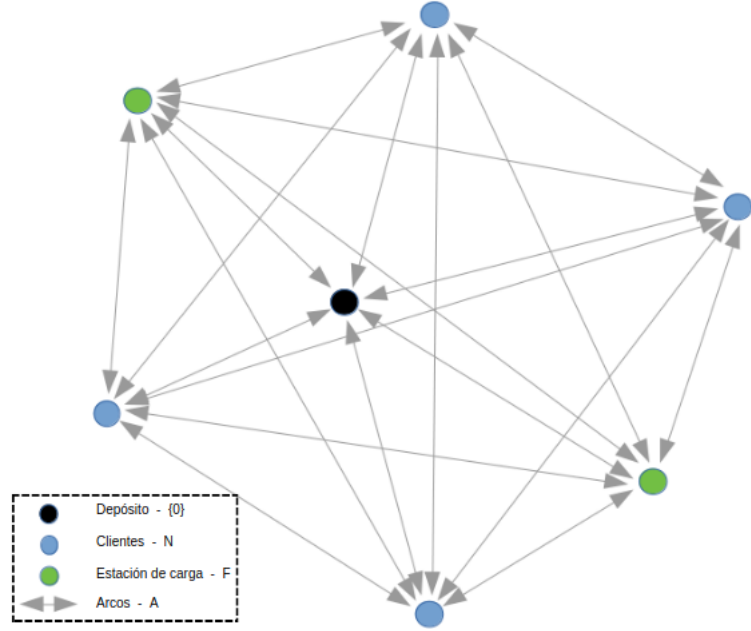


Figura 3.1: Ejemplo de grafo dirigido.

Cada cliente $n \in N$ tiene una demanda en peso D_n que debe ser satisfecha. El tiempo de servicio T_n para cada cliente, se encuentra restringido en una ventana de tiempo $[T_n^-, T_n^+]$, es decir, el vehículo debe llegar al cliente n antes de T_n^+ y después de T_n^- ; sin embargo, si el vehículo llega antes de que comience la ventana (T_n^-), puede esperar hasta que esta comience. De este modo, debido a la variación del tráfico, puede ser conveniente esperar en el cliente anterior después de entregar D_{n-1} ; o en el cliente actual antes de entregar D_n .

Con el objetivo de llegar a todos los clientes, se dispone de una flota M de EV disponibles. Cada vehículo $m \in M$ tiene tres limitaciones:

1. Puede transportar una masa de hasta \bar{D}_m
2. Tiene una capacidad de batería máxima \bar{Q}_m
3. Posee una duración máxima de recorrido $\bar{T}_m = \bar{T} \forall m \in M$.

Durante la operación, cuando un EV no posee la suficiente energía para completar su recorrido, puede visitar una CS $f \in F$, donde cada una tiene una capacidad limitada r_f para realizar operaciones de recarga de manera simultánea. Una operación de carga produce un aumento q en el estado de carga (SOC, por sus siglas en inglés) de la batería del vehículo, de modo tal que si un vehículo llega a la estación con un SOC p , al terminar la carga el SOC del vehículo será de $p + q$. El tiempo que toma la operación de recarga de batería, es modelado según el método presentado en [26].

Dado que operar la batería cerca del 0% ó 100% de su capacidad acelera su degradación [3], en la modelación del problema se incluye una política que restringe el SOC para operar dentro de límites bien establecidos. Esta política SOC está definida por el intervalo (20, 95) %.

Finalmente, son tres las variables que definen la operación de cada vehículo $m \in M$:

- La secuencia de nodos $S = [S^1, S^2, \dots, S^m]$ a visitar.
- El plan de recarga $L = [L^1, L^2, \dots, L^m]$.
- El tiempo de partida $x_0 = [x_0^1, x_0^2, \dots, x_0^m]$, que define el momento del día en el que comienza el servicio de cada EV de la flota.

De esta forma, una solución a este problema de optimización debe retornar: la secuencia óptima (S^*) de nodos a visitar, el plan de recarga óptimo (L^*) y los tiempos óptimos de partida (x_0^*) de cada EV.

3.3. Modelación de la dinámica de los vehículos

La función a optimizar en un E-VRP no tiene una estructura fija y puede constituir la suma de diversos costos, dependiendo de lo que se quiera optimizar. Para introducir el problema de optimización, primero se modelan las variables que definirán las rutas de los EV, para luego definir las variables de estado del sistema.

La secuencia de nodos que debe visitar el EV $m \in M$ se define como [3]:

$$S^m = [S_0^m, S_{s_m-1}^m]^T$$

donde S_k^m es el nodo k -ésimo que visita el vehículo m y s_m es el largo del vector S^m , que depende del número de clientes y las operaciones de recarga en la ruta. Además, el plan de recarga para el EV m se define como:

$$L^m = [L_0^m, L_{s_m-1}^m]$$

donde L_k^m representa el incremento de SOC en la parada k -ésima. Luego, el tiempo de inicio del servicio para el EV m será:

$$x^m = x_0^m$$

Ahora, la velocidad del EV al viajar desde el nodo i hasta el nodo j , depende de la hora del día $t_{día}$ en que el EV deja el nodo i . Esto se denota como $v_{ij}(t_{día})$. Sin embargo, esta velocidad es estocástica, ya que un cambio en el tráfico puede ocasionar una reducción considerable de la velocidad de viaje entre los dos nodos. Entonces, se asume que la velocidad distribuye normal [3] en cada momento del día, es decir:

$$v_{ij}(t_{día}) \sim \mathcal{N}\{\mu_{ij}(t_{día}), \sigma_{ij}(t_{día})\}$$

A continuación, es crítico calcular los tiempos de viaje y consumos energéticos entre dos nodos, pues de ellos dependen el tiempo y SOC al llegar a los distintos destinos. Las ecuaciones (3.1) y (3.2) describen el cálculo de tiempos de viaje y consumo energético, respectivamente, a partir del valor de la velocidad.

$$tt_{ij} = \frac{d_{ij}}{v_{ij}} \quad (3.1)$$

$$Ec_{ij} = m \cdot \left[\frac{g C_r}{1000} \cdot \langle d_{ij}, \cos(\Theta_{ij}) \rangle \cdot (c_1 v_{ij} + c_2) + g \cdot \langle d_{ij}, \sin(\Theta_{ij}) \rangle \right] + \frac{1}{2} \rho_{aire} A_f C_d v_{ij}^2 d_{ij} \quad (3.2)$$

en donde d_{ij} corresponde a la distancia entre los nodos i y j y Θ_{ij} es el perfil de inclinación del camino entre i y j . En la Tabla 3.1 se observa la descripción y el valor de cada una de las constantes utilizadas en estas ecuaciones.

Tabla 3.1: Descripción y valor de constantes utilizadas para calcular el consumo de energía entre cada par de nodos.

Constante	Descripción	Valor
m	Masa (EV + conductor)	1521 [kg]
g	Aceleración de gravedad	9,8 [m/s ²]
C_r	Coefficiente de resistencia a la rodadura	1,75
c_1	-	4,575
c_2	-	1,75 [s/m]
ρ_{aire}	Densidad de masa de aire	1,2256 [kg/m ³]
A_f	Zona frontal del EV	2,3316 [m ²]
C_d	Coefficiente aerodinámico de arrastre EV	0,28

Como se puede observar, la relación entre tiempos de viaje y consumos energéticos es no lineal con la velocidad. Esta no linealidad produce una gran distorsión de la distribución de los tiempos de llegada, por lo que no se puede asumir que estos distribuirán normal, incluso al linealizar. Por lo tanto, se debe considerar utilizar métodos de propagación de incertidumbre para estimar las distribuciones de los tiempos y SOC de llegada.

A continuación, es necesario indicar que tanto el tiempo de operación como la demanda en peso en la parada k del EV m corresponden a variables determinísticas; mientras que el tiempo de viaje y la disminución de SOC entre las paradas k y $k + 1$ del EV m son variables estocásticas.

Luego, la formulación del problema, considera la modelación de tres variables de estado para cada EV de la flota. Dos de estas variables de estado (x_1 y x_2) siguen una función de densidad de probabilidad (PDF) de la forma:

$$P[a < X < b] = \int_a^b f_X(x) dx \quad (3.3)$$

Sin embargo, obtener la función explícita de esta PDF es muy complicado, por lo que se necesita de algún método para obtener las incertezas de esas variables. Las variables de estado del problema se definen a continuación:

- $x_1^m(k)$ es una variable aleatoria que sigue la distribución del momento del día en que el EV m llega a la parada k . Luego, $x_1^m(k + 1)$ corresponde a la suma de: la distribución del momento del día en que el EV m llega a la parada k ($x_1^m(k)$); el tiempo que demore el vehículo m en estar en la parada k ($T_{op}^m(k)$); el tiempo de viaje entre las paradas k y

$k + 1$ ($t^m(k)$); y el tiempo de espera en la parada $k + 1$ (h^m).

$$x_1^m(k + 1) = x_1^m(k) + T_{op}^m(k) + t^m(k) + h^m$$

- $x_2^m(k)$ es una variable aleatoria que sigue la distribución del estado de carga (SOC, por sus siglas en inglés) de la batería, al momento en que el EV m llega a la parada k . Luego, $x_2^m(k + 1)$ corresponde a la suma de: la distribución del SOC al momento en que el EV m llega a la parada k ($x_2^m(k)$); el incremento del SOC en la parada k (L_k^m); y el descuento del SOC al terminar el servicio en la parada k ($e^m(k)$).

$$x_2^m(k + 1) = x_2^m(k) + L_k^m - e^m(k)$$

- $x_3^m(k)$ corresponde a la masa que le queda por transportar al EV m al llegar a la parada k . Luego, $x_3^m(k + 1)$ corresponde a la suma de la masa del EV m al llegar a la parada k ($x_3^m(k)$), menos la demanda del cliente en la parada k ($D^m(k)$).

$$x_3^m(k + 1) = x_3^m(k) - D^m(k)$$

Entonces, el problema de optimización que se busca resolver en esta memoria, incluye la definición de cuatro costos a minimizar: tiempo total de viaje (ecuación (3.4)), tiempo total de recarga (ecuación (3.5)), energía total consumida (ecuación (3.6)) y el costo total de recarga (ecuación (3.7)); sujetos a restricciones sobre la capacidad de carga de los vehículos, las ventanas de tiempo de los clientes, el SOC de las baterías y las condiciones iniciales y de término.

Los costos a considerar son los que se describen a continuación:

$$J_1 = \sum_{m \in M} \sum_{k \in K^m} E\{t^m(k)\} \quad (\text{tiempo total de viaje}) \quad (3.4)$$

$$J_2 = \sum_{m \in M} \sum_{k \in K^m} \xi_{S_k} L_k^m \quad (\text{tiempo total de recarga}) \quad (3.5)$$

$$J_3 = \sum_{m \in M} \sum_{k \in K^m} E\{e^m(k)\} \quad (\text{energía consumida total}) \quad (3.6)$$

$$J_4 = \sum_{m \in M} \sum_{k \in K^m} \Delta_{S_k}(E\{x_2^m(k)\}, L_k^i) \quad (\text{costo de recarga total}) \quad (3.7)$$

Para considerar la estocasticidad del problema, estos costos están definidos para cada uno de los escenarios muestreados (ver Sección 3.4.4.3). Luego, el costo total será:

$$J = \omega_1 \cdot J_1 + \omega_2 \cdot J_2 + \omega_3 \cdot J_3 + \omega_4 \cdot J_4 \quad (3.8)$$

donde $\omega_i, i = \{1, 2, 3, 4\}$ son pesos ajustables e indican la importancia que se le quiere dar a cada costo.

Las restricciones de este problema de optimización, se clasifican en 6 categorías:

1. Condiciones iniciales:

$$S_0^m = 0 \quad (\text{inicio en depósito}) \quad (3.9)$$

$$L_0^m = 0 \quad (\text{incremento nulo de SOC inicial}) \quad (3.10)$$

$$x_1^m(0) = x_0^m \quad (\text{tiempo de partida}) \quad (3.11)$$

$$x_2^m(0) = \alpha^+ \quad (\text{SOC inicial}) \quad (3.12)$$

$$x_3^m(0) = \sum_{S_k^m \in N} D_{S_k^m} \quad (\text{demanda en peso inicial}) \quad (3.13)$$

2. Condiciones de término: la ecuación (3.14) obliga a todos los vehículos a terminar en el depósito y la ecuación (3.15) evita que la duración total del recorrido supere el tiempo máximo \bar{T} .

$$S_{s_m-1}^m = 0 \quad (3.14)$$

$$P_{x_1}^{k,m} \left\{ \frac{x_1^m(s_m - 1) - x_1^m(0)}{\bar{T}} \leq 1 \right\} \geq p_1 \quad (3.15)$$

3. La restricción (3.16) previene que los vehículos transporten más peso que el máximo permitido.

$$\sum_{S_k^m} D_{S_k^m} \leq \bar{D}_m \quad (3.16)$$

4. Ventanas de tiempo: la restricción (3.17) establece que la distribución del momento del día en que el EV m llega a la parada k , debe ser mayor o igual a un umbral mínimo $T_{S_k^m}^-$; mientras que la restricción (3.18) establece que el tiempo que dura el servicio en la parada k , debe ser menor o igual a un umbral máximo $T_{S_k^m}^+$.

$$P_{x_1}^{k,m} \left\{ \frac{x_1^m(k)}{T_{S_k^m}^-} \geq 1 \right\} \geq p_2 \quad (3.17)$$

$$P_{x_1}^{k,m} \left\{ \frac{x_1^m(k)}{T_{S_k^m}^+ - T_{op}(k)} \geq 1 \right\} \geq p_3 \quad (3.18)$$

5. Estado de salud de las baterías: la ecuación (3.19) limita el SOC de los vehículos que llegan a un nodo, mientras que la ecuación (3.20) limita el SOC de los vehículos que salen de un nodo.

$$P_{x_2}^{k,m} \left\{ \alpha^- \leq x_2^m(k) \leq \alpha^+ \right\} \geq p_4 \quad (3.19)$$

$$P_{x_2}^{k,m} \left\{ \alpha^- \leq x_2^m(k) + L_k^m \leq \alpha^+ \right\} \geq p_5 \quad (3.20)$$

6. Capacidad de las estaciones de carga: se trabaja bajo el supuesto de que las CS disponibles, tienen una capacidad limitada de EV que pueden estar cargando sus baterías simultáneamente. Esta limitación se observa en la ecuación (3.21).

$$CS_{capacity} = 2 \quad (3.21)$$

A continuación se observa un resumen del problema de optimización a resolver:

$$\begin{aligned} \min \quad & J(S, L, x_0) = \omega_1 \cdot J_1 + \omega_2 \cdot J_2 + \omega_3 \cdot J_3 + \omega_4 \cdot J_4 \\ \text{s.a} \quad & \text{restricciones (3.9) - (3.21)} \end{aligned}$$

donde $S = [S^1, \dots, S^m]$, $L = [L^1, \dots, L^m]$ y $x_0 = [x_0^0, \dots, x_0^m]$ son las variables de decisión de grupo de cada EV.

3.4. Formulación de la toma de decisiones estocástica para el ruteo de EV

La toma de decisiones estocástica implementada, corresponde a la ejecución del GA incorporando las incertezas de las variables. Esto comprende la evaluación estocástica de cada individuo de la población. Para ello, se utilizan escenarios, caracterizando así las restricciones probabilísticas del problema de ruteo. Sin embargo, esta evaluación estocástica es muy costosa computacionalmente, ya que se debe evaluar con todos los escenarios a todos los individuos de la población, en todas las generaciones del GA. Para reducir este costo, se utiliza una heurística que permite evaluar de manera estocástica solo cierto número de individuos de la población.

3.4.1. Obtención de escenarios

La primera etapa de la formulación implementada, corresponde a la obtención de los escenarios. En este caso en particular, estos se obtienen antes de ejecutar el GA; sin embargo, pueden existir formulaciones en donde la obtención de los escenarios ocurran dentro del GA. Cabe destacar que el lenguaje utilizado para la programación descrita a continuación, fue Python.

Para obtener los escenarios, se cuenta con una instancia generada artificialmente, que posee la velocidad de viaje (media y desviación estándar) al ir de un nodo a otro a distintas horas del día. En específico, la instancia posee las velocidades entre nodos, cada 30 minutos, es decir, posee las velocidades de viaje a las $t = 00^{00}$, $t = 00^{30}$, $t = 01^{00}$, $t = 01^{30}$, ..., $t = 23^{00}$, $t = 23^{30}$.

Luego, un escenario corresponde a un valor de velocidad distinto para cada arco que une dos nodos en la red a cierta hora del día. Es decir, en el escenario η a las t horas se cuenta con una velocidad $v_{\eta_{12}}(t)$ para ir desde el nodo 1 al nodo 2, una velocidad $v_{\eta_{23}}(t)$ para ir desde el nodo 2 al nodo 3, y así sucesivamente; mientras que en el mismo escenario a las $t + 1$ horas, se cuenta con una velocidad $v_{\eta_{12}}(t + 1)$ para ir desde el nodo 1 al nodo 2, una velocidad $v_{\eta_{23}}(t + 1)$ para ir desde el nodo 2 al nodo 3, etc. Estos valores de velocidad, se generan a partir de muestras aleatorias de la distribución de probabilidad normal de la velocidad. El procedimiento se divide en dos etapas:

1. **Obtención de velocidades desde la instancia:** Primero, se obtienen los estadísticos de las velocidades desde la instancia que se está utilizando. En la instancia, se definen

la media y la desviación estándar de las velocidades de viaje entre cada arco que une dos nodos de la red de tráfico, en intervalos de 30 minutos. El procedimiento en detalle, se observa en el Algoritmo 2. En particular, los datos de velocidad de la instancia, se guardan en un diccionario de **Python** para luego poder acceder a ellos.

Algoritmo 2 *Obtención de velocidades desde la instancia*

Input: Cantidad de nodos n_{nodos} de la instancia I .

Output: Velocidades por arco, cada 30 minutos $\mathcal{V}_{arco-tpo}$

```

 $\mathcal{V}_{arco-tpo} \leftarrow \{\};$  ▷ Creación diccionario vacío
Minutos  $\leftarrow [0, 30, 60, \dots, 1410];$  ▷ Lista con minutos en un día
for minuto in Minutos do ▷ Por cada minuto en la lista
     $\mathcal{V}_{arco-tpo}[\text{minuto}] \leftarrow \{\};$ 
    for  $nodo_{desde} \leftarrow 0$  to  $n_{nodos}$  do ▷ Por cada arco de la instancia
        for  $nodo_{hacia} \leftarrow 0$  to  $n_{nodos}$  do
             $\mathcal{V}_{arco-tpo}[\text{minuto}][nodo_{desde}, nodo_{hacia}] \leftarrow (\mu_I, \sigma_I);$  ▷ Se guardan las velocidades
return  $\mathcal{V}_{arco-tpo}$ 

```

2. **Creación de escenarios:** Una vez guardadas las velocidades de los arcos de la instancia, se deben crear las realizaciones de velocidad para cada uno de los escenarios. Al igual que las velocidades de la instancia, cada realización tendrá valores de velocidad en intervalos de 30 minutos, para cada uno de los arcos de la red. El procedimiento en detalle, se observa en el Algoritmo 3.

Algoritmo 3 *Creación de realizaciones de velocidad*

Input: Velocidades $\mathcal{V}_{arco-tpo}$ extraídas de la instancia.

Output: Velocidades por escenario $\mathbb{V}_{escenario}$; Cantidad de escenarios $N_{escenarios}$

```

 $\mathbb{V}_{escenario} \leftarrow \{\};$ 
for id  $\leftarrow 0$  to  $N_{escenarios}$  do
     $\mathbb{V}_{escenario}[\text{id}] \leftarrow \{\};$  ▷ Creación diccionario vacío para cada escenario
    for minuto in  $\mathcal{V}_{arco-tpo}$  do
         $\mathbb{V}_{escenario}[\text{id}][\text{minuto}] \leftarrow \{\};$ 
        for arco  $\in \mathcal{V}_{arco-tpo}[\text{minuto}]$  do
             $\mu, \sigma = \mathcal{V}_{arco-tpo}[\text{minuto}][\text{arco}];$  ▷ Se obtiene la velocidad de instancia
             $\mathbb{V}_{escenario}[\text{id}][\text{minuto}][\text{arco}] \leftarrow \text{SampleNormal}(\mu, \sigma);$  ▷ Muestreo de la velocidad
return  $\mathbb{V}_{escenario}$ 

```

3.4.2. Codificación de Individuos

Para manejar programáticamente a los individuos de las poblaciones, se utiliza la codificación presentada en [3], en donde cada individuo corresponde a la concatenación de 3 bloques (ver Figura 3.2).

El primer bloque, almacena la secuencia de clientes que debe visitar cada EV. A modo de ejemplo, en la Figura 3.2 se tiene una flota de dos EV, en donde el EV 1 debe visitar las

paradas k_0, k_4, k_3, k_9 y k_7 ; mientras que el EV 2 debe visitar las paradas k_5, k_1, k_8, k_2 y k_6 , en ese orden. El segundo bloque, hace referencia a las operaciones de recarga de batería a las que pueden optar los EV. A modo de ejemplo, en la Figura 3.2, se tiene la operación de carga a^j , cuya interpretación es la siguiente: después de visitar al nodo cliente a_1^j , el EV visitará el nodo CS a_2^j , en donde cargará una cantidad a_3^j . Luego, el último bloque corresponde a los tiempos de partida de cada EV. En la Figura 3.2, se observa la codificación de los tiempos de partida para los dos EV del ejemplo anterior. Finalmente, la codificación para el individuo I será:

$$I = [S, a, t]$$

Secuencia de clientes	Operaciones de carga	Tiempos de partida
$S = [k_0^1, k_4^1, k_3^1, k_9^1, k_7^1, , k_5^2, k_1^2, k_8^2, k_2^2, k_6^2]$	$a^j = [a_1^j, a_2^j, a_3^j]$	$t = [t_1, t_2]$

Figura 3.2: Codificación de individuos. Elaboración propia

Luego, el largo del individuo viene dado por

$$L = N + 3m + 3n^*$$

donde N corresponde a la cantidad de clientes disponibles en la red, m es el tamaño de la flota y n^* el número de operaciones de recarga. Además, se permitirá un máximo de de operaciones de recarga igual al doble del tamaño de la flota, esto es:

$$n^* = 2m$$

Un ejemplo que ilustra esta codificación se presenta en el Anexo A. Además, las operaciones de *crossover* y mutación realizadas para cada individuo de la población codificado con los bloques descritos anteriormente, se especifican en el Anexo B.

3.4.3. Población Inicial

Al igual que en el caso anterior, para generar la población inicial, se utiliza la heurística presentada en [3], con una pequeña modificación para incluir todos los escenarios. Esta heurística se describe a continuación:

- El algoritmo recibe un individuo candidato construido a partir de la heurística presentada en [3]. En resumen, se ordenan los clientes según el límite inferior de su ventana de tiempo; luego, atraviesa los clientes solicitados, los agrega a una nueva ruta y cada vez que los requisitos en la ruta superan el límite de peso de los EV, se inicia una nueva ruta. Como resultado, se genera una solución candidata y el tamaño inicial de la flota.
- Crea nuevos individuos al mutar el individuo del paso anterior. Esto se repite hasta que se haya creado un tercio de la población.
- Se rellena la población con individuos aleatorios.

El procedimiento anterior se realiza para cada escenario, obteniendo una población inicial para cada escenario. Luego, se elige aleatoriamente una de estas poblaciones para introducir como población inicial en el GA.

3.4.4. Evaluación de Fitness y Manejo de Restricciones

Debido al elevado costo computacional que implica la evaluación estocástica de todos los individuos en todos los escenarios y para todas las generaciones del GA, se necesita utilizar una heurística que permita evaluar estocásticamente sólo cierto número de individuos por generación. En este contexto, se consideran dos formas de evaluar a los individuos obtenidos mediante la implementación del GA: el *fitness* determinístico y el *fitness* estocástico. El primero se utilizará para evaluar a todos los individuos de la población; y el segundo, sólo para evaluar a los H individuos con mejor *fitness* determinístico (ver sección 3.4.4.1). La diferencia entre ambas formas de evaluar a los individuos, radica en que con el *fitness* determinístico se supone que todas las restricciones del problema son determinísticas [3]; mientras que para el *fitness* estocástico se considera la estocasticidad del tiempo de viaje y el consumo de energía, según las restricciones (3.15), (3.17), (3.18), (3.19) o (3.20).

La formulación matemática tanto para el *fitness* determinístico como para el estocástico, es equivalente para ambos. En general, sea $\Gamma(I_j)$ el *fitness* del individuo I_j , se tiene que

$$\Gamma(I_j) = -J(\Psi(I_j)) - \Pi(I_j, \mathcal{K}) \quad (3.22)$$

en donde $J(\cdot)$ corresponde a la función de costo de la ecuación (3.8), $\Psi(I_j)$ decodifica al individuo I_j en las variables de decisión S , L y x_0 , \mathcal{K} es una constante positiva y $\Pi(\cdot)$ es una función de penalización que solo afecta a los individuos infactibles.

3.4.4.1. Evaluación Determinística y Estocástica del Fitness

Como ya se ha mencionado, existen dos formas de evaluar a los individuos obtenidos mediante la implementación del GA, con el objetivo de reducir el elevado costo computacional asociado a la propagación de incerteza. La estrategia consiste en lo siguiente: todos los individuos de la población se evaluarán con el *fitness* determinístico, para luego elegir a H individuos, con mejor *fitness* determinístico, a evaluar con el *fitness* estocástico. De esta forma, se evalúan detalladamente sólo aquellos individuos que tengan alta probabilidad de ser factibles.

Debido a que se está evaluando sólo cierto número de individuos de manera estocástica, puede ocurrir que el *fitness* (determinístico) de un individuo que no fue seleccionado para evaluación estocástica, sea mejor que el *fitness* (estocástico) de uno de los H mejores individuos; por lo que, en la próxima generación, este individuo (con *fitness* determinístico) puede ser elegido, erróneamente, entre los H mejores individuos. Para evitar cometer este error, se necesita de un mecanismo que haga que el *fitness* de los individuos que no fueron evaluados de manera estocástica, sea peor que el de los que si lo fueron.

El mecanismo implementado se observa en el Algoritmo 4. En primer lugar, se debe obtener la diferencia entre el *fitness* determinístico y estocástico de los H mejores individuos

seleccionados. A continuación, la mayor de estas diferencias (más una constante) se agrega al *fitness* determinístico de todos los individuos que no fueron evaluados estocásticamente, asegurando así que el *fitness* de estos individuos (aquellos que no se evaluaron de manera estocástica) sea mayor al *fitness* de los H mejores individuos seleccionados para ser evaluados estocásticamente.

Por otro lado, existen tres formas de seleccionar el número H de individuos a evaluar estocásticamente. El primer método consiste en evaluar estocásticamente todos los individuos que son factibles determinísticamente (H_1); el segundo, corresponde a evaluar estocásticamente los H mejores individuos que son factibles determinísticamente (H_2); y, en el tercero, se evalúan estocásticamente las H mejores soluciones factibles o infactibles determinísticamente (H_3), a través de las generaciones del GA.

Algoritmo 4 *Evaluación estocástica de la población $P(t)$*

- 1: Evaluación determinística de $P(t)$.
 - 2: Ordenamiento creciente de los *fitness* determinísticos de la población.
 - 3: Selección de H individuos a evaluar con *fitness* estocásticamente.
 - 4: Evaluación estocástica de los H individuos.
 - 5: Calcular diferencia entre *fitness* estocástico y determinístico de los H individuos.
 - 6: Guardar la diferencia mayor y sumar una constante: $diff = \max(\text{diferencias}) + \delta$.
 - 7: Sumar este valor a los *fitness* determinísticos de los demás individuos.
-

3.4.4.2. Penalizaciones

En la implementación de este método de escenarios, se realizan dos tipos de penalizaciones, una para cada tipo de evaluación de *fitness*. En primer lugar, las penalizaciones utilizadas para calcular el *fitness* determinístico, deben obtenerse por incumplir las restricciones determinísticas, es decir, por no satisfacer las restricciones (3.15), (3.16) y (3.21) y la formulación determinística de las restricciones (3.17), (3.18), (3.19) y (3.20) [3].

Para introducir matemáticamente esta penalización, se considera un problema con p restricciones y q variables, de modo que se tiene el vector de optimización $\bar{x}_j \in \mathbb{R}^q$ asociado al individuo I_j , la matriz $A \in \mathbb{R}^{p \times q}$ y el vector $b \in \mathbb{R}^p$. Luego, las restricciones determinísticas se pueden escribir como

$$A\bar{x}_j \leq b$$

Así, para medir cuán factible es un individuo, se utiliza una penalización cuadrática, definida matemáticamente como:

$$D_i = \begin{cases} (A_i\bar{x}_j - b_i)^2 & \text{si } A_i\bar{x}_j > b \\ 0 & \text{si } A_i\bar{x}_j \leq b \end{cases} \quad (3.23)$$

donde el subíndice i corresponde a la restricción i -ésima, A_i es la fila i -ésima de A y b_i el elemento i -ésimo de b . Luego, la penalización al evaluar determinísticamente el individuo I_j

corresponde a la suma acumulada de las distancias y un valor constante grande:

$$\Pi_d(I_j, \mathcal{K}_1) = \begin{cases} \sum_{i=1}^p D_i + \mathcal{K}_1 & \text{si se viola una restricción} \\ 0 & \text{si no se viola ninguna restricción} \end{cases} \quad (3.24)$$

En este caso las restricciones de igualdad siempre se alcanzan. Por lo tanto, no se requiere de otras metodologías para enfrentar este tipo de restricciones [3].

El segundo tipo de penalización, se aplica cuando se calcula el *fitness* estocástico. En este caso, la penalización final corresponderá a la suma de los siguientes términos:

1. Una penalización debido a que la probabilidad de satisfacer el límite de duración total de recorrido de cada EV (restricción estocástica (3.15)) sea menor a un umbral $P_{MXTT} = p_k$.
2. Una penalización por sobrepasar el cargamento que pueden llevar los EV (restricción determinística (3.16)).
3. Una penalización debido a que la probabilidad de satisfacer la restricción probabilística de las ventanas de tiempo de los clientes (restricciones estocásticas (3.17) y (3.18)) sea menor a un umbral $P_{TW} = p_k$.
4. Una penalización debido a que la probabilidad de satisfacer la restricción probabilística del SOC de las baterías de los EV (restricciones estocásticas (3.19) y (3.20)) sea menor a un umbral $P_{SOC} = p_k$ ¹.
5. Una penalización por sobrepasar la capacidad de vehículos que pueden cargar su batería simultáneamente en una estación de carga (restricción determinística (3.21)).

Debido a que en este caso se tienen escenarios, se debe manejar esta penalización de modo de obtener solo una penalización para cada individuo de la población. En primer lugar se tendrá una penalización por escenarios, que corresponderá a la penalización por no satisfacer las restricciones determinísticas del problema. Matemáticamente, esta penalización se ve igual que la de la ecuación (3.23), pero considerando solo las restricciones determinísticas, es decir, aquellas descritas en las ecuaciones (3.16) y (3.21). Dado que esta penalización está definida por escenarios, para obtener la penalización determinística final del individuo, se considera la media de las penalizaciones determinísticas por escenarios.

Por otro lado, también se penaliza por incumplir las restricciones probabilísticas del problema. En este caso se cuenta con un valor p_k que corresponde a la probabilidad con la que se desea que se cumpla la restricción estocástica k y \mathbb{P}_k es la probabilidad de, en efecto, satisfacer la restricción k y se calcula como

$$\mathbb{P}_k = \frac{\text{Cantidad de escenarios que satisfacen la restricción } k}{\text{Total de escenarios}}$$

¹ En este caso en particular, se utilizó que $P_{MXTT} = P_{TW} = P_{SOC}$. Sin embargo, estas probabilidades son independientes entre sí, por lo que pueden ser distintas.

Entonces, la penalización al evaluar estocásticamente el individuo I_j corresponde a la suma de la media de las penalizaciones determinísticas por escenario y la penalización estocástica:

$$\Pi_e(I_j, \mathcal{K}_2) = \begin{cases} \underbrace{\frac{1}{H_T} \cdot \sum_{\eta=1}^{H_T} D_\eta}_{\text{P. Determinística}} + \underbrace{\sum_{k=1}^5 [\omega_{1_k} \cdot (\mathbb{P}_k - p_k) + c_{1_k}] + \mathcal{K}_2}_{\text{P. Estocástica}} & \text{si se viola una restricción} \\ 0 & \text{si no se viola ninguna restricción} \end{cases} \quad (3.25)$$

donde H_T es la cantidad total de escenarios utilizados, el sub-índice k corresponde a la k -ésima restricción estocástica, ω_{1_k} es un peso y c_{1_k} y \mathcal{K}_2 son constantes.

3.4.4.3. Costos operacionales

Los costos operacionales de cada individuo a evaluar, se calculan mediante dos métodos. Dado que cada uno de los escenarios es una realización distinta de la velocidad, es de esperarse que cada escenario tenga un costo operacional distinto.

En un principio, se cuenta con cuatro costos operacionales por escenario, que corresponden al tiempo total de viaje (J_1 en (3.4)), el tiempo total de recarga (J_2 en (3.5)), el consumo de energía total (J_3 en (3.6)) y el costo de recarga total (J_4 en (3.7)). Estos valores deben ser trabajados, con el objetivo de obtener solo un costo de cada uno para el individuo I_j . A continuación, se proponen dos maneras de hacerlo.

La primera opción consiste en calcular el costo promedio de todos los escenarios. Así, sea H_T la cantidad de escenarios, el costo operacional del individuo I_j será:

$$J_{1_{I_j}} = \frac{\sum_{i=1}^{H_T} J_1}{H_T} \quad (\text{tiempo total de viaje}) \quad (3.26)$$

$$J_{2_{I_j}} = \frac{\sum_{i=1}^{H_T} J_2}{H_T} \quad (\text{tiempo total de recarga}) \quad (3.27)$$

$$J_{3_{I_j}} = \frac{\sum_{i=1}^{H_T} J_3}{H_T} \quad (\text{energía consumida total}) \quad (3.28)$$

$$J_{4_{I_j}} = \frac{\sum_{i=1}^{H_T} J_4}{H_T} \quad (\text{costo de recarga total}) \quad (3.29)$$

La segunda opción consiste en comparar los costos de cada uno de los escenarios y asignar como costo del individuo I_j aquel de mayor valor. Para ello, primero se calcula el costo operacional por escenario, para después compararlos. Así, para el escenario κ se tiene el costo operacional

$$J_{T_\kappa} = J_{1_\kappa} + J_{2_\kappa} + J_{3_\kappa} + J_{4_\kappa}$$

A continuación, se comparan los costos operacionales obtenidos por cada escenario y aquel que tenga mayor valor, será el costo del individuo I_j . Así, si el escenario η es aquel con un mayor costo, es decir:

$$J_\eta = \max_{\kappa} J_{T_\kappa} \quad \forall \kappa \in \{1, \dots, H_T\}$$

luego:

$$J_\eta = J_{1_\eta} + J_{2_\eta} + J_{3_\eta} + J_{4_\eta}$$

entonces el costo del individuo I_j corresponderá a:

$$J_{1_{I_j}} = J_{1_\eta} \quad (\text{tiempo total de viaje}) \quad (3.30)$$

$$J_{2_{I_j}} = J_{2_\eta} \quad (\text{tiempo total de recarga}) \quad (3.31)$$

$$J_{3_{I_j}} = J_{3_\eta} \quad (\text{energía consumida total}) \quad (3.32)$$

$$J_{4_{I_j}} = J_{4_\eta} \quad (\text{costo de recarga total}) \quad (3.33)$$

Finalmente, en ambos casos el costo operacional del individuo I_j se calcula como:

$$J(\Psi(I_j)) = \omega_1 \cdot J_{1_{I_j}} + \omega_2 \cdot J_{2_{I_j}} + \omega_3 \cdot J_{3_{I_j}} + \omega_4 \cdot J_{4_{I_j}}$$

3.4.5. Algoritmo Genético Implementado

El Algoritmo 5 resume todos los procedimientos que realiza el GA implementado para resolver el E-VRP utilizando el método de escenarios. De aquí se observa que:

1. Primero se crea la población inicial, según lo expuesto en la sección 3.4.3.
2. Luego, se realiza la evaluación de esta población: todos los individuos se evalúan de manera determinística y los H mejores se evalúan de manera estocástica utilizando el método de escenarios, según la heurística del Algoritmo 4.
3. A continuación, a través de las generaciones, se aplican los operadores del GA sobre la población: selección, mutación y *crossover*, para generar la descendencia.
4. Después, esta nueva población se evalúa determinística y estocásticamente, obteniendo la población de la siguiente generación.
5. Una vez que se hayan ejecutado todas las generaciones, se obtiene el mejor individuo en términos de *fitness*.

Algoritmo 5 *Algoritmo Genético utilizado [3]*

Input: Red D ; Flota M ; Hiperparámetros (H); Individuo inicial (I^0)**Output:** Secuencia de nodos, plan de carga y tiempos de partida óptimos (S^* , L^* , x_0^*)

Inicialización;

 $\mathcal{P} \leftarrow []$;Agregar I^0 a \mathcal{P} ; ▷ Creación de población inicialLlenar \mathcal{P} hasta $H[\text{tamañoPoblación}]/3$ con mutaciones de I^0 ;Llenar el resto de \mathcal{P} con individuos aleatorios;Evaluación determinística y estocástica de \mathcal{P} ;**for** *generación* $\leftarrow 1$ **to** $H[\text{maxGeneraciones}]$ **do** $\text{IndividuosElite} \leftarrow$ Clonar los mejores $H[\kappa]$ individuos en \mathcal{P} ; $\hat{\mathcal{P}} \leftarrow \text{SelecciónTorneos}(\mathcal{P}, H[\Upsilon])$;▷ Operador Selección **for** *hijo* in $\hat{\mathcal{P}}$ con probabilidad $H[\text{MUTPB}]$ **do** *hijo* \leftarrow Mutación (*hijo*, H);▷ Operador Mutación **for** (*hijo 1*, *hijo 2*) en $\hat{\mathcal{P}}$ con probabilidad $H[\text{CXPB}]$ **do** (*hijo 1*, *hijo 2*) \leftarrow (*hijo 1*, *hijo 2*, H);▷ Operador *Crossover* Evaluación determinística y estocástica de $\hat{\mathcal{P}}$; Eliminar los $H[\kappa]$ individuos con menor *fitness* de $\hat{\mathcal{P}}$; Insertar IndividuosElite en $\hat{\mathcal{P}}$; $\hat{\mathcal{P}} \leftarrow \mathcal{P}$;▷ Actualización población nueva generación $\text{MejorIndividuo} \leftarrow$ Individuo con el *fitness* mas alto; $(S^*, L^*, x_0^*) \leftarrow \Psi^\alpha(\text{MejorIndividuo})$;**return** S^*, L^*, x_0^*

Capítulo 4

Estudios por Simulación del Modelo Desarrollado

4.1. Configuración Experimental

Para verificar el rendimiento del sistema de toma de decisiones, se realizó el estudio de una instancia generada artificialmente a partir de datos reales. En la Figura 4.1, se observa una imagen de la instancia utilizada para resolver el E-VRP. En ella, se puede observar que la instancia cuenta con: 25 clientes, a quienes se les debe satisfacer su demanda; 3 estaciones de carga, que, de ser necesario, los vehículos pueden visitar para cargar su batería; y un depósito, que es lugar desde dónde los vehículos parten su recorrido y en dónde lo finalizan. La instancia cubre un área de 10.000 [m²].

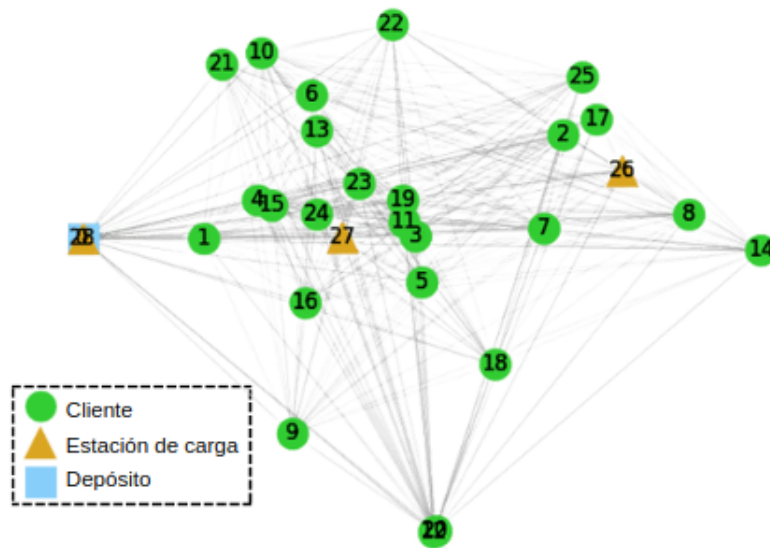


Figura 4.1: Instancia utilizada para ejecutar el GA.

Para resolver esta instancia, se utilizan dos EV homogéneos. En la Tabla 4.1 se muestran los valores de las constantes de las restricciones operativas, para estos EV. En particular, notar que se exige un mínimo del 80 % de probabilidad de cumplir las restricciones, para que la solución sea factible.

Tabla 4.1: Constantes utilizadas en las restricciones del problema [3].

Variable	Descripción	Valor
α^+	SOC máximo	95 %
α^-	SOC mínimo	20 %
\bar{D}	Peso máximo	553[kg]
\bar{T}	Tiempo máximo de recorrido	21.600[s]
p_1	Probabilidad de cumplimiento del tiempo de duración del recorrido	80 %
$p_2 = p_3$	Probabilidad de cumplimiento de ventanas de tiempo	80 %
$p_4 = p_5$	Probabilidad de cumplimiento del SOC	80 %

4.1.1. Pesos de la Función de Costos

En la Tabla 4.2 se muestran los pesos utilizados en la función de costo. Estos valores fueron obtenidos mediante la heurística descrita en [3]. Como se puede notar, en la función de costo se le da una importancia mayor al consumo de energía. En cuánto a la obtención de los costos por individuo, se utiliza el método de calcular el costo promedio de los escenarios descrito en la Sección 3.4.4.3.

Tabla 4.2: Pesos de la función de costos.

Peso	Costo por el que se multiplica	Valor
ω_1	Tiempo de viaje	$45.085 \cdot 10^{-3}$
ω_2	Tiempo de recarga	$1 \cdot 10^{-2}$
ω_3	Costo de recarga	$2 \cdot 10^{-3}$
ω_4	Consumo de energía	2.291

4.2. Estudios realizados

A continuación, se estudia el rendimiento del GA implementado para resolver el *Off-E-VRP* utilizando el método de escenarios, sobre la instancia descrita en la sección 4.1. A modo de ejemplo, en la Figura 4.2, se puede ver la velocidad de viaje entre los nodos 0 y 17 a distintas horas del día. Aquí se observa la media y la desviación estándar de la velocidad que entrega la instancia ya descrita; y los valores muestreados de esta velocidad, que generan las velocidades de los distintos escenarios.

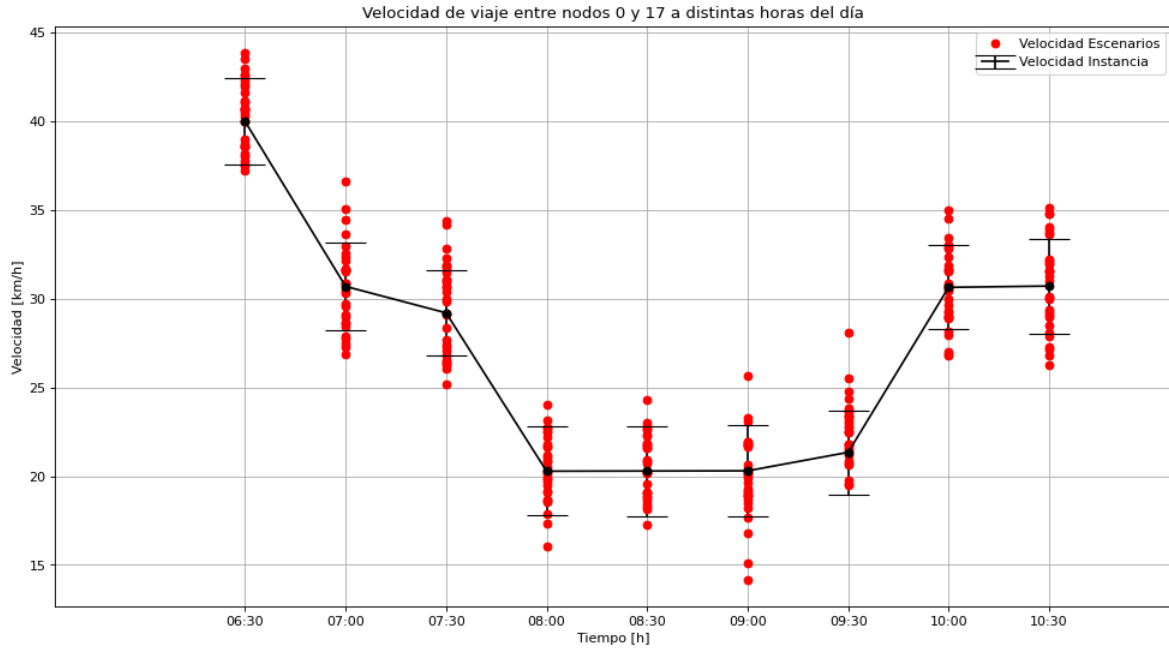


Figura 4.2: Comparación de la velocidad de viaje de la instancia y de los escenarios, entre los nodos 0 y 17, a distintas horas del día.

En la Tabla 4.3, se muestran los hiperparámetros utilizados para resolver el problema. El criterio utilizado para seleccionar el tamaño de la población y el número máximo de generaciones del GA, corresponde al presentado en [3]. En particular, para esta instancia, se obtiene que el tamaño de la población es de 115 individuos y que el máximo de generaciones para el GA es de 220.

Tabla 4.3: Valores hiperparámetros del problema

Hiperparámetro	Descripción	Valor
CXPB	Probabilidad de <i>crossover</i>	0.7
MUTPB	Probabilidad de mutación	0.9
μ	Tamaño de la población	115
ρ	Máximo de generaciones	220
κ	Individuos elite	1
\mathcal{K}_1	Constante de penalización determinística	$5 \cdot 10^5$
\mathcal{K}_2	Constante de penalización estocástica	10^6
Υ	Tamaño de la selección de torneos	5
H_T	Número de escenarios	30
H	Número individuos para evaluación estocástica	{6, 10, 14}

4.2.1. Cumplimiento de Restricciones Estocásticas

Para analizar el funcionamiento del GA, en la Figura 4.3 se visualiza un ejemplo de la evolución de las variables de estado para una ruta factible (en este caso, el EV0 de la flota), evaluando estocásticamente 10 individuos por cada generación del GA. Los números que aparecen en cada parada, corresponden a la identificación del nodo visitado según la Figura 4.1. Por ejemplo, el EV0 parte en el depósito (nodo 0) y visita a los clientes 16 y 1, para luego ir a la estación de carga 27, y así sucesivamente.

En primer lugar, en cuanto a los resultados de tiempos de salida y llegada y del SOC de la batería, se tiene que, en color azul, se encuentra la formulación determinística para resolver el E-VRP; y en color gris, la solución utilizando el método de escenarios. Luego, a la derecha, se observa que se satisface la restricción determinística que limita el cargamento que puede llevar el EV. Notar que la masa que acarrea el vehículo en cada una de las paradas, es igual para todos los escenarios.

En la formulación determinística, se observa que se cumplen tanto las restricciones de ventanas de tiempo de clientes, como las del SOC de la batería del EV. Para cumplir esta última, el EV debe recurrir a visitar una estación de carga para recargar su batería. Sin embargo, esta formulación resulta riesgosa, pues hay ocasiones en las que cumple las restricciones pero sin márgenes de error. Por ejemplo, por un lado en el gráfico de los tiempos de salida y llegada, en la parada 8 (al visitar al cliente 9), se observa que el EV0 llega justo al comienzo de la ventana de tiempo de este cliente; mientras que en el gráfico de SOC de la batería, se muestra que el EV termina su recorrido justo con un 20 % de SOC, para cumplir las restricciones. En cambio, con la formulación estocástica implementada, se tiene una mayor holgura para estos casos, y si bien hay escenarios que incumplen las restricciones, la probabilidad de cumplirlas se mantiene por sobre el umbral establecido.

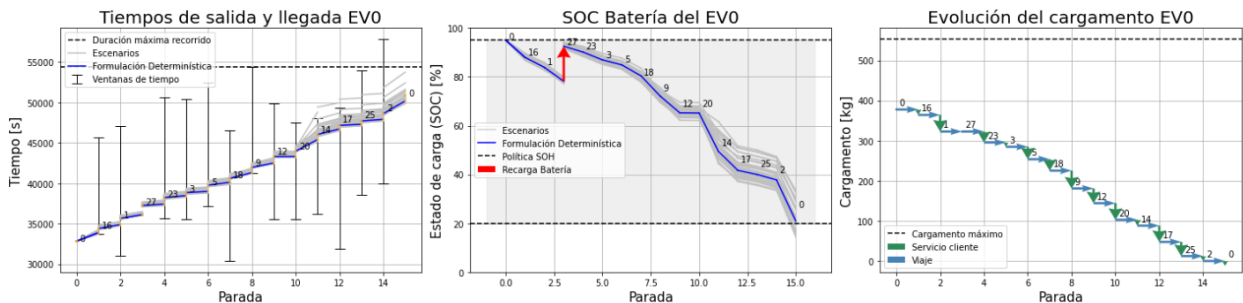


Figura 4.3: Ejemplo de ruta factible.

Con el objetivo de visualizar de manera detallada el cumplimiento de las restricciones estocásticas, en la Figura 4.4 se examinan algunas ventanas de tiempo de interés. En concreto, en la Figura 4.4.a, se observa en detalle la ventana de tiempo del cliente 20 (según Figura 4.1) visitado por el EV0, en la posición 10. Aquí, se tiene que se cumplen las restricciones de las ventanas de tiempo -tanto para el límite superior como para el límite inferior- con un 100 % de probabilidad con ambas formulaciones. Luego, en la Figura 4.4.b se tiene la ventana de tiempo del cliente 14 (según Figura 4.1) visitado en la posición 11. Aquí, se observa que la probabilidad tanto de llegar como de salir de esta parada dentro de la ventana de tiempo del cliente es del 96,67 %, dado que existe un escenario que termina de realizar el servicio

en un momento que está fuera de la ventana de tiempo del cliente. Además, se puede notar que existe un escenario en el que se activa la restricción para el límite superior de la ventana. A pesar de esto, como la probabilidad de cumplir esta restricción en esta parada es mayor al 80 %, es decir, que más del 80 % de los escenarios cumple la restricción de la ventana de tiempo del cliente, la solución estocástica sigue siendo factible.

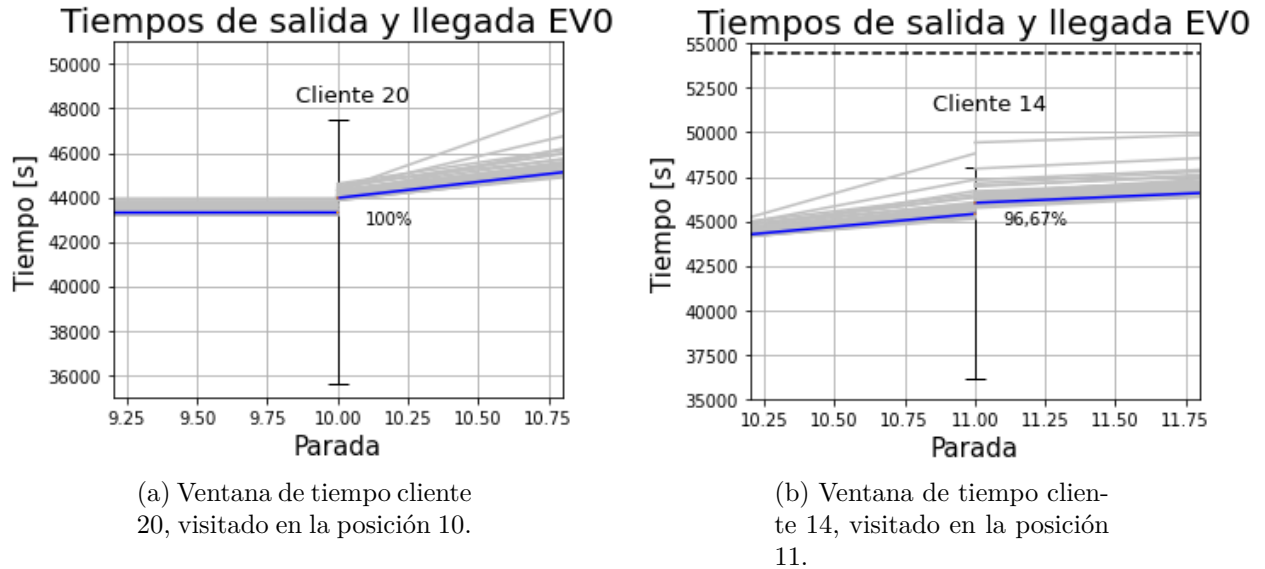


Figura 4.4: Vista en detalle de algunas ventanas de tiempo de clientes.

A continuación, en la Figura 4.5 se observa con mayor detalle el SOC de la batería del EV0, al terminar el recorrido. De aquí se desprende que en la formulación determinística, el EV alcanza a terminar su recorrido sin violar la restricción del límite inferior de SOC. Por otro lado, en cuanto a la formulación estocástica se tiene que hay escenarios en donde esta restricción sí se viola. A pesar de esto, la restricción estocástica del límite inferior del SOC se satisface con un 83,33 % de probabilidad con el método de escenarios, por lo tanto, la solución sí es factible.

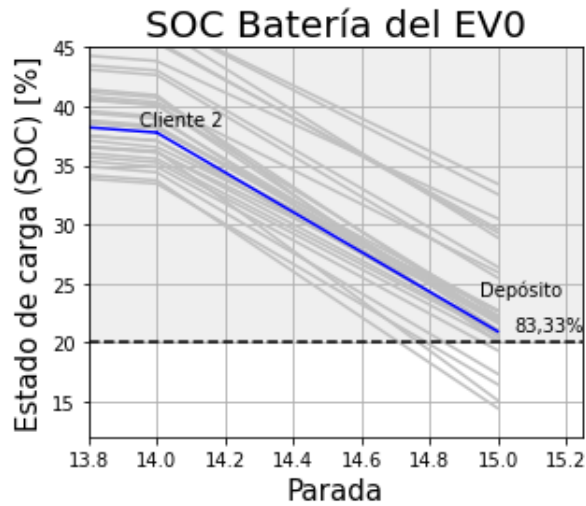
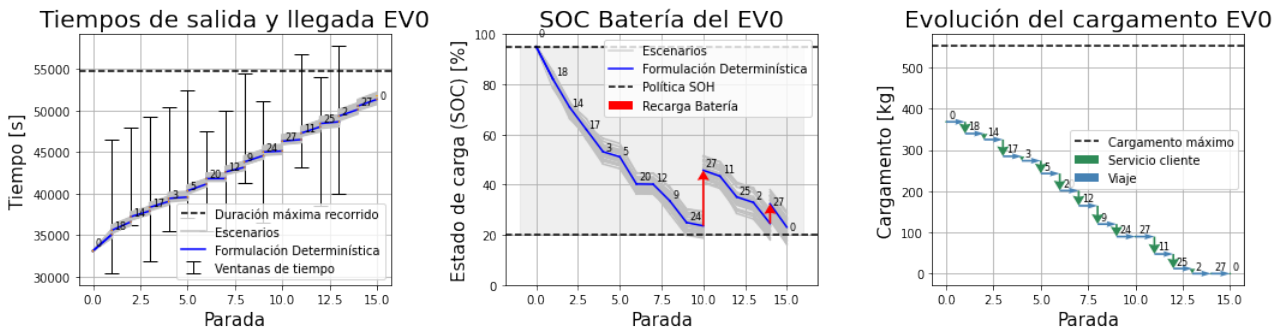
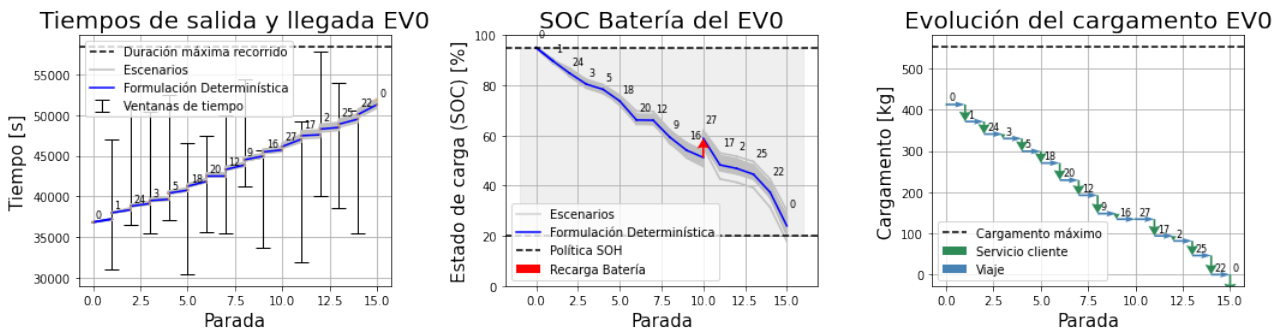


Figura 4.5: Vista en detalle del SOC de la batería al terminar el recorrido.

Finalmente, en la Figura 4.6, se observan dos ejemplos de la evolución de las variables de estado para una solución factible, evaluando estocásticamente 10 individuos por cada generación del GA. Para la Figura 4.6.a, se utilizaron 50 escenarios y para la Figura 4.6.b se utilizaron 100 escenarios. Cabe destacar que el GA encontró soluciones distintas para los dos casos.



(a) Ejemplo para EVO, utilizando 50 escenarios



(b) Ejemplo para EVO, utilizando 100 escenarios

Figura 4.6: Ejemplos de soluciones factibles, al aumentar el número de escenarios muestreados

En cuanto a los tiempos de salida y llegada, al utilizar 50 escenarios se cumplen todas las restricciones con un 100 % de probabilidad; mientras que al usar 100 escenarios, se satisface la ventana de tiempo de los clientes 16 y 22 con una probabilidad del 90 %, por lo que la solución es factible (ya que ambas probabilidades son mayores al 80 %). En relación con el SOC de la batería, al utilizar 50 escenarios, esta restricción se viola al llegar al cliente 24, a la estación de carga 27 y al finalizar el recorrido en el depósito (nodo 0); mientras que al utilizar 100 escenarios, la restricción de SOC se satisface con una probabilidad del 93 % al llegar al depósito. Sin embargo, al igual que en el caso anterior, como la probabilidad de satisfacer esta restricción, fue mayor al 80 % en ambos casos, las soluciones son factibles. Respecto a la restricción de cargamento que puede llegar el EV, en ambos casos esta restricción fue satisfecha.

Con estos resultados, es posible inferir que la cantidad de escenarios no está directamente relacionada con la probabilidad de satisfacer restricciones, es decir que aumentar el número de escenarios, no implica que menos restricciones serán satisfechas. Es más, en los tres casos propuestos se encontraron soluciones factibles.

4.2.2. Cantidad de Individuos Evaluados Estocásticamente

Para manejar el costo computacional asociado a la propagación de incertidumbre, se utiliza una heurística que permite evaluar de manera estocástica solo ciertos individuos en cada generación del GA. Para seleccionar estos individuos, se utilizan tres métodos:

- H_1 : Se evalúan estocásticamente todos los individuos factibles determinísticamente.
- H_2 : Se evalúan estocásticamente los mejores H individuos factibles determinísticamente.
- H_3 : Se evalúan estocásticamente los mejores H individuos factibles o infactibles determinísticamente.

Otro método que surge de manera natural, corresponde a evaluar estocásticamente todos los individuos de la población en cada generación del GA. Este método se denominará H_4 . A continuación, en la Tabla 4.4 se observa el rendimiento en tiempo computacional y *fitness* al ejecutar el GA 30 veces, utilizando tres valores distintos de H . Notar que todas las soluciones obtenidas fueron factibles para esta nueva formulación, por lo que el *fitness* obtenido solo tiene la componente de costo operacional y ninguna penalización. Entonces, según la ecuación (3.22), se cumple que:

$$\Gamma(I_j) = -J(\Psi(I_j))$$

Tabla 4.4: Métricas al utilizar distintos métodos para calcular la cantidad de escenarios a evaluar estocásticamente.

Método	Valor de H	Costo Operacional	Tiempo Computacional [s]
H_1	-	1.146, 15 \pm 158, 75	2.221, 75 \pm 217, 58
	6	1.147, 17 \pm 151, 07	511, 38 \pm 102, 49
H_2	10	1.162, 51 \pm 167, 88	818, 78 \pm 207, 28
	14	1.107, 76 \pm 165, 73	1.007, 96 \pm 228, 55
H_3	6	1.097, 63 \pm 132, 26	609, 00 \pm 156, 75
	10	1.147, 02 \pm 136, 48	972, 58 \pm 220, 86
	14	1.154, 92 \pm 178, 86	1.034, 16 \pm 91, 71
H_4	-	1.158, 61 \pm 128, 08	2.247, 44 \pm 172, 46

Para una mejor visualización, en la Figura 4.7 se grafican los datos mostrados en la Tabla 4.4.

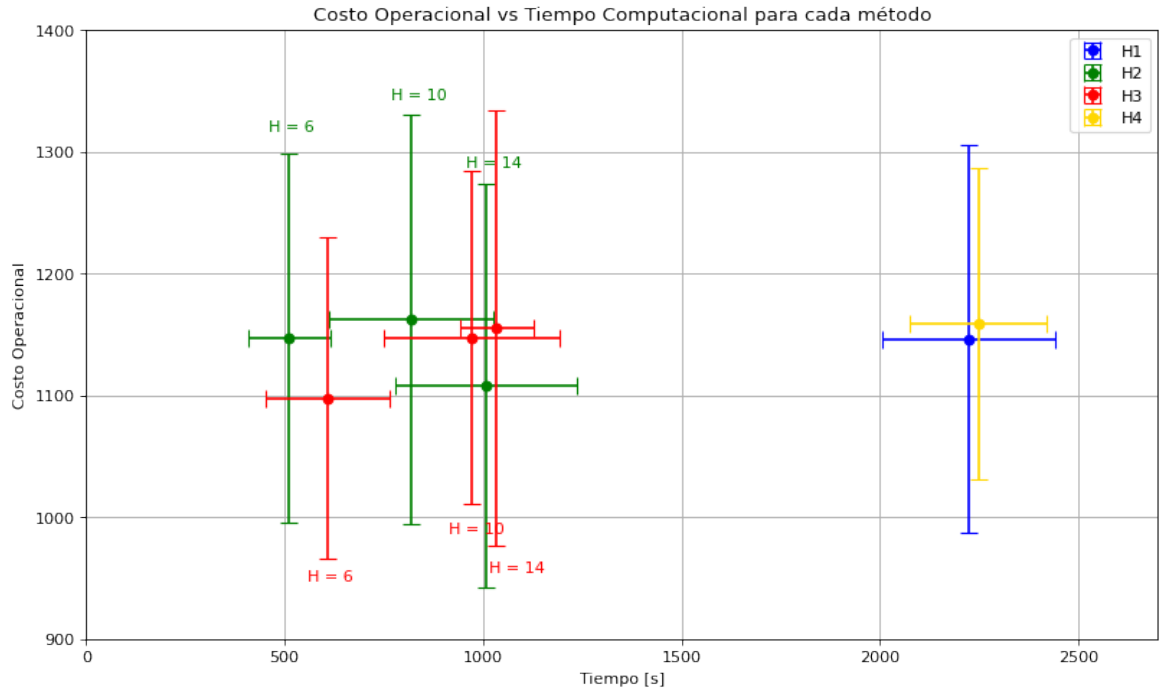


Figura 4.7: Comparación de costo operacional y tiempo computacional entre los distintos métodos utilizados.

De aquí, se obtiene que en cuanto al tiempo computacional, aquel método más rápido -para un mismo valor de H - corresponde a H_2 , le sigue H_3 , y aquellos más lentos fueron H_1 y H_4 . En cuanto al compromiso entre optimalidad y tiempo computacional, los métodos que pueden ser considerados mejores, son H_2 y H_3 utilizando $H = 6$, ya que, el primero demora menos tiempo, pero el segundo tiene un costo operacional menor. Luego, dependiendo de lo que se pretenda optimizar (el tiempo o el costo operacional), es posible elegir un método u otro. Por otro lado, al comparar el tiempo computacional que le toma a cada método para encontrar una solución, se observa que el método H_2 (utilizando $H = 6$) es hasta cuatro

veces más rápido que los métodos H_1 y H_4 . Sin embargo, llama la atención que la diferencia en términos de tiempo computacional es estrecha, entre los métodos H_1 y H_4 , considerando que el último método debería evaluar más individuos que H_1 en cada generación del GA.

El método H_2 es el que menos tiempo demora porque en las primeras generaciones del GA, se evalúa solo un individuo (o unos pocos individuos), ya que aún no se han encontrado individuos que sean factibles de manera determinística. Sin embargo, a medida que el algoritmo avanza, se descubren más y más individuos factibles determinísticos, por lo que se comienzan a evaluar estocásticamente un mayor número de individuos. En particular, con el método H_2 se tiene una cota superior de soluciones a evaluar estocásticamente, por lo que demorará menos tiempo. Con el método H_3 , siempre se evaluarán estocásticamente H individuos, sin importar si estos son factibles o infactibles determinísticamente. Luego, con el método H_1 , no se tiene una cota superior de individuos a evaluar estocásticamente, por lo que en las generaciones posteriores puede ocurrir que el 60 % de los individuos sean factibles de manera determinística y todos ellos deben ser evaluados estocásticamente. Finalmente, con el método H_4 se espera que todos los individuos de la población se evalúen de manera estocástica, por lo que debe ser aquel método que mas tiempo demore.

A modo de ejemplo, este comportamiento se puede observar en la Figura 4.8, de donde se desprende que a partir de la generación 50, el método H_1 comenzará a evaluar entre el 40 % y 60 % de la población total en cada generación, lo que se traduce en un tiempo computacional mayor, considerando que los métodos H_2 y H_3 evalúan, estocásticamente, aproximadamente el 10 % de los individuos de las poblaciones. Luego, el método H_4 evalúa en torno al 70 % y 80 % de la población en cada generación. Se observa que no se evalúa el 100 % de la población, por lo siguiente: ocurre que hay individuos que pertenecen a la población $\mathcal{P}(t)$ y debido a su valor de *fitness*, son parte de la descendencia, es decir, pertenecerán también a la población $\mathcal{P}(t+1)$; luego, como ya se calculó el *fitness* de este individuo en $\mathcal{P}(t)$, no se volverá a calcular en $\mathcal{P}(t+1)$.

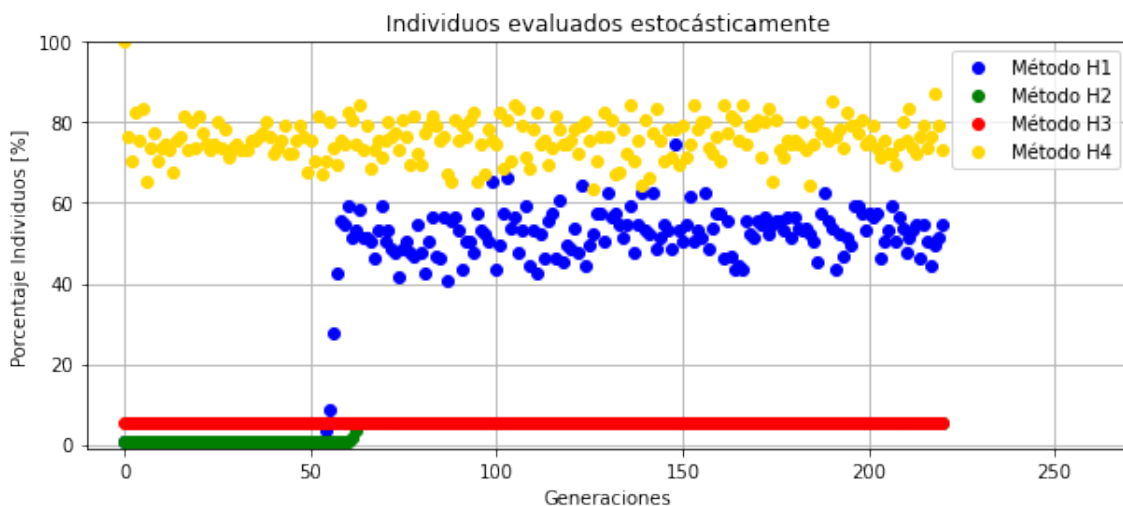


Figura 4.8: Porcentaje de individuos evaluados estocásticamente a través de las generaciones del GA, con los métodos mencionados, para $H = 6$.

Por otro lado, una pregunta interesante tiene relación con el progreso obtenido, en tér-

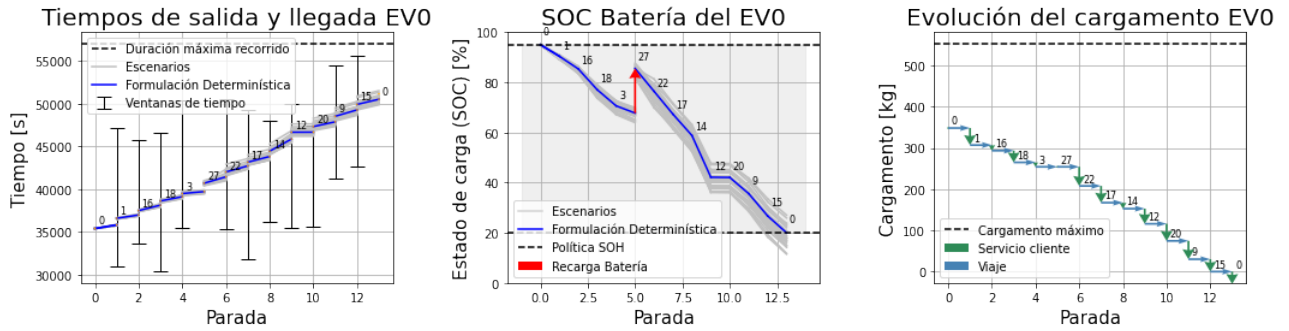
minos de *fitness* o costo operacional, al evaluar estocásticamente una mayor cantidad de individuos. De la Figura 4.4 se tiene que, en general, el valor el costo operacional se mantiene entre rangos similares para todos los métodos (1100 y 1200). Es más, el costo operacional del método H_4 fue mayor a los obtenidos con los otros métodos y, teniendo en consideración que es el que más tiempo computacional demora, este no parece ser el método más adecuado para evaluar individuos estocásticamente.

Es importante recordar que estos análisis son válidos para la instancia descrita en la Figura 4.1. Sin embargo, lo descrito anteriormente puede no cumplirse en instancias mas complejas (con un mayor número de clientes), por lo que es necesario estudiar el comportamiento de esta heurística con esas instancias, ya que, para estos propósitos, es atractivo considerar heurísticas de este estilo.

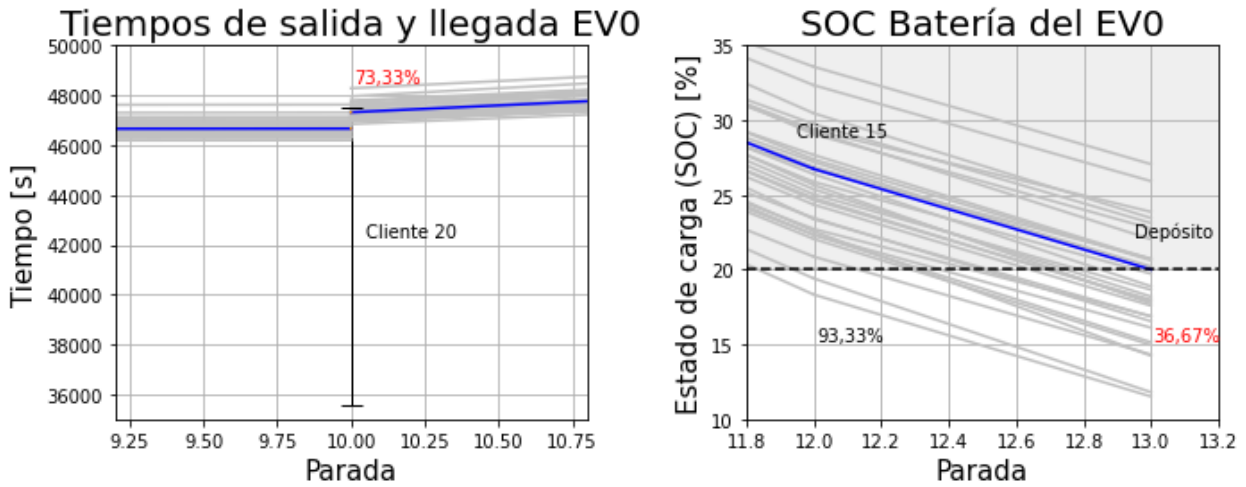
4.2.3. Comparación con la Formulación Determinística

En esta sección se busca realizar una comparación entre la formulación determinística y la estocástica desarrollada en este trabajo. Para ello, se analiza la probabilidad de infringir las restricciones estocásticas: duración máxima del recorrido, ventanas de tiempo de los clientes y SOC de las baterías de los EV, en ambas formulaciones.

Para calcular estas probabilidades en la formulación determinística, se realiza el siguiente procedimiento: se ejecuta el GA de la formulación determinística desarrollada en [3] para encontrar una solución; a continuación, esta solución se evalúa con los escenarios muestreados de la velocidad, para luego verificar qué porcentaje de las restricciones se incumplen. En la Figura 4.9 se observa el resultado de aplicar el procedimiento descrito. En primer lugar, se destaca que el costo operacional de esta solución fue de 1.168,51, lo que está dentro de lo esperado, al ser comparado con los costos operacionales de la formulación estocástica, presentados en la Figura 4.7.



(a) Tiempos de salida y llegada, SOC de la batería y evolución del cargamento del EVO.



(b) Vista en detalle de paradas de interés.

Figura 4.9: Aplicación de escenarios a la solución determinística obtenida.

Los resultados de los tiempos de salida y llegada, el SOC de la batería y la evolución del cargamento del EVO de la solución obtenida con la aplicación de los escenarios, se observan en la Figura 4.9.a. De aquí se tiene que la formulación determinística cumple todas las restricciones del problema (tiempo máximo de duración del recorrido, ventanas de tiempo, SOC de la batería y el cargamento máximo que puede llevar el EV). Sin embargo, hay paradas en las que las restricciones se activan precisamente en el límite de una restricción.

Lo anterior, se observa en la Figura 4.9.b, en donde se tiene el detalle de algunas paradas de interés de esta solución. En particular, al llegar al cliente 20 se observa que sólo el 73,33% de los escenarios cumplen la restricción del límite superior de la ventana de tiempo. Dado que esta probabilidad es menor al 80%, esta solución no es factible. Por otro lado, en cuanto al SOC, se tiene que al visitar al penúltimo cliente (el cliente 15), el 93,33% de los escenarios cumple con la restricción. Sin embargo, al finalizar el recorrido y llegar al depósito, sólo el 36,67% de los escenarios cumple con la restricción final de SOC, por lo que, como esta probabilidad es menor al 80% de los escenarios, la solución no es factible. Entonces, dado que se incumplen restricciones de ventanas de tiempo y de SOC de la batería, esta solución -obtenida al agregar los escenarios a la formulación determinística- no es factible.

Entonces, al comparar los resultados obtenidos al aplicar este método y el estocástico, se observa que el resultado determinístico entrega 100 % factibilidad, mientras que el estocástico no. En un principio, esto puede sonar pesimista, pero no lo es, ya que el método estocástico considera la incertidumbre de las variables y es capaz de afirmar cuán fiable es la solución encontrada. Entonces, el método estocástico es más robusto que el determinístico, ya que entrega soluciones que se ajusten a la realidad.

Esta es la ventaja fundamental de la formulación estocástica desarrollada, por sobre la determinística: en la primera se manejan las incertezas de las variables, logrando regular la probabilidad con la que se acepta que se violen las restricciones (en este caso, se permitía un máximo de 20 %) para obtener soluciones factibles. Como esto no sucede en la formulación determinística, cuando esta cumple las restricciones con un margen muy estrecho de diferencia entre los límites de ventanas de tiempo o de SOC, es probable que al agregar incerteza al problema, la solución ya no sea factible.

Capítulo 5

Conclusiones y Trabajo Futuro

En este trabajo de memoria de título, se ha desarrollado una nueva estrategia para optimizar rutas para flotas de vehículos eléctricos, que asigna clientes y define las potenciales rutas que debe seguir cada vehículo, antes de comenzar su operación. Esta formulación toma en consideración la estocasticidad de la velocidad y del consumo energético, mediante el uso de restricciones probabilísticas sobre las ventanas de tiempo de los clientes y el SOC de las baterías. La propagación de las incertezas de estas variables, se realiza mediante el método de escenarios, generando distintas realizaciones de velocidad para cada uno de los arcos que unen los nodos de la red.

Para resolver este E-VRP, se desarrolla un GA con una heurística que permite manejar y reducir el elevado costo computacional asociado a la propagación de incertezas de las variables. Esta heurística corresponde a evaluar toda la población del GA con la formulación determinística, para luego seleccionar a aquellos individuos con mejor *fitness* determinístico, y evaluarlos con la formulación estocástica desarrollada. De esta forma, se evalúa en detalle solo a aquellas soluciones que tengan potencial de ser óptimas.

Para estudiar el rendimiento del sistema, se realizan dos estudios principalmente. El primero, consiste en la evaluación -en términos de costo operacional y tiempo computacional- de cuatro métodos para la selección de individuos a evaluar estocásticamente. Los resultados muestran que no existen mejoras sustanciales en torno al costo operacional de la solución al aumentar el número de individuos a evaluar estocásticamente (el costo operacional se mantiene en el mismo intervalo, para todos los métodos), por lo que resulta más adecuado utilizar métodos que limiten la cantidad de individuos a evaluar estocásticamente, reduciendo así el costo computacional del algoritmo.

El segundo estudio, comprende la comparación entre la formulación propuesta y la formulación determinística del E-VRP desarrollada en [3]. Los resultados muestran que el método estocástico permite guiar la probabilidad con la que se permite que se infringan las restricciones del problema, aumentando la robustez de la solución y permitiendo que esta se ajuste a la realidad, ya que al considerar la incertidumbre de las variables, es capaz de establecer cuán fiable es la solución obtenida.

En cuánto al trabajo a futuro, todavía queda realizar un análisis del grado de optimalidad del GA implementado, para estudiar la probabilidad con la que las soluciones se acercan al

óptimo global. Por otro lado, aún se puede estudiar el comportamiento de esta nueva implementación, en instancias más grandes: que tengan un mayor número de clientes y que necesiten una flota de mayor tamaño para resolver el E-VRP. Además, también se puede realizar un análisis de comparación entre este método estocástico y uno estocástico que considere distribuciones de probabilidad normal sobre las variables que tienen incerteza. Finalmente, según lo mencionado en el Capítulo 3, el algoritmo implementado en esta memoria solo comprende la resolución de la pre-operación (*Off*-E-VRP), por lo que aún queda por desarrollar *On*-GA para la operación *online* del E-VRP.

Bibliografía

- [1] Salazar, B., “Indicadores Ambientales - Contribución del sector a las emisiones GEI Nacionales.” <https://energia.gob.cl/indicadores-ambientales-contribucion-del-sector-las-emisiones-gei-nacionales>, 2022. [Online; accedido el 03-Mayo-2022].
- [2] de Energía, M., “Plataforma de electromovilidad.” <https://energia.gob.cl/electromovilidad/introduccion/beneficios-de-la-electromovilidad>, 2022. [Online; accedido el 10-Mayo-2022].
- [3] Futalef, J. P., Muñoz-Carpintero, D., Rozas, H., y Orchard, M., “An online decision-making strategy for routing of electric vehicle fleets,” *Information Sciences*, 2023, doi: doi.org/10.1016/j.ins.2022.12.108.
- [4] Toth, P., *Vehicle routing: Problems, methods, and applications*. Society for Industrial and Applied Mathematics, 2 ed., 2014.
- [5] A. Alshamrani, K. M. y Ballou, R. H., “Reverse logistics: simultaneous design of delivery routes and returns strategies,” *Computers Operations Research*, vol. 34, no. 2, p. 595–619, 2007, doi:[10.1016/j.cor.2005.03.015](https://doi.org/10.1016/j.cor.2005.03.015).
- [6] “A web-based decision support system for waste lube oils collection and recycling,” *European Journal of Operational Research*, vol. 195, no. 3, pp. 676–700, 2009, doi: [10.1016/j.ejor.2007.11.004](https://doi.org/10.1016/j.ejor.2007.11.004).
- [7] “A flexible adaptive memory-based algorithm for real-life transportation operations: Two case studies from dairy and construction sector,” *European Journal of Operational Research*, vol. 179, no. 3, pp. 806–822, 2007, doi:[10.1016/j.ejor.2005.03.059](https://doi.org/10.1016/j.ejor.2005.03.059).
- [8] Erdelić, T. y Carić, T., “A survey on the electric vehicle routing problem: Variants and solution approaches,” *Journal of Advanced Transportation*, vol. 2019, pp. 1–48, 2019, doi:[10.1155/2019/5075671](https://doi.org/10.1155/2019/5075671).
- [9] Schildbach, G., Fagiano, L., y Morari, M., “Randomized solutions to convex programs with multiple chance constraints,” *SIAM Journal on Optimization*, vol. 29, no. 4, pp. 2479–2501, 2013.
- [10] Carr, J., “An introduction to genetic algorithms,” *Senior Project*, vol. 1, no. 40, p. 7, 2014.
- [11] Kramer, O., “Genetic algorithm,” en *Genetic Algorithm Essentials*, cap. 2, pp. 11–19, Springer International Publishing, 2017, doi:[10.1007/978-3-319-52156-5_2](https://doi.org/10.1007/978-3-319-52156-5_2).
- [12] Kumar, M., Husian, M., Upreti, N., y Gupta, D., “Genetic algorithm: Review and application,” *International Journal of Information Technology and Knowledge Management*, vol. 2, no. 2, pp. 451–454, 2010.
- [13] Kumar, A., “Encoding schemes in genetic algorithm,” *International Journal of Advanced*

Research in IT and Engineering, vol. 2, pp. 1–7, 2013.

- [14] Futalef, J. P., Muñoz-Carpintero, D., Rozas, H., y Orchard, M., “An evolutionary algorithm for the electric vehicle routing problem with battery degradation and capacitated charging stations,” Annual Conference of the Prognostics and Health Management Society, 2020, doi:10.36001/phmconf.2020.v12i1.1281.
- [15] Shao, S., Guan, W., Ran, B., He, Z., y Bi, J., “Electric vehicle routing problem with charging time and variable travel time,” Mathematical Problems in Engineering, 2017, doi:10.1155/2017/5098183.
- [16] Potvin, J.-Y., “State-of-the art review. evolutionary algorithms for vehicle routing,” INFORMS Journal on Computing, vol. 21, no. 4, pp. 518–548, 2009, doi:10.1287/ijoc.1080.0312.
- [17] Wang, Y., Bi, J., Guan, W., y Zhao, X., “Optimising route choices for the travelling and charging of battery electric vehicles by considering multiple objectives,” Transportation Research Part D: Transport and Environment, vol. 64, pp. 246–261, 2018, doi:10.1016/j.trd.2017.08.022.
- [18] Hiermann, G., Hartl, R. F., Puchinger, J., y Vidal, T., “Routing a mix of conventional, plug-in hybrid, and electric vehicles,” European Journal of Operational Research, vol. 272, no. 1, pp. 235–248, 2019, doi:10.1016/j.ejor.2018.06.025.
- [19] Munévar-Peña, M. A., Ramos-Cañón, A. M., y Prada-Sarmiento, L. F., “Incorporación de la incertidumbre aleatoria de las propiedades del suelo en la geoestadística. aplicación al diseño de cimentaciones basado en confiabilidad,” Boletín de Ciencias de la Tierra, pp. 38 – 48, 2018.
- [20] Ramos, A. y Cerisola, S., Optimización bajo incertidumbre. Madrid, España: Universidad Pontificia Comillas (Publicaciones), 2008.
- [21] Zhang, J., “Modern monte carlo methods for efficient uncertainty quantification and propagation: A survey,” Wiley Interdisciplinary Reviews: Computational Statistics, 2017, doi:0.1002/wics.1539.
- [22] Han, J., Lee, C., y Park, S., “A robust scenario approach for the vehicle routing problem with uncertain travel times,” Transportation Science, vol. 48, no. 3, pp. 373–390, 2013, doi:10.1287/trsc.2013.0476.
- [23] Sassi, O., Cherif, W. R., y Oulamara, A., “Vehicle routing problem with mixed fleet of conventional and heterogenous electric vehicles and time dependent charging costs,” 2014, doi:10.5281/zenodo.1099756.
- [24] Schneider, M., Stenger, A., y Goeke, D., “The electric vehicle-routing problem with time windows and recharging stations,” Transportation Science, vol. 48, no. 4, pp. 500–520, 2014, doi:10.1287/trsc.2013.0490.
- [25] Shao, S., Guan, W., Raa, B., He, Z., y Bi, J., “Electric vehicle routing problem with charging time and variable travel time,” Mathematica Problems in Engineering, vol. 2017, pp. 1–13, 2017, doi:10.1155/2017/5098183.
- [26] Montoya, A., Guéret, C., Mendoza, J. E., y Villegas, J. G., “The electric vehicle routing problem with nonlinear charging function,” Transportation Research Part B: Methodological, vol. 103, pp. 87–110, 2017, doi:10.1016/j.trb.2017.02.004.

- [27] Bi, X. y Tang, W. K. S., “Logistical planning for electric vehicles under time-dependent stochastic traffic,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3771–3781, 2019, [doi:10.1109/tits.2018.2883791](https://doi.org/10.1109/tits.2018.2883791).
- [28] Chu, J. C., Yan, S., y Huang, H. J., “A multi-trip split-delivery vehicle routing problem with time windows for inventory replenishment under stochastic travel times,” *Networks and Spatial Economics*, vol. 17, no. 1, pp. 41–68, 2015, [doi:10.1007/s11067-015-9317-3](https://doi.org/10.1007/s11067-015-9317-3).
- [29] Messaoud, E., “A chance constrained programming model and an improved large neighborhood search algorithm for the electric vehicle routing problem with stochastic travel times,” *Evolutionary Intelligence*, 2021, [doi:10.1007/s12065-021-00648-0](https://doi.org/10.1007/s12065-021-00648-0).
- [30] Basso, R., Kulcsár, B., Sanchez-Diaz, I., y Qu, X., “Dynamic stochastic electric vehicle routing with safe reinforcement learning,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 157, p. 102496, 2022, [doi:10.1016/j.tre.2021.102496](https://doi.org/10.1016/j.tre.2021.102496).
- [31] Ge, X., Zhu, Z., y Jin, Y., “Electric vehicle routing problems with stochastic demands and dynamic remedial measures,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–15, 2020, [doi:10.1155/2020/8795284](https://doi.org/10.1155/2020/8795284).
- [32] Lipschutz, S. y Lipson, M. L., *Matemáticas Discretas*, cap. Grafos dirigidos, pp. 201 – 234. The McGraw-Hill Companies, Inc, tercera ed., 2007.

Anexos

Anexo A. Ejemplo de codificación

A continuación, se presenta se ilustra el modo de operar de la codificación de individuos para el GA descrita en la Sección 3.4.2 y presentada por primera vez en [3].

Se considera una flota de tamaño $m = 2$, un máximo de $n^* = 3$ de operaciones de recarga (aquí no se utiliza la heurística presentada) y las siguientes rutas, planes de carga y tiempos de partida para cada EV:

$$\begin{aligned} S^1 &= [0, 5, 2, 4, 11, 7, 0] \\ L^1 &= [0, 0, 0, 0, 25.2, 0, 0] \\ x_1^1(0) &= 510.0 \quad (8 : 30AM) \end{aligned}$$

$$\begin{aligned} S^1 &= [0, 3, 1, 9, 12, 8, 6, 0] \\ L^1 &= [0, 0, 0, 0, 33.5, 0, 0, 0] \\ x_1^1(0) &= 540.0 \quad (9 : 00AM) \end{aligned}$$

Estas variables de decisión se pueden codificar usando el siguiente individuo candidato:

$$\left[\underbrace{5, 2, 4, 7, |, 3, 1, 9, 8, 6.}_{\text{Ruta 1} \quad \text{Ruta 2}}, \underbrace{9, 12, 33.5, -1, 11, 50.5, 4, 11, 25.5}_{\text{Bloque operaciones de carga}}, \underbrace{510.0}_{\text{Tpo. Partida 1}}, \underbrace{540.0}_{\text{Tpo. Partida 2}} \right]$$

Bloque secuencia de clientes Bloque tiempos de partida

Notar que como el primer valor del sub-bloque a^2 corresponde a -1 , esta operación de carga en particular no se efectúa.

Anexo B. Operaciones de crossover y mutación

A continuación, se describe cómo se realizan las operaciones de mutación y *crossover* para las poblaciones del GA, codificadas según los bloques de la Figura 3.2, conforme a lo descrito en [3]. Con la misma probabilidad, el GA selecciona uno de los tres bloques (secuencia de clientes, operaciones de carga o tiempos de partida) del individuo, para luego realizar mutación o *crossover* según el tipo de bloque. Además, las operaciones se realizan con una probabilidad de ocurrencia.

Bloque de secuencia de clientes

• Mutación

- Se elige una secuencia de clientes al azar. Luego se intercambian dos clientes en esa secuencia. (Probabilidad: 0.4)
- Se intercambian dos valores aleatorios en el bloque. (Probabilidad: 0.4)
- Elegir un cliente aleatorio en una secuencia de clientes. Luego, eliminar ese cliente de la secuencia e insertarlo en una secuencia al azar. (Probabilidad: 0.2)

• Crossover

- Se combinan dos secuencias de clientes seleccionadas al azar, según lo siguiente: primero se encuentran los clientes que se repiten entre las secuencias; luego se eliminan los clientes repetidos del individuo; a continuación, los clientes repetidos se insertan en otras secuencias de clientes en una posición aleatoria; y finalmente, las secuencias originales se intercambian entre los individuos. (Probabilidad: 1.0)

Bloque de operaciones de carga

• Mutación

- Primero, se elige un operación de carga al azar a^j . Luego, se muestrean tres valores: a^* de $N \cup \{-1\}$, b^* de F (conjunto de nodos estaciones de carga), y c^* de $U(-10, 10)$. Después, la operación de carga se modifica según lo siguiente:

$$a^j = [a_1^j + a^*, a_2^j + b^*, |a_3^j + c^*|]$$

(Probabilidad: 0.8)

- Primero se selecciona al azar una operación de carga a^j . Luego, se muestrean tres valores: a^* de $N \cup \{-1\}$, b^* de F (conjunto de nodos estaciones de carga), y c^* de $U(20, 30)$. Después, la operación de carga se modifica según lo siguiente:

$$a^j = [a^*, b^*, c^*]$$

(Probabilidad: 0.2)

• Crossover

- Permuta dos operaciones de carga en las mismas posiciones entre dos individuos. (Probabilidad: 1.0)

Bloque de tiempos de partida

- **Mutación**

- Elige un índice aleatorio i en el rango de la longitud del bloque. Si el bloque contiene el valor d_i , este valor se reemplaza por $|d_i + a^*|$, donde a^* se muestrea de $U(-60, 60)$. (Probabilidad: 1.0)

- **Crossover**

- Permuta dos valores en una posición seleccionada al azar. (Probabilidad: 1.0)