Aprendizaje por refuerzo multiagente para la optimización de rutas del transporte público.

Ahumada García, Fermin y Durazo Duarte, Chelsea.

Licenciatura en ciencias de la computación de la Universidad de Sonora.

A 22 de mayo de 2024.

Resumen: Este proyecto nace con el objetivo de rediseñar las rutas actuales de la ciudad de Hermosillo, Sonora. Este problema ha sido abordado en la literatura utilizando técnicas de programación dinámica o algoritmos genéticos. Un enfoque reciente de resolución sería el de aplicar Multi-Agent Reinforcement Learning (MARL) e inteligencia artificial cooperativa para encontrar las soluciones adecuadas y equilibrar la ganancia entre operador y usuarios. En este proyecto, se utilizará MARL para resolver el problema del diseño de rutas en una ciudad como Hermosillo y posteriormente el establecimiento de frecuencias de las unidades.

En un análisis previo, se realizó una recolección de encuestas a los usuarios de tres rutas del transporte público en Hermosillo, mostrando que la mayoría de los usuarios consideran que el servicio es regular y estarían dispuestos a pagar una tarifa más alta por un mejor servicio. Además, se identificaron los largos tiempos de espera, que serán abordados mediante la optimización de las rutas y frecuencias utilizando MARL.

1. INTRODUCCIÓN

El transporte público es de suma importancia en la movilidad urbana, siendo el principal medio de desplazamiento para millones de personas en áreas urbanas densamente pobladas como Hermosillo, que según el Censo de Población y Vivienda 2020, levantado por el Instituto Nacional de Estadística y Geografía (INEGI), contaba con 936,263 habitantes en 2020. Sin embargo, la calidad y eficiencia de este servicio pueden variar considerablemente debido a diversos factores, desde la planificación de rutas hasta el estado de las unidades y el comportamiento de los conductores.

En esta situación, el aprendizaje por refuerzo emerge como una herramienta poderosa para mejorar la optimización de las rutas del transporte público. Esta técnica de aprendizaje automático permite a un agente tomar decisiones secuenciales en un entorno dinámico, buscando maximizar una señal de recompensa a largo plazo. Aplicar el aprendizaje por refuerzo al diseño y operación de rutas de transporte público puede significar mejoras notables en la eficiencia, puntualidad y satisfacción de los usuarios.

Esta técnica es innovadora ya que en la literatura usualmente se ha abordado el problema con técnicas convencionales como metaheurísticas, algoritmos genéticos, recocido simulado, búsqueda tabú, GRASP, o colonia de hormigas. Además, el uso de inteligencia artificial (IA) para la solución de problemas permite abordar cuestiones complejas y dinámicas de manera más efectiva.

En un análisis realizado en 2018, se llevaron a cabo encuestas a los usuarios de tres rutas del transporte público en Hermosillo. De las 1,286 encuestas realizadas, se encontró que el 36.39% utiliza la Ruta 1, el 27.14% la Ruta 9 y el 36.47% la Ruta 18. La mayoría de los usuarios (63.3%) tenían entre 18 y 25 años, y el 58.98% eran estudiantes. Un 61.36% utilizaba el transporte para ir a estudiar, y el 33.90% de los encuestados reportó que su viaje duraba 20 minutos.

El servicio de transporte público en Hermosillo en 2018 enfrentaba varios desafíos, como el mal estado físico de las unidades, largos tiempos de espera y la falta de respeto de los choferes a las paradas. El 52.29% de los usuarios consideraba que el servicio era regular, y el 40.75% esperaba alrededor de 20 minutos por el transporte. La tarifa actual era considerada razonable por el 60.02% de los usuarios, y el 73.88% estaría dispuesto a pagar una tarifa de $8.00 si se mejorara el servicio.

Comparando esta situación con la actualidad en 2024, se observa la necesidad de implementar soluciones

avanzadas para mejorar el sistema de transporte público. El aprendizaje por refuerzo multi-agente (MARL) se propone como una solución avanzada para optimizar el sistema de transporte público en Hermosillo. En lugar de tratar cada ruta de manera aislada, MARL permite a múltiples agentes (rutas) interactuar y coordinarse en un entorno compartido, maximizando la eficiencia del sistema global. MARL puede abordar tanto la cooperación como la competencia entre rutas, mejorando la planificación y operación del transporte público de manera integral.

Los objetivos de nuestro proyecto incluyen evaluar la viabilidad de la aplicación de MARL en comparación con otros métodos, como las metaheurísticas, y explorar el aprendizaje en esta área de la inteligencia artificial. Al hacer esto, buscamos proporcionar un sistema de transporte público más eficiente, puntual y satisfactorio para los usuarios, abordando los desafíos actuales del servicio y beneficiando a la comunidad urbana en general.

2. ¿QUÉ SON LAS MARL?

Multi-Agent Reinforcement Learning (MARL) es una extensión del aprendizaje por refuerzo (Reinforcement Learning, RL) que involucra múltiples agentes interactuando en un entorno compartido. En MARL, cada agente aprende a tomar decisiones para maximizar su propia recompensa, mientras tiene en cuenta las acciones y políticas de otros agentes. Esto permite la cooperación y coordinación entre agentes para resolver problemas complejos y dinámicos de manera más efectiva.

Inicialmente, el proyecto planeaba abordar el problema del diseño de rutas del transporte público en Hermosillo utilizando el aprendizaje por refuerzo. Sin embargo, dado que el problema involucra múltiples rutas y la optimización de cada una de ellas podría ser un proceso lento y menos eficiente, se decidió emplear MARL. Esta técnica permite maximizar la recompensa total del entorno de manera más eficiente, al considerar las interacciones y colaboraciones entre diferentes rutas y agentes.

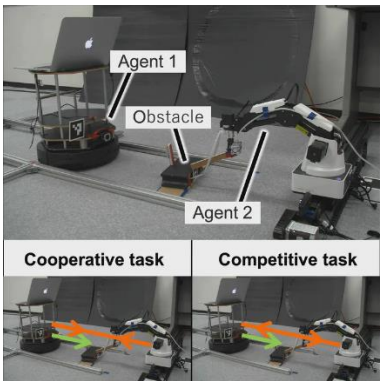


Fig. 1Ejemplo de MARLS

MARL se clasifica en tres categorías principales:

Cooperativo: Los agentes trabajan juntos hacia un objetivo común.

Competitivo: Los agentes compiten entre sí para alcanzar sus metas.

Mixto: Combina cooperación y competencia, como en un partido de baloncesto, donde los jugadores de un equipo colaboran, pero compiten contra el equipo contrario.

3. MARCO TEÓRICO

Aprendizaje por Refuerzo:

El aprendizaje por refuerzo (RL) es una técnica de aprendizaje automático orientada a objetivos desarrollada para guiar a un agente autónomo a maximizar la recompensa del agente a lo largo de repeticiones de procedimientos computacionales en un determinado entorno

1. Agente: El agente aprende y realiza acciones interactuando con un entorno.
2. Entorno: Todo lo que está fuera del agente y con el que interactúa.
3. Acción a: El conjunto de acciones disponibles para el agente.
4. Estado (S): La representación del entorno en un momento dado.
5. Factor de descuento (γ): El factor de descuento se utiliza para ponderar la importancia de las recompensas a largo plazo frente a las recompensas inmediatas.
6. Recompensa (R): La retroalimentación recibida por el agente por sus acciones, basada en cuánto se acerca a los objetivos de optimización predefinidos.

Q-learning

A medida que el agente interactúa con el entorno en una secuencia de pasos, el entorno se formula en base al proceso de decisión de Markov...

La asignación del agente de estado a acción se llama política π, que se mejora de manera iterativa a medida que el agente gana experiencia. Durante la fase de entrenamiento, se ejecutan numerosas pruebas para que el agente adquiera experiencia y encuentre, la mejor política, que determina la mejor acción cuando el agente está en un estado particular.

La actualización de la tabla de valores Q se realiza utilizando la siguiente ecuación de un paso del Q-learning:

$$Qi_{new}(s,a_1,a_2,\dots,a_n) \leftarrow Qi_{old}(s,a_1,a_2,\dots,a_n) + \alpha(ri\ + \gamma maxQi(s',a'_1,a'_2,\dots,a'_n) - Qi_{old}(s,a_1,a_2,\dots,a_n))$$

Qi_{new} es el nuevo valor de Q del siguiente estado s' bajo la acción a en el tiempo t

Qi_{old} es el valor Q registrado previamente del estado s' bajo la acción a en el tiempo t

α es la tasa de aprendizaje en el tiempo $t(0 \leq \alpha \leq 1)$

γ es la tasa de aprendizaje en el tiempo $t(0 \leq \gamma \leq 1)$

$maxQi(s', a'_1, a'_2, \dots, a'_n)$ representa el máximo valor de Q entre las acciones posibles del siguiente estado s'

r_i *es la recompensa recibida por el agente i*

Q table

La tabla en MARL es una estructura de datos que mantiene la estimación de la utilidad de tomar una determinada acción en un estado particular, considerando las acciones de todos los agentes. Si hay N estados y M acciones posibles por agente, la tabla Q tendrá las dimensiones $N \times M1 \times M2 \dots \times M_N$ donde cada celda $(s, a_1, a_2, \dots, a_n)$ representa el valor Q para el estado s y las acciones $a_1, a_2, \dots, a_n$ de los agentes.

La representación de la tabla Q se puede entender como una matriz donde las filas representan los estados posibles y las columnas representan las acciones posibles que el agente puede tomar en esos estados. Cada celda de la tabla contiene el valor Q, que es una estimación de la utilidad de tomar esa acción en ese estado específico.

$$\boldsymbol{Q(s,a)} = \begin{bmatrix} \boldsymbol{Q(s_1, a'_{1,1}, a'_{2,1}, \dots, a'_{n,1})} & \boldsymbol{Q(s_1, a'_{1,2}, a'_{2,2}, \dots, a'_{n,2})} & \dots & \boldsymbol{Q(s_1, a'_{1,M1}, a'_{2,M2}, \dots, a'_{n,Mn})} \\ \boldsymbol{Q(s_2, a'_{1,1}, a'_{2,1}, \dots, a'_{n,1})} & \boldsymbol{Q(s_2, a'_{1,2}, a'_{2,2}, \dots, a'_{n,2})} & \dots & \boldsymbol{Q(s_2, a'_{2,M1}, a'_{2,M2}, \dots, a'_{n,Mn})} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{Q(s_N, a'_{1,1}, a'_{2,1}, \dots, a'_{n,1})} & \boldsymbol{Q(s_N, a'_{1,2}, a'_{2,2}, \dots, a'_{n,2})} & \dots & \boldsymbol{Q(s_N, a'_{1,M1}, a'_{2,M2}, \dots, a'_{n,Mn})} \end{bmatrix}$$

$s_1, s_2, \dots\dots, s_N$ representan los estados posibles

$a_1, a_2, \dots\dots, a_M$ representan las acciones posibles

$Q(s_i, a_j)$ es el valor Q para el estado s_i y la a_j

MDP

Procesos de decisión de Markov (MDP)

Un proceso de decisión de Markov es un marco matemático de toma de decisiones utilizado para modelar problemas estocásticos en los que se toma decisiones basándose en un modelo predefinido (o política). Tal decisión llevaría al agente de un estado a otro basándose en una

distribución de probabilidad, que depende en gran medida de cuándo se tomó la decisión.

Formalmente, un MDP es una tupla (S, A, P_t, R_a) donde:

• S es un conjunto finito de estados

• A es un conjunto finito de acciones

• $P_t$es la probabilidad de pasar del estado s al s' después de una acción a en el tiempo t

• R_a es la recompensa inmediata recibida debido a la transición de un estado a otro.

4. METODOLOGÍA

Abordaremos el problema de TNDFSP (Transit Network Design and Frequency Setting Problem – Diseño de la Red de Tránsito y Establecimiento de Frecuencias). El problema se dividirá en dos partes: primero, el diseño de la red de transporte y, segundo, el establecimiento de frecuencias. El problema de establecimiento de frecuencias requiere de una red de tránsito ya existente sobre la cual se asignarán demandas entre las paradas, con el objetivo de minimizar el tiempo total de viaje para cada pasajero.

Diseño de la Red de Transporte

Para el diseño de la red de transporte, se seguirá un enfoque sistemático que incluirá los siguientes pasos:

1. Recolección y Análisis de Datos:

Los datos geográficos de las paradas que fueron obtenidos y analizados. Estos datos incluirán la latitud

y longitud de cada parada, permitiendo su representación en un sistema de coordenadas.

Se identificará la ubicación exacta de cada una de las 2927 paradas de la ciudad, como se muestra en el ejemplo con paradas como Oscar Olea y Vista Real, Lopez Riesgo y Dr. Oscar Olea, y Av. Hacienda de los Rios / Hacienda los Mezquites.

2. Construcción del Grafo de la Red de Transporte:

Se representarán las paradas como nodos en un grafo, y las posibles conexiones entre paradas como arcos.

Utilizando un algoritmo de distancia, se calcularán las distancias entre cada par de paradas. Cada parada será conectada a un máximo de cinco nodos cercanos, asegurando una red suficientemente densa pero manejable.

3. Optimización de la Red:

La red de transporte se optimizará eliminando redundancias y asegurando que todas las paradas estén eficientemente conectadas.

Esta optimización considerará tanto la conectividad como la minimización de los tiempos de viaje entre paradas.

Establecimiento de Frecuencias

Para el establecimiento de frecuencias, se buscará optimizar el tiempo total de viaje para los pasajeros. El proceso incluirá:

1. Análisis de Demanda:

Se analizarán los patrones de demanda en diferentes momentos del día y en diferentes días de la semana para determinar las necesidades de transporte.

Al no contar con una demanda real, un ejemplo de una seria la evaluación en función de los datos históricos de uso del transporte público, encuestas y otros estudios relevantes.

2. Determinación de Capacidades:

Se evaluará la capacidad de cada autobús y la frecuencia requerida para satisfacer la demanda sin causar sobrecarga.

Se considerarán variables como la capacidad de asientos, la frecuencia de paso y la distribución de pasajeros a lo largo del día.

3. Implementación de MARL (Multi-Agent Reinforcement Learning):

En lugar de utilizar algoritmos tradicionales como los genéticos, recocido simulado, búsqueda tabú, GRASP o colonias de hormigas, se implementará el aprendizaje por refuerzo multiagente (MARL).

MARL permite que múltiples agentes (por ejemplo, rutas de autobuses) aprendan y optimicen sus políticas de manera simultánea, considerando tanto sus propios objetivos como la interacción con otros agentes.

Cada agente (ruta) aprenderá a ajustar sus frecuencias de servicio para maximizar la eficiencia y minimizar el tiempo total de viaje de los pasajeros.

4. Optimización de Frecuencias:

Los agentes utilizarán técnicas de aprendizaje por refuerzo para ajustar las frecuencias de los autobuses, minimizando el tiempo de espera en las paradas y el tiempo de viaje total.

La optimización se realizará mediante iteraciones y simulaciones, donde los agentes aprenden de sus interacciones con el entorno y con otros agentes para mejorar sus decisiones.

5. Implementación y Validación:

Las frecuencias optimizadas se implementarán en un entorno de simulación para validar su eficacia.

Se realizarán ajustes basados en los resultados de la simulación para asegurar que las frecuencias propuestas cumplan con los objetivos de minimización del tiempo de viaje y satisfacción del usuario.

Esta metodología permitirá diseñar y operar una red de transporte público eficiente, mejorando la conectividad y la experiencia del usuario en Hermosillo mediante el uso de técnicas avanzadas de inteligencia artificial como MARL.

5. Diseño de la Red de Transporte:

Se busca crear una red de rutas conectadas que satisfaga las necesidades de conectividad de la ciudad, y se puede expresar como

$$G(N, E)$$

donde N es un conjunto de nodos que representan las paradas, y E es un conjunto de arcos que representan los caminos disponibles. En este caso, los datos de las paradas fueron proporcionados por el Sistema integral de Transporte, se limpió todo el data que se nos entregó y nos quedamos con los siguientes campos: Nombre, Latitud, y Longitud. Cada uno de estos nodos corresponde a una parada específica dentro de la ciudad de Hermosillo.

Datos de las Paradas

El archivo CSV proporcionado contiene un total de 2927 registros de paradas, cada uno con su nombre, latitud y longitud. Estos datos son cruciales para la planificación y optimización de la red de transporte público. A continuación, se muestra un ejemplo de los datos:

Oscar Olea y Vista Real: (Latitud: 28.996039, Longitud: -110.941957)

Lopez Riesgo y Dr. Oscar Olea: (Latitud: 28.996147, Longitud: -110.943611)

Av. Hacienda de los Rios / Hacienda los Mezquites: (Latitud: 28.996389, Longitud: -110.968702)

Estos puntos representan las ubicaciones geográficas exactas de las paradas, lo que permite modelar la red de transporte de manera precisa. Sin embargo, los datos iniciales representan solo puntos individuales sin conexiones entre ellos. Para transformar estos puntos en una red de rutas conectadas, se realizó un algoritmo que conecta cada parada con un máximo de cinco nodos cercanos.

Algoritmo de Conexión de Paradas

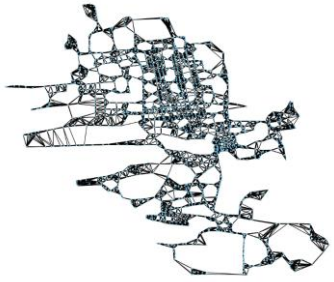
Para construir la red de transporte, se implementó un algoritmo de conexión que une cada parada con sus nodos colindantes más cercanos. El proceso se realizó de la siguiente manera:

Cálculo de Distancias: Se calcularon las distancias entre cada par de paradas utilizando la fórmula de distancia euclidiana basada en las coordenadas de latitud y longitud.

Selección de Nodos Cercanos: Para cada parada, se identificaron los cinco nodos más cercanos. Este límite de cinco conexiones por parada asegura que la red sea suficientemente densa para mantener la conectividad, pero no tan densa que complique innecesariamente la planificación y operación.

Creación de Arcos: Una vez identificados los nodos más cercanos, se crearon arcos (conexiones) entre las paradas y sus vecinos seleccionados. Cada arco representa un posible camino disponible en la red de transporte.

Optimización de la Red: La red resultante se optimizó para eliminar redundancias y asegurar que todas las paradas estén conectadas de manera eficiente, minimizando el tiempo total de viaje para los pasajeros.



6. RESTRICCIONES DEL PROBLEMA

Tomaremos algunas de las restricciones utilizadas por Ahmed Darwish et. al. en su artículo *"Optimising Public Bus Transit Networks Using Deep Reinforcement Learning"* [2]

1. La red resultante es un grafo conectado, refiriendo que cualquier parada puede conectar con cualquier otra dentro de la red.
2. Todas y cada una de las paradas deberán pertenecer al menos a una de las rutas.
3. Las rutas deben ser diferentes entre sí.
4. El número de las rutas totales es predefinido para cada ejecución.
5. Todos los autobuses tienen la misma capacidad.
6. No se repiten paradas dentro de una ruta. Estas deben comenzar en un punto y terminar en otro.
7. Cada ruta tendrá un límite de distancia máxima para resguardar el tiempo de conducción sano de los conductores.

7. DEMANDA

La matriz de origen destino para la demanda de pasajeros se puede expresar como:

$$D = \{d_{ab} | a, b \in N\}$$

Donde d_{ab} es el número de pasajeros que viajan de la parada *a* a la parada *b*.

Dado que no tenemos una demanda real, para hacer las pruebas se creo una matriz de unos para asegurar una demanda mínima de uno de cualquier parada a otra.

Cálculo de Recompensas

La demanda de pasajeros se integra en el cálculo de recompensas para los agentes. Cuando un agente se mueve de una parada a otra, el modelo ajusta la recompensa en función de la demanda recogida en esa ruta específica. Si hay demanda entre dos paradas, se otorga una bonificación al agente por satisfacer esa demanda. Una vez que la demanda es recogida, se elimina de la matriz para indicar que la demanda ha sido satisfecha.

Bonificación por Recoger Demanda

La bonificación se aplica cuando un agente recoge pasajeros entre dos paradas. Esto incentiva a los agentes a seguir rutas que maximicen la satisfacción de la demanda, reflejando un comportamiento más eficiente y orientado al servicio.

Penalización por Distancia y Repetición

Adicionalmente, se imponen penalizaciones por la distancia recorrida y por la repetición de nodos. Estas penalizaciones se contrarrestan con las bonificaciones por la demanda recogida, equilibrando el comportamiento de los agentes para que no solo minimicen la distancia recorrida, sino que también optimicen la satisfacción de la demanda.

Este enfoque asegura que los agentes en el modelo siempre tengan una demanda mínima que satisfacer.

En este proyecto, actualmente no se cuenta con datos de demanda real. Sin embargo, se ha creado una matriz de unos para asegurar una demanda mínima entre todas las paradas, lo que permite realizar pruebas y validar el funcionamiento del modelo. Esta aproximación simplificada permite seguir explorando y desarrollando el proyecto. En el futuro, se pueden integrar datos de demanda reales o simulados, ajustando y refinando el modelo para evaluar su rendimiento y optimización de manera más precisa.

8. DISEÑO DE LAS SOLUCIONES

Obtendremos como resultado una lista de n listas con una longitud máxima dada en las que cada una de las listas de dentro representará cada una de las n lineas establecidas en el inicio. Por ejemplo, si decidimos crear una red de 3 líneas para una ciudad de 9 paradas, una solución válida podría ser la siguiente:

[[1, 2, 3], [2, 4, 6, 8], [1,6, 7, 5, 4, 9]]

Un estado final dentro de nuestro modelo podría ser cualquier solución valida.

Estado Final en el Modelo

Un estado final en este contexto es cualquier solución válida que cubra las paradas según las reglas establecidas. Es decir, cada agente debe formar una ruta que:

Conectividad: Las paradas en la ruta deben estar conectadas por aristas válidas en el grafo de la ciudad.

Cobertura Completa: Idealmente, todas las paradas deben ser cubiertas por al menos una línea (agente) en la solución final.

Cumplimiento de Reglas de Recompensas y Penalizaciones: La solución debe ser evaluada y ajustada según las recompensas y penalizaciones definidas en el modelo, que incluyen factores como la distancia recorrida, la satisfacción de la demanda, y la diversificación de rutas.

Este enfoque asegura que la red de transporte diseñada no solo cubre todas las paradas, sino que también lo hace de manera eficiente, optimizando la satisfacción de la demanda y minimizando los costos operativos.

9. FUNCIÓN DE RECOMPENSA

Para recompensar las acciones de cada agente se tomaron en cuanta cada uno de los siguientes puntos:

1. Recompensa si el grafo creado por los agentes está conectado.
2. Recompensa si el estado es final, es decir que todas las paradas pertenecen al menos a un agente.
3. Penalización por la distancia recorrida.
4. Recompensa por la demanda satisfecha.
5. Penalización si hay agentes que comparten una cantidad significable de paradas.
6. Penalización por seleccionar una parada que ya estaba dentro del agente.

7. Penalización si la ruta termina muy cerca del punto de partida
8. Recompensa por visitar un nodo no explorado previamente
9. Penalización Significativa por No Encontrar una Arista Válida
10. Penalización Adicional por Repetición
11. Penalización por Estar Cerca del Punto de Partida

Este conjunto de recompensas y penalizaciones está diseñado para equilibrar la eficiencia y la eficacia del sistema, optimizando tanto la cobertura del grafo como la satisfacción de nuestro problema

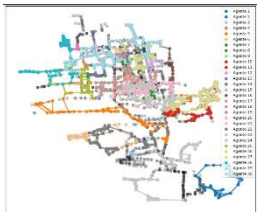
10. MULTIAGENT DEEP Q-LEARNING

Implementamos una libreta para ejecutar un entorno con las especificaciones dadas previamente junto con una implementación de Deep Q-Learning multiagente. El código se basó en la libreta de GitHub publicada por Moya (2024)[3], que consiste en una implementación de Deep Q-Learning.

Modificamos esta implementación para adaptarla a nuestra necesidad de trabajar sobre un problema multiagente cooperativo, y como resultado obtuvimos una libreta publicada en GitHub [4].

Los resultados obtenidos fueron procesados en un entorno de Google Colab y serán presentados a continuación.

11. RESULTADOS

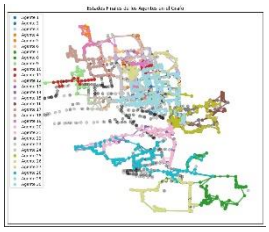


Configuración de Parámetros:

Número de Agentes: 30.

Steps por Agente: 100.

Total de Episodios: 50.



Configuración de Parámetros:

Número de Agentes: 30.

Steps por Agente: 160.

Total de Episodios: 50.

Selección de Parámetros:

Los parámetros fueron seleccionados basándose en la necesidad de explorar y cubrir todas las paradas en el grafo de la red de transporte.

Se compara la selección de parámetros que al aumentar el reward significativamente como el ejemplo 2 el alcance es menor por que cubre la penalización.

Resultados de la Simulación:

La simulación mostró que al utilizar MARL, los agentes fueron capaces de coordinarse eficientemente para cubrir una gran parte de las paradas en la red de transporte.

Sin embargo, no se logró abarcar la totalidad del grafo debido a que faltaron algunos nodos por conectar, lo cual sugiere que el número de episodios fue insuficiente para una cobertura completa.

Análisis de Desempeño:

Cobertura de Paradas: Aunque todos los agentes lograron cubrir una gran parte de las paradas, no se alcanzó una cobertura total del grafo. Esto indica que, si bien el enfoque es prometedor, se necesita más tiempo de entrenamiento o una mayor cantidad de episodios para alcanzar una cobertura completa.

Optimización de Frecuencias: Los agentes aprendieron a ajustar sus frecuencias de servicio de manera óptima, reduciendo el tiempo de espera en las paradas y mejorando la eficiencia general.

Interacciones entre Agentes: Los agentes demostraron una capacidad efectiva para coordinarse y evitar redundancias en la cobertura de rutas, lo que es crucial para la eficiencia operativa.

La implementación de MARL permitió a los agentes encontrar soluciones óptimas para la cobertura de paradas y el establecimiento de frecuencias en una red de transporte compleja. Sin embargo, los resultados también revelaron que, aunque se satisfizo una gran parte del grafo, no se logró abarcarlo completamente debido a la insuficiencia en la cantidad de episodios y el número limitado de pasos por agente. Esto sugiere que, para una implementación futura, sería necesario incrementar el número de episodios y posiblemente ajustar otros parámetros de entrenamiento.

A pesar de estas limitaciones, los resultados son prometedores y proporcionan una base sólida para futuras investigaciones. La aplicación práctica de MARL en el diseño y operación de redes de transporte urbano puede ser de apoyo a la planificación y gestión de los sistemas de transporte público, mejorando significativamente la eficiencia y la satisfacción de los usuarios.

12. CONCLUSIONES

Este proyecto ha sido un buen comienzo como prototipo para abordar un problema real de diseño de rutas en el transporte público. Hemos demostrado que es posible aplicar el aprendizaje por refuerzo multi-agente (MARL) para optimizar tanto el diseño de la red de transporte como el establecimiento de frecuencias. A través de esta metodología, se ha logrado crear una red de rutas conectadas que satisface las necesidades de conectividad de la ciudad y se ha optimizado el tiempo total de viaje para los pasajeros.

El desarrollo de este proyecto ha permitido establecer una base sólida para abordar el desafío del diseño de rutas en el transporte público de manera efectiva. Al aplicar el aprendizaje por refuerzo multi-agente (MARL), hemos logrado optimizar tanto el diseño de la red de transporte como el establecimiento de frecuencias, alcanzando una red de rutas conectadas que mejora significativamente la conectividad urbana y reduce el tiempo total de viaje para los pasajeros.

El objetivo principal de evaluar la viabilidad de la aplicación de MARL en comparación con otros métodos tradicionales se ha cumplido. Hemos podido comprobar que MARL puede manejar de manera efectiva la complejidad y la dinámica del sistema de transporte público, lo cual hasta ahora solo se había explorado en aplicaciones teóricas, pero no en contextos prácticos.

Como trabajo futuro, se puede considerar probar el aprendizaje por refuerzo utilizando un modelo individual para cada agente, mejorando así las decisiones individuales que toma cada uno. Esto puede ser considerado como una metaheurística o simplemente como una exploración de los distintos resultados que se pueden obtener con este cambio.

Además, sería una gran mejora implementar una demanda real para cada una de las paradas. Esto permitiría ajustar las frecuencias de los autobuses de manera más precisa, alineándolas con la demanda real de los pasajeros. La integración de datos de demanda real mejoraría aún más la eficiencia y la satisfacción del usuario.

El proyecto ha demostrado que la implementación de MARL en el diseño y operación del transporte público es no solo viable, sino también beneficiosa. Esta técnica avanzada de inteligencia artificial puede revolucionar la forma en que se planifican y gestionan los sistemas de transporte público, proporcionando una base sólida para futuras investigaciones y desarrollos en este campo.

13. REFERENCIAS

[1] D. Moctezuma García, «Implementación y análisis de estrategias para optimización de redes de transporte público considerando variaciones de precio», Tesis de maestría, Universidad de Colima, Colima, México, 2018.

[2] K. Darwish, A., Khalil, M., & Badawi, «Optimising Public Bus Transit Networks Using Deep Reinforcement Learning.», *IEEE Xplore*, 2018.

[3] R. Moya, «2_02_Deep_Q-Learning.ipynb», GitHub, 2024. [En línea]. Disponible en: https://github.com/RicardoMoya/Reinforcement_Learning_with_Python

[4] C. Durazo Duarte y F. A. Ahumada Garcia, «Multiagent Deep Q-learning For Transport Network Design Problem», GitHub, 2024. [En línea]. Disponible en: https://github.com/chelseadz/MultiagentDeepQ-learningForTransportNetworkDesignProblem

[5] K. V. Krishna Rao, S. Muralidhar, y S. L. Dhingra. Public transport routing and scheduling using Genetic Algorithms.En 8th International Conference on Computer AidedScheduling of Public Transport, Berlin, Alemania, 2000.