

UNIVERSIDAD NACIONAL DE INGENIERÍA
ESCUELA DE CIENCIA DE LA COMPUTACIÓN



**UNIVERSIDAD
NACIONAL DE
INGENIERÍA**

**Enrutamiento óptimo para el envío de pedidos
compartidos basado en aprendizaje por refuerzo**

Seminario de tesis:
Ciencia de la Computación

AUTOR

Brando Miguel Palacios Mogollon

Asesor

César Jesús Lara Avila

LIMA-PERU

2020

Agradecimientos

Dedicado a mi padre y a mi madre por su apoyo incondicional a lo largo de la carrera.

A mi asesor César Lara Avila por su guía y compromiso para el desarrollo de la presente tesis.

A la universidad y a todos mis docentes, quienes me convirtieron en un profesional preparado para asumir todo tipo de desafíos.



Resumen

Debido a la aparición de la pandemia, tanto conductores como pasajeros han cambiado la forma de elegir su transporte, haciendo más recurrente el servicio de transporte privado. Al decidir entre todas las posibilidades de los servicios de transporte, los pasajeros consideran una serie de criterios en base a su condición y conveniencia, priorizando que el viaje dure lo más corto posible. El viaje compartido, ha surgido como solución potencial para reducir el tiempo de espera, costo y duración del viaje. Sin embargo, un problema central para tales servicios es la planificación de rutas para que cada conductor cumpla con las solicitudes y condiciones que satisfacen un buen enrutamiento entre los pasajeros.

En el presente trabajo de investigación, abordamos una forma diferente de obtener el enrutamiento de viajes compartidos identificando algunas condiciones que involucran este tipo de servicio; con el objetivo de mejorar su calidad, realizamos una modelo de aprendizaje por refuerzo profundo (DRL) que permita la selección de los pedidos con punto de recojo y destino más cercanos representados por un grafo dirigido cuyos pesos están designados bajo criterios de la demanda, tiempo de espera y tráfico vehicular. Nuestro enfoque se valida utilizando millones de viajes regulares extraídos del conjunto de datos públicos de Chicago Taxi Trips.

Palabras claves: viajes compartidos, aprendizaje por refuerzo profundo, transporte, grafos, enrutamiento

Abstract

Due to the emergence of the pandemic, drivers and passengers have changed the way they choose their transportation, making the private transportation service more recurrent. When deciding between all the possibilities of transportation services, passengers consider a series of criteria based on their condition and convenience, prioritizing the shortest possible travel time and cost. Ridesharing has emerged as a solution to reduce waiting time, cost and travel time. However, a central problem for such services is the planning of routes so that each driver meets requests for conditions that satisfy good routing among passengers. In the present research work, we address a different way to obtain ridesharing routing by identifying some conditions involving this type of service; with the objective of improving its quality, we perform a deep reinforcement learning (DRL) model that allows the selection of orders with closest pickup point and destination represented by a directed graph whose weights are designated on demand, waiting time and vehicular traffic. Our approach is validated using millions of regular trips extracted from the Chicago Taxi Trips public dataset.

Palabras claves: ridesharing, deep reinforcement learning, transportation, routing, graph

Índice general

1. Generalidades	8
1.1. Problemática	8
1.2. Objetivos	10
1.2.1. Objetivo general	10
1.2.2. Objetivos específicos	10
1.2.3. Resultados	10
1.3. Alcance y limitaciones	11
1.3.1. Hipótesis	11
1.3.2. Alcance	11
2. Marco teórico	12
2.1. Viajes compartidos	12
2.1.1. Precios	13
2.1.2. Patrones y variables	14
2.1.3. Beneficios y riesgos	16
2.2. Aprendizaje por refuerzo	16
3. Estado de arte	19
3.1. Objetivo de la revisión	19
3.2. Método usado en la revisión	19
3.3. Estado del Arte	19

4. Metodología	22
4.1. Base de datos	22
4.1.1. Conjunto de datos Chicago Taxi Trips	22
4.1.2. Boundaries Community Areas	24
4.1.3. Estructura del conjunto de datos	24
4.2. Preprocesamiento	25
4.3. Definición del modelo	26
5. Experimentación y Resultados	28
5.1. Flujo conceptual	28
5.2. Estructura del modelo	29
5.3. Métricas	29
5.4. Resultados	30
6. Conclusiones	33
6.1. Conclusiones	33
6.2. Trabajos futuros	33

Índice de figuras

2.1. Ilustración del patrón 1, viaje idéntico.	14
2.2. Ilustración del patrón 2, viaje incluido.	15
2.3. Ilustración del patrón 3, donde, (i). $u_b = pk_b$, (ii). $v_b = dp_b$	15
2.4. Ilustración del patrón 4, viaje por desvío.	16
2.5. Ilustración de la interacción entorno-agente.	17
4.1. Ilustración del grafo dirigido pick-drop.	26
5.1. Ilustración del flujo de conexión del proyecto.	28
5.2. Ilustración de la estructura del modelo.	29
5.3. Gráfica de resultado de la distancia promedio durante las épocas	30
5.4. Gráfica de resultado del costo durante las épocas	31
5.5. Ilustración de los nodos generados en QGIS	31
5.6. Ilustración del grafo con el camino propuesto por el modelo QNet usando el grafo pick-drop	32

Generalidades

1.1. Problemática

A día de hoy existen muchas formas de movilizarse en las ciudades más urbanizadas, entre ellas, el transporte privado es el servicio personalizado e independiente del desplazamiento de las personas, el cual no está abierto para un público en general, donde podemos encontrar vehículos de gran cantidad de capacidad vial por pasajero que viaja (coche particular, taxi, mototaxi, etc).

Una encuesta realizada por MarketWIN (2018) titulada “Estudio de Tráfico y Tendencias de Movilidad Urbana en Lima”, reportó que una gran cantidad de limeños realizan movimientos de largas distancias entre las 6:00 y 8:00 a.m por las mañanas presentando un pico máximo de congestión. Los encuestados señalaron en un 51 % que pierden más de dos horas al día debido a la congestión vehicular culpando el exceso de autos en la ciudad. Esta congestión vehicular se debe a un gran flujo de personas recurriendo a servicios de transporte, siendo el bus, microbús, el Metropolitano y autos propios los más frecuentados; entre estos autos privados 4 de cada 10 conductores realiza el desplazamiento de varias personas en sus vehículos (carpooling).

Según el Instituto Nacional de Estadística e Informática (2020), en el 2019 el Perú tenía en circulación 3 millones de vehículos a nivel nacional, donde el 1 982 650 están en la capital (66 %). Ello debido a que, la capital posee alrededor de 10 millones de personas con la necesidad de desplazarse por diferentes partes de Lima.

Por tal motivo diferentes empresas de redes de transporte (ERT) nacionales e internacionales, las cuales permiten la conexión entre conductores y pasajeros bajo plataformas digitales, se han fijado en el transporte urbano del país como: Uber, Taxi Beat, DiDi, entre otros. Esta creciente demanda de medios de transporte ha provocado el aumento de tráfico en la capital.

En esta investigación tomamos en cuenta uno de los servicios privados de gran demanda internacional para las ERTs antes de la pandemia, por su eficiencia, ajuste económico y ayuda medioambiental: el servicio de viajes compartidos.

El servicio de viaje compartido (ridesharing) permite a los viajeros compartir vehículos con aquellos que tienen orígenes, destinos y horarios similares con poca antelación, lo que reduce el número de vehículos en circulación en una área al aumentar el número de pasajeros por

transporte. Este proceso permite a los pasajeros compartir el costo del conductor.

En ciudades como Beijing, encontramos un gran desafío por la densidad demográfica que se presentan en las zonas más urbanizadas, para solucionarlo la compañía DiDi Chuxing, evaluó la incorporación de lugares de recojo y despacho más explícitos y fáciles que las descripciones de los pasajeros de su posición, con el objetivo de un ahorro de energía. Según el noticiero chino, SohuNews (2017), el servicio titulado “Station Carpool” logró reducir en un 30 % el kilometraje consumido por los automóviles. Método que tomó Uber para la creación de su servicio “Express Pool” (Uber, 2018) y Lyft con “Shader Saver” (Lyft, 2019).

En este contexto se describe la pregunta que motiva la presente investigación: ¿Cómo podemos disminuir el kilometraje de los vehículos que realizan transporte privado y beneficiar a los pasajeros con un algoritmo de inteligencia artificial?. Para este fin, se propone desarrollar un modelo de aprendizaje por refuerzo profundo que permita reconocer en una lista de pedidos pendientes, quienes pueden ser enrutados en base a diferentes parámetros, particularmente, tiempo de espera del cliente, tiempo estimado del viaje, congestión vehicular, etc.

Para lograr dicho propósito, se busca utilizar diferentes tecnologías que describan la solución al problema: Inicialmente, la construcción de un grafo dirigido para representar de forma zonal partes de un mapa donde se ubicará información como: punto de recojo y punto de despacho. De esta forma tener prefijado los puntos de recorrido del modelo

Segundo, desarrollar un modelo de aprendizaje por refuerzo profundo que interactúe con el grafo dirigido, seleccionando un nodo en el grafo que simboliza la decisión de enrutamiento del vehículo.

Finalmente, el diseño de métricas como el kilometraje ahorrado por el enrutamiento compartido y la eficiencia del modelo en disminución de coste.

Para resumir, el servicio de viajes compartidos puede llegar a ser una respuesta al problema del tráfico vehicular, facilitando el desplazamiento de personas en un mismo vehículo disminuyendo el consumo de combustibles y el tiempo de espera, por ello, el modelo busca sentar las bases a soluciones futuras ya sea para seguir incentivando el aprendizaje por refuerzo aplicado al transporte o para servir como base a futuras investigaciones.

1.2. Objetivos

1.2.1. Objetivo general

Analizar y comprender las técnicas de aprendizaje por refuerzo (RL) y los parámetros aplicados durante el ruteo del servicio de viajes compartidos (ridesharing) bajo demanda.

1.2.2. Objetivos específicos

1. Formular los parámetros del entorno estudiado que afectan la arquitectura propuesta.
2. Identificar los beneficios que presenta un servicio ridesharing para la optimización de entrega de pedidos.
3. Cotejar con otros modelos de enrutamiento de vehículos.
4. Plantear un prototipo del modelo básico no escalable.

1.2.3. Resultados

En el cuadro 1 se presentan los resultados esperados propuestos para cada objetivo:

Objetivos	Resultados	
OE1	R1	Modelo matemático que relacione todos los factores del transporte urbano analizados en el proyecto.
OE2	R2	Un gráfico comparativo de kilómetros consumidos por el servicio de ride sharing y ride hailing basado en conjunto de datos estudiado.
OE3	R3	Un grafico de eficiencia de modelos relacionados al enrutamiento.
	R4	Evaluación de parámetros de otros modelos de enrutamientos de viajes compartidos.
OE4	R5	Una arquitectura y prototipo que permite obtener que pedidos se han enrutado en un solo viaje.

Cuadro 1.1: Resultados esperados por objetivo

1.3. Alcance y limitaciones

1.3.1. Hipótesis

Mediante el uso de grafos es posible representar pedidos de viajes compartidos y carreteras que permitan realizar un modelo de RL para determinar que pedidos podemos enrutar en una lista de pedidos pendientes haciendo posible que ambos compartan el mismo vehículo disminuyendo el costo del viaje y el kilometraje.

1.3.2. Alcance

El proyecto desarrolla un diseño de un modelo de aprendizaje por refuerzo profundo. Se centra el procesamiento de datos de una ciudad y como esta maneja sus pedidos de taxis. Es un proyecto aplicado al área de transporte urbano.

Marco teórico

2.1. Viajes compartidos

Seattle Smart Traveler define al viaje como la acción de trasladarse de un punto geográfico a otro (Dailey y col., 1999). El viaje compartido es un viaje conjunto de al menos 2 participantes que comparten un mismo vehículo. Un viaje compartido exitoso requiere coordinación con respecto al tiempo y las especificaciones de recojo y despacho de un pasajero. Estas pueden necesitar de otros factores como el costo de viaje, reputación del conductor, tipo de vehículo, entre otros.

Un *viaje compartido no organizado* suele involucrar factores más cotidianos que involucran familiares y amigos, donde no se tiene claro el punto de destino de cada pasajero sino únicamente durante la ruta. Este tipo de viajes no son posibles de escalar, debido a la deficiencia de información inicial, que implicara fallas en la estimación debido a la comunicación limitada.

Un *viaje compartido organizado* es operado bajo alguna agencia remota sin tener en cuenta alguna relación entre los pasajeros.

Debido a esto, el viaje compartido ofrece un potencial como servicio escalable. El acuerdo previo por parte de el proveedor del servicio y el pasajero es una característica clave en los viajes compartidos ofreciendo mayor confianza y credibilidad al servicio, a diferencia de un taxi común o transporte público de la calle. El acuerdo de viaje pueden comenzar cuando las solicitudes u ofertas de viajes se envían a través de Internet o por teléfono a los proveedores de servicios, que luego tienen como objetivo igualar la oferta y la demanda de viajes.

Existen 2 tipos de proveedores:

- **Servicio operador:** Operan bajo su propia flota de vehículos y conductores.
- **Agencia de emparejamiento:** Facilitan el servicio mediante emparejamiento con conductores independientes, usando el método de crowdsourcing.

Una característica notable es que la mayoría de las decisiones las toman los operadores de servicios, mientras que los participantes simplemente deciden si o no participar. Por el contrario, las agencias de emparejamiento se centran en los servicios de emparejamiento de viajes entre

conductores de automóviles y pasajeros individuales. A diferencia de los operadores de servicios, las agencias de emparejamiento no proporcionan vehículos ni conductores. En cambio, los conductores individuales tienen sus propios planes de viaje.

2.1.1. Precios

Dado que la tarifa del viaje determina no solo el precio que el pasajero tiene que pagar por el viaje, sino también el ingreso para el conductor, las decisiones de fijación de precios influyen en las distribuciones de la demanda y la oferta a través del precio, por ejemplo, el uso de precios de aumento durante las horas pico.

Normalmente, los participantes en el sistema de transporte compartido están motivados por el hecho de compartir los costes del viaje, viajar rápido mediante el uso de vehículos grandes y mitigar los problemas medioambientales, en lugar de obtener beneficios. Además, dado que la forma final del transporte compartido sólo se determina cuando se determina el último pasajero de un vehículo, los participantes no pueden evaluar el valor del transporte compartido en el momento de presentar el pedido, a diferencia de las bolsas de valores y de anuncios. Por tal motivo, las agencias de emparejamiento, optan por ampliar la fijación de precios para evitar la falta de trazabilidad del cálculo de estos.

La fijación de precios especifica la cantidad de dinero que se transfiere entre las partes implicadas, incluyendo la forma de compartir los costes de la gasolina, el peaje y el aparcamiento, y la forma de cobrar las tasas de transacción por parte de las agencias de correspondencia. En la industria se utilizan los siguientes tres tipos de reglas de fijación de precios:

- **Precio de catálogo:** Los conductores o pasajeros especifican sus precios límite al hacer el listado. Por lo tanto, los propietarios del listado determinan su precio.
- **Precio basado en reglas:** El precio se determina mediante una fórmula de cálculo de costes especificada por la agencia. Normalmente, la fórmula es una tarifa estándar predeterminada por distancia multiplicada por una distancia calculada entre los lugares de recogida y entrega.
- **Precios basados en la negociación:** La agencia no participa en la fijación de precios. Se negocia entre los socios potenciales mientras determinan los lugares de recogida y entrega.

2.1.2. Patrones y variables

En la literatura, (Morency, 2007) clasificó al carpooling mediante elementos posicionales. Entre la presentación de estos elementos tenemos los más importantes:

- **Real distance** dR_b : Nos referimos a la distancia original del viaje tomado si hubiera un solo pasajero, para el pasajero b .
- **Relative distance** d_b : Nos referimos a la distancia relativa del viaje tomado por el pasajero b no necesariamente solo.
- **Rideshare route** $R(a, B)$: Ruta de viaje compartido. Donde a es el conductor de la ruta y B es el conjunto de pasajeros del viaje.
- **Pick-up location** pk_b : Punto de recojo de un pasajero b .
- **Drop-off location** dp_b : Punto de despacho de un pasajero b .

A continuación, presentamos algunos patrones de viaje más comunes en los viajes compartidos.

- **Patrón 1 (viaje idéntico)**: En este es el flujo original del problema, donde el punto de destino del conductor y el punto de despacho del pasajero son los mismos, esto puede ser aplicado a más de un pasajero.

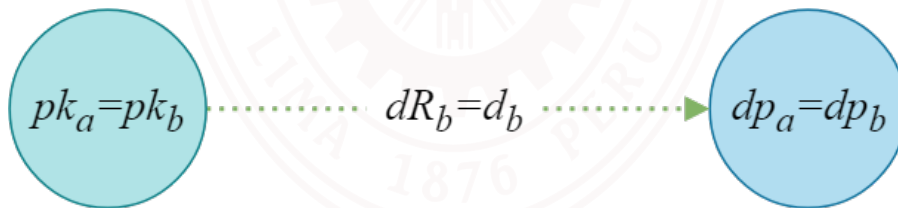


Figura 2.1: Ilustración del patrón 1, viaje idéntico.

- **Patrón 2 (viaje incluido)**: Para este, el origen y destino de un pasajero b esta en el camino del conductor a , quien tiene otra ruta con otro pasajero c , tenemos en cuenta además que la ruta del viaje de cada pasajero es distinta (es decir $R(a, c) \neq R(a, b)$); para este caso la distancia real del viaje cambia para ambas rutas (dR_b, dR_c), puesto que para el pasajero b sus puntos de recojo y destino están incluidos en la ruta del conductor a ($pk_b, dp_b \in R(a, c)$), por lo que podemos decir que el recorrido real del pasajero c será la distancia recorrida por el conductor ($dR_c = d_a$).

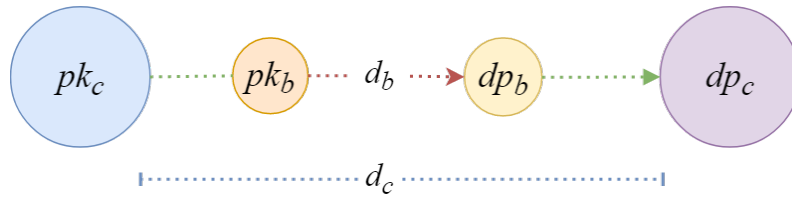


Figura 2.2: Ilustración del patrón 2, viaje incluido.

- **Patrón 3 (viaje parcial):** Dada la ruta de un conductor a con el pasajero c , $R(a, c)$. Se presenta el pasajero b para el cual el punto recojo pk_b o el de entrega dp_b no están en la ruta, es decir, $pk_b \in R(a, c) \wedge dp_b \in R(a, c)$; por lo tanto existe dos puntos $u_b, v_b \in R(a, c)$ y $(u_b = pk_b) \vee (v_b = dp_b)$, esto implica que no podemos asegurar que la distancia del pasajero c sea igual a la distancia recorrida por el conductor, debido a la existencia del nuevo pasajero b . El viaje es parcial porque el pasajero sigue la ruta una parte del viaje.

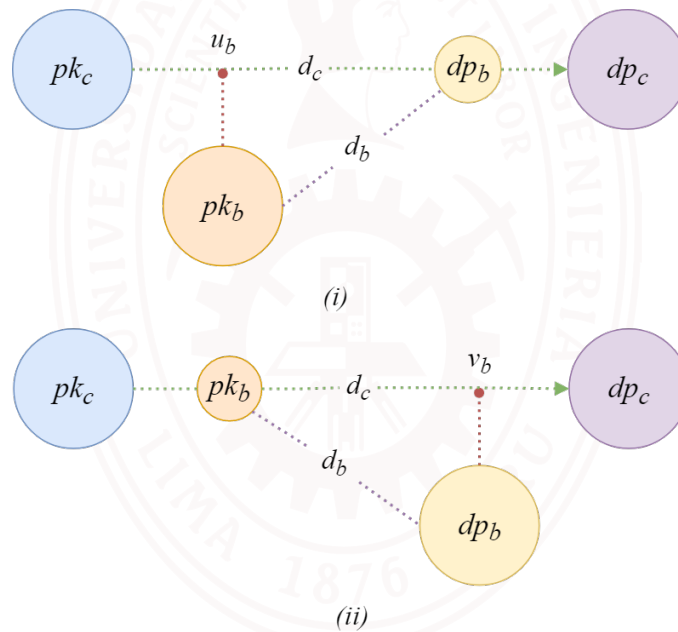


Figura 2.3: Ilustración del patrón 3, donde, (i). $u_b = pk_b$, (ii). $v_b = dp_b$.

- **Patrón 4 (viaje por desvío):** Finalmente el último y más común patrón de viaje, este sucede al momento de tomar al pasajero b , siendo que ninguno de sus puntos de recojo y entrega se encuentran en la ruta del conductor ($pk_b, dp_b \in R(a, c)$), siendo que este tendrá que realizar un desvío en su ruta original para recoger y dejar al pasajero b . Esto generará distancias directas entre los puntos de inicio y partida de los pasajeros, y la distancia relativa será mucho más extensa.

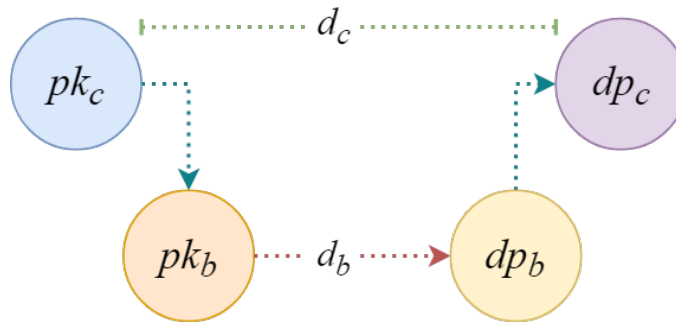


Figura 2.4: Ilustración del patrón 4, viaje por desvío.

2.1.3. Beneficios y riesgos

Segun Shaheen y col., 2018, el carpooling ofrece beneficios a largo plazo como la reducción del consumo del combustible, comprobando el aporte en reducir gases de efecto invernadero por los automóviles. Se comprobó que los vehículos que disponen viajes compartidos reducen el desplazamiento de autos en un 4% disminuyendo el tráfico vehicular, además estos servicios abren las puertas a viajes más económicos y rápidos.

Entre sus deficientes esta la falta de privacidad y problemas en la salud pública por la alta probabilidad de contraer enfermedades virales; la revista Forbes este año realizó un análisis de la baja en la rentabilidad de los viajes compartidos por su falta de seguridad en la salud (Cristóbal, 2021), en este se especifica la pérdida de pasajeros en las agencias más populares (Uber y Lyft) de un 70%, siendo la salud uno de su principal defecto en la actualidad.

2.2. Aprendizaje por refuerzo

Aprendizaje por refuerzo, en inglés (reinforcement learning) pertenece a una de las categorías del machine learning, el enfoque de RL es capacitar a un agente de manera que este pueda optimizar su comportamiento acumulado y aprendiendo de sus experiencias, interactuando con un entorno. La optimalidad del modelo depende de la maximización de la recompensa total acumulada por el agente.

RL se basa en el marco del proceso de decisión de Markov (MDP), donde nuestro agente tiene un estado s de espacio de estados \mathcal{S} y puede realizar acciones A definida por el entorno. Una vez ejecutada la acción, el agente recibirá una recompensa inmediata $R(s, a)$ del entorno y su estado cambia de acuerdo a los parametros de transición. El proceso se repetirá hasta obtener un estado terminal s_f , segun la figura 2.5 dado el control de los tripletes $(s_t, a_t, r_t)_t$. Donde t es el indice de la iteración.

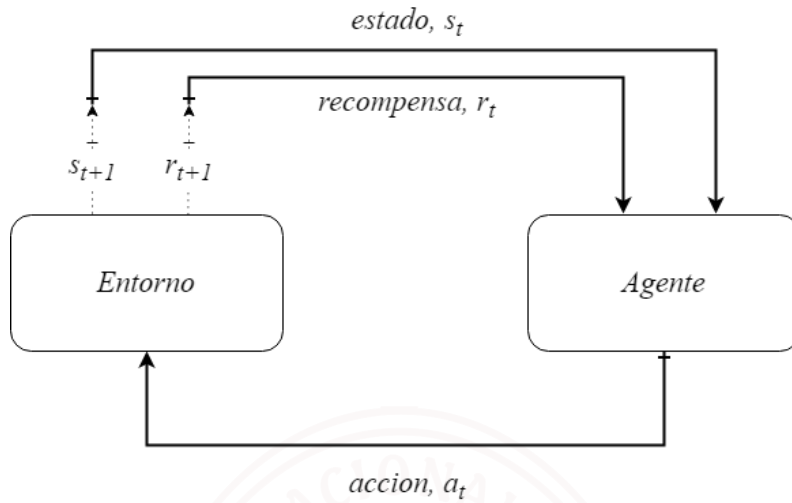


Figura 2.5: Ilustración de la interacción entorno-agente.

El objetivo del MDP es maximizar la recompensa acumulada. Una variable elemental es la función de valor que representa la esperanza de la recompensa acumulada respecto al estado s

$$V(s) = \mathbf{E}[\sum_k^n \gamma_k r_t | s_k] \quad (2.1)$$

Esta función de valor satisface la ecuación de Bellman,

$$V(s_t) = \sum_{a_t} P(s_{t+1}, r_t | s_t, a_t) (r_t(s_t, a_t) + \gamma V(s_{t+1}))$$

Para la función Q valor, la función actúa bajo la acción y valor

$$Q(s, a) = \mathbf{E}[\sum_k^n \gamma_k r_t | s_k, a_k] \quad (2.2)$$

donde la política es una función $\pi(s) : \mathcal{S} \rightarrow \mathcal{A}$.

Podemos representar a la ecuación 2.1 usando la forma de la ecuación de Bellman (2.2) en su forma iterativa.

$$V(s) \leftarrow \max_a \sum_{s', r} P(s', r | s, a) (r + \gamma V(s')) \quad (2.3)$$

A partir de esto podemos armar un método sin modelos que aprenda bajo una función de valor y se optimiza la política sin alterar el entorno. El ejemplo más común es el temporal-difference (TD) learning, que realiza actualizaciones iterativas de V .

$$V(s) \leftarrow V(s) + \alpha (r + \gamma V(s') - V(s)) \quad (2.4)$$

donde α es la tasa de aprendizaje. Ahora si este valor de aprendizaje dependiera de las acciones posibles por un estado, dando mayor control al objetivo de la recompensa, da como resultado a

Q-learning, que se basa en actualizar la función Q-valor .

$$Q(s, a) \leftarrow Q(s, a) + \alpha(r + \gamma \max_{a'} Q(s', a') - Q(s, a)) \quad (2.5)$$

En la búsqueda de poder realizar operaciones más complejas usando el formato de entrenamiento de las redes neuronales, se planeó el deep Q-network (DQN), Mnih y col., 2015.

Este aproxima $Q(s, a)$ por una red neuronal a lo largo de algunas heurísticas que ofrecen estabilidad al entrenamiento.

Un método basado en políticas aprende directamente calculando el gradiente de política (PG), el gradiente de la recompensa acumulada con respecto a los parámetros de política θ ,

$$\nabla_{\theta} \mathcal{J}(\theta) = \sum_s \mu(s) \sum_a Q^{\pi}(s, a) \nabla_{\theta} \pi(a|s, \theta) \quad (2.6)$$

donde μ es la distribución de la política en π .

Estado de arte

En este capítulo se realizó una revisión de las investigaciones previas donde se vinieron utilizando técnicas de aprendizaje por refuerzo en el transporte. Se realizó la búsqueda de los estudios previos mediante cadenas de búsqueda, para encontrar información útil para posibles estrategias de solución.

3.1. Objetivo de la revisión

El objetivo de esta búsqueda es principalmente para encontrar estrategias del planteamiento del problema y herramientas para la implementación de la problemática planteada para la investigación.

3.2. Método usado en la revisión

Se planteó el uso de preguntas iniciales que impliquen los objetivos y además cadenas de búsqueda en las bases de datos Scopus, IEEEExplore y ACM. Al tratarse de un tema de aplicación novedoso y que aun posee continuidad se buscó las revisiones más recientes posibles. Además, tomando en cuenta el impacto de la sobre nuestro enfoque de aplicación. En el Tabla 3.1 se enlistan las bases de datos revisadas y la consulta realizada en cada una.

3.3. Estado del Arte

Realizando el análisis exhaustivo de todas las revisiones con respecto a palabras específicas se obtuvieron 28 fuentes. Donde se tomaron en cuenta las 5 más relevantes.

En Braverman y col. (2019) , se toma encuentra un modelo de una red de colas basada en estados, el cual toma en cuenta carros completamente vacíos y los enruta con pedidos. Logran

Base de Datos	Consulta	Resultados
Scopus	TITLE-ABS-KEY (reinforcement learning AND routing AND (rideshading OR ride shading) OR graph AND (LIMIT-TO (PUBYEAR , 2021) OR LIMIT-TO (PUBYEAR , 2017))	Article (349) Magazines (234) Books (14)
ACM	"query": {AllField:(reinforcement learning) AND AllField:(routing AND transportation AND traffic) AND AllField:(ride shading OR rideshading) AND AllField:(NOT (robot AND multi agent)) AND AllField:(routing OR transportation OR traffic)} "filter": {ACM Content: DL}	Journals (34) Article (16)
IEEEExplore	(All Metadata":reinforcement learning) OR (All Metadata":ridepooling OR All Metadata":ridesharing) AND (All Metadata":transportation OR All Metadata":taxi) AND (All Metadata":routing) NOT (All Metadata":robot AND All Metadata":multi)	Journals (310)

Cuadro 3.1: Cuadro de consultas en Bases de Datos Bibliográficas.

la optimización del flujo de automóviles mediante una optimización basada en límites, los resultados predicen el flujo de carros vacíos en horarios del día, llegando a un punto de equilibrio.

Para Haliem y col. (2021), se desarrolla un servicio que se centra en generar rutas óptimas para cada vehículo, además el algoritmo planteado está basando en la recompensa donde el flujo esta influenciado y calculado por una Deep Q-network. Se considera importante la demanda en línea, los precios asociados con cada viaje, las capacidades y ubicaciones del vehículo, para realizar sus decisiones.

Este modelo fue entrenado por el conjunto de datos públicos de New York City Taxi.

Añadido a esto, en Rahili y col. (2018) el objetivo será resolver el problema en tiempo real, donde su arquitectura de basa en un proceso de decisión de Markov (MDP), planteando un algoritmo SARSA distribuido para capturar los errores del modelo y el cambio frecuente del entorno como la posición de clientes, tráfico y tarifas. Ellos aplican este modelo a todos sus agentes, donde el algoritmo SARSA realiza de tarea de asignación para designar a cada carro un cliente libre.

Para el enfoque de optimización de rutas podemos encontrar que Khan y Lapkin (2020), plantea una arquitectura de diseño que permite al agente de aprendizaje por refuerzo utilizar

el ensayo y error para limitar su búsqueda a las rutas más prometedoras, en lugar de enumerar exhaustivamente las soluciones. En cada iteración, el agente emplea datos recopilados previamente para guiar la búsqueda de nuevas trayectorias.

Finalmente, tenemos a Almasan y col. (2020) donde se muestra un análisis significativo del problema de toma de decisiones mediante DRL el cual busca solucionar los problemas de optimización de enrutamiento, dado que una red neuronal presenta dificultades para predecir un entorno que dependa de una estructura de grafos, dado que tendrá problemas para solucionar redes neuronales completamente conectadas. En este documento se analiza la posibilidad de utilizar Graph Neural Networks (GNN) en conjunto con DRL. Donde GNN tiene la tarea de modelar los grafos propuestos y en conjunto tiene la capacidad de operar bajo una topología. El proyecto se analizó en un escenario de transporte óptico (OTN).



Metodología

En este capítulo, se explican las metodologías adoptadas para generar el análisis de datos y arquitectura del modelo.

4.1. Base de datos

Para poder realizar nuestro modelo debemos tener datos que puedan explicar perfectamente la información necesaria en el espacio. Para ellos buscamos utilizar datos que ofrezcan información en el formato estándar de representación geográfica, el formato GeoJson y el estándar de GIS, que ofrece información geográfica y cartográfica (Hugh-jones, 1997).

4.1.1. Conjunto de datos Chicago Taxi Trips

4.1.1.1. Resumen

El conjunto de datos de estudio (Chicago Portal, 2021) es un reporte de viajes en taxi en la ciudad de Chicago, bajo un proceso de regularización y conteo de taxis en la ciudad, todas las empresas afiliadas a estos servicios registran sus taxis mediante un identificador en la public data city of Chicado para proteger la privacidad.

El conjunto de datos de Chicado Taxi Trips fue lanzado desde enero del 2013 hasta la fecha, sin embargo según el reporte de actualizaciones del conjunto de datos de Chicago, en octubre del 2017 desactivó la subida de nuevos datos por problemas en la recolección de datos, hasta abril del 2019 que reanudó el reporte de datos (Chicago Data Portal, 2019).

4.1.1.2. Justificación

Sé tomaron los datos de Chicado Taxi por su uso de datos espaciales SQL, los cuales representan información sobre la ubicación física y la forma de los objetos geométricos (Microsoft, 2020). Estos objetos pueden ser ubicaciones puntuales u objetos más complejos como carreteras, etc.

4.1.1.3. Estructura del conjunto de datos

Columna	Descripción	Tipo de dato
Trip ID	Un identificador único para el viaje.	Texto
Taxi ID	Un identificador único para el taxi.	Texto
Trip Start Timestamp	Cuando comenzó el viaje.	Timestamp
Trip End Timestamp	Cuando terminó el viaje.	Timestamp
Trip Seconds	Tiempo del viaje en segundos.	Número
Trip Miles	Distancia del viaje en millas.	Número
Pickup Census Tract	La sección censal donde comenzó el viaje.	Número
Dropoff Census Tract	La sección censal donde terminó el viaje.	Número
Pickup Community Area	El Área Comunitaria donde comenzó el viaje	Número
Dropoff Community Area	El Área Comunitaria donde terminó el viaje.	Número
Fare	La tarifa del viaje.	Número
Tips	La propina para el viaje.	Número
Tolls	Los peajes para el viaje.	Número
Extras	Cargos adicionales por el viaje.	Número
Trip Total	Cargos adicionales por el viaje.	Número
Payment Type	Tipo de pago del viaje.	Texto
Company	La compañía de taxis.	Texto
Pickup Centroid Latitude	La latitud del centro de la sección censal de recogida.	Número
Pickup Centroid Longitude	La longitud del centro de la sección censal de recogida.	Número
Pickup Centroid Location	La ubicación del centro de la sección censal de recogida.	POINT
Dropoff Centroid Latitude	La latitud del centro de la sección censal de abandono.	Número
Dropoff Centroid Longitude	La longitud del centro de la sección censal de abandono.	Número
Dropoff Centroid Location	La ubicación del centro de la sección censal de entrega.	POINT

Cuadro 4.1: Tabla de estructura del conjunto de datos Taxi Trip.

El dataset posee 23 columnas los que se describen brevemente en cuadro 4.1.

4.1.2. Boundaries Community Areas

4.1.2.1. Resumen

El conjunto de datos de estudio es una lista los límites actuales del área comunitaria de Chicago. Estos límites están representados por varias sub áreas, las cuales representan un sector del mapa de Chicago, el cual no ha cambiado en los últimos 4 años (Chicago Data Portal, 2013).

4.1.2.2. Justificación

Se plantea el uso de este dataset para reconocer en que área se encuentran los pedidos, haciendo más fácil el uso de los datos.

4.1.3. Estructura del conjunto de datos

Columna	Descripción	Tipo de dato
the_geom	Recuadro del área de la comunidad.	MULTIPOLYGON
PERIMETER	Perímetro del área de la comunidad.	Número
AREA	Área del área de la comunidad.	Número
COMAREA_	Sin Descripción.	Número
COMAREA_ID	Sin Descripción.	Número
AREA_NUMBE	Número asignado a cada área de comunidad.	Número
COMMUNITY	Nombre de cada área de comunidad.	Texto
AREA_NUM.1	Sin Descripción.	Número
SHAPE_AREA	Área perimetral del recuadro.	Número
SHAPE_LEN	Distancia perimetral del recuadro.	Número

Cuadro 4.2: Tabla de estructura del conjunto de datos Boundaries Community Areas.

4.2. Preprocesamiento

Buscamos eliminar parte de las filas y columnas que no nos sirven, para ello seleccionamos las columnas que menos relevancia nos dará para el entrenamiento.

Taxi Trips	Community Areas
Pickup Census Tract	PERIMETER
Dropoff Census Tract	AREA
Fare	COMAREA_
Tips	COMAREA_ID
Extras	AREA_NUM_1
Tolls	
Payment Type	
Company	
Dropoff Centroid Longitude	
Dropoff Centroid Latitude	
Pickup Centroid Latitude	
Pickup Centroid Longitude	

Cuadro 4.3: Tabla de parámetros eliminados del conjunto de datos Community Areas y Taxi Trips.

Teniendo esto en cuenta, ambos conjuntos de datos usan el formato de datos espaciales. Debido a esto y a la carga de más de 60 gbs de datos, optamos por realizar lo siguiente:

- En el conjunto de datos Taxi Trips eliminamos todos los datos que no tengan **Pickup Centroid Location** y **Dropoff Centroid Location**, esto debido a que son necesarios en nuestras pruebas.
- El proyecto intentó evitar el costo de recursos de memoria RAM; cargamos los datos a una base de datos espacial Postgres (PostGIS), evitando así la carga de datos en memoria que resulten costosos de tratar, como el formato csv; además, usando una imagen de PostGis dockerizada, evitando el consumo excesivo de los recursos.
- Separar los puntos de recojo (pickup points) y los puntos de entrega (dropoff points), esto con el objetivo de crear muchas más combinaciones con los datos.

4.3. Definición del modelo

El modelo a trabajar estará basado en una red profunda de Q-Learning, donde el entorno será una matriz de adyacencia de un *grafo dirigido pick-drop* (figura 4.1), para ello debemos definir la estructura de dicho grafo de entrada.

Buscamos crear una matriz de adyacencia capaz de interpretar los valores de las distancias relativas por cada punto, teniendo en cuenta las siguientes reglas:

- Los puntos de recojo (**pickup**) están *totalmente conectados*, se permite dirigirnos a algún otro punto de recojo al igual que ir a cualquier punto de despacho.
- Los puntos de despacho (**dropoff**) están *parcialmente conectados*, esto se debe a que para cada punto de despacho no les será posible dirigirse a un punto de recojo.
- Solo existe un recorrido por cada punto, por lo que apenas se pasa por un punto, los nodos que conectan a este quedan eliminados en la siguiente movimiento.

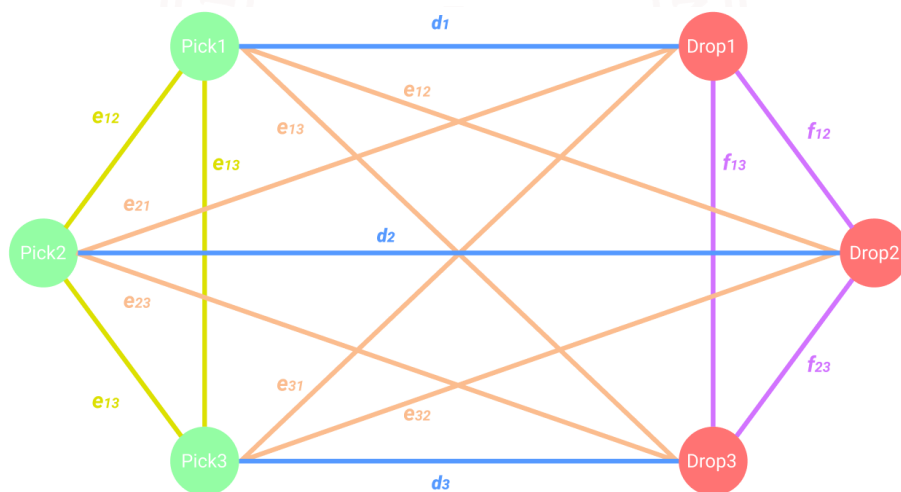


Figura 4.1: Ilustración del grafo dirigido pick-drop.

La tarea del modelo será obtener la ruta de estos 3 pasajeros que poseen distancias relativas entre sí; sabemos que la distancia real de los pedidos es d_i por lo que se buscará minimizar esta distancia máxima.

La matriz de adyacencia para el grafo pick-drop, será el siguiente:

$$\begin{pmatrix} 0 & d_1 & e_{12} & f_{12} & e_{13} & f_{13} \\ d_1 & 0 & f_{21} & l_{12} & f_{31} & l_{13} \\ e_{12} & f_{21} & 0 & d_2 & e_{23} & f_{23} \\ f_{12} & l_{12} & d_2 & 0 & f_{32} & l_{23} \\ e_{13} & f_{31} & e_{23} & f_{32} & 0 & d_3 \\ f_{13} & l_{13} & f_{23} & l_{23} & d_3 & 0 \end{pmatrix} \quad (4.1)$$

Donde:

- d_i : Representa las distancias de los puntos de recojo a sus destinos originales.
- e_{ij} : Representa las distancias entre el punto de recojo i al punto de recojo j , donde $i \neq j$.
- f_{ij} : Representa las distancias entre el punto de recojo i al punto de destino j , donde $i \neq j$.
- l_{ij} : Representa las distancias entre el punto de destino i al punto de destino j , donde $i \neq j$.

Experimentación y Resultados

5.1. Flujo conceptual

Debido a la creación de un modelo capaz de trabajar con un conjunto de datos tan grande, el proyecto debía ser capaz de soportar una gran cantidad de épocas, en el menor tiempo posible, para esto, se definió un flujo de conexiones, que se puede ver representado en la figura 5.2.

El proyecto se conecta con la base de datos y realizar las consultas necesarias para obtener

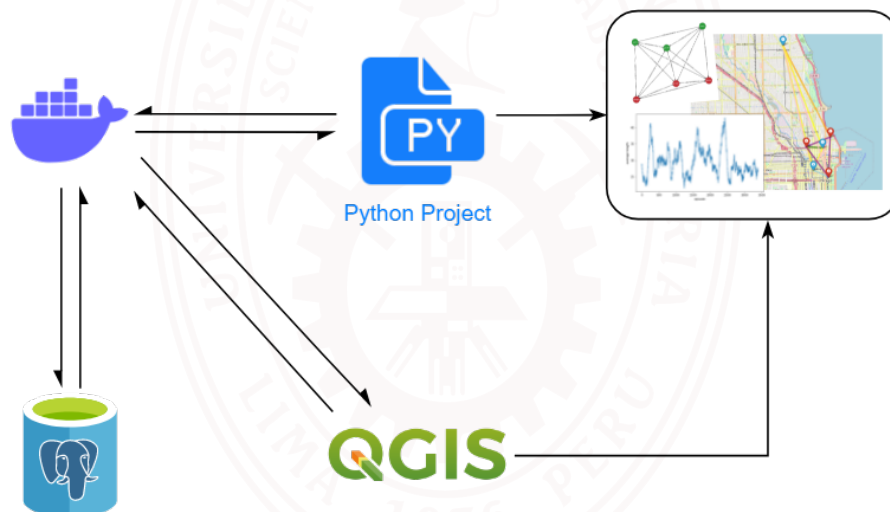


Figura 5.1: Ilustración del flujo de conexión del proyecto.

los valores en formato GIS, para luego de entrenar el modelo generar tablas de respuestas con los puntos con las conexiones creadas, que podran ser visualizadas, en QGIS una aplicación profesional de GIS de software Libre.

Esta haría más accesible para el modelo seleccionar 3 puntos de recojo y 3 puntos de entrega, centrándonos únicamente al costo computacional del modelo.

5.2. Estructura del modelo

La estructura del modelo está basada por la técnica DQN (2.2) donde aplicamos espacios de memoria para almacenar información temporal del ruteo.

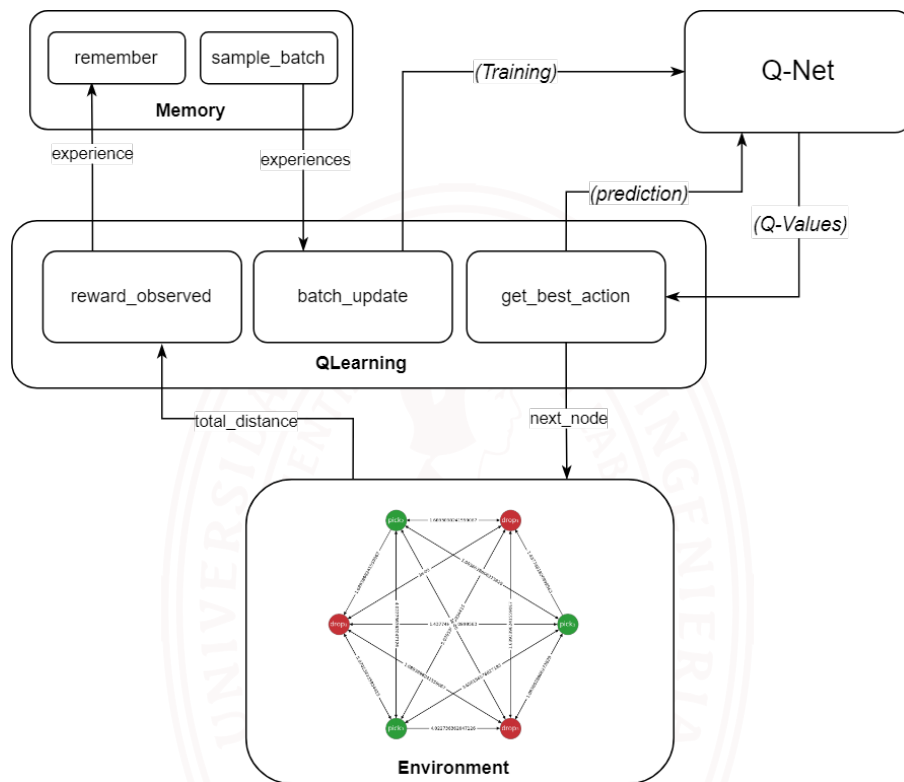


Figura 5.2: Ilustración de la estructura del modelo.

En este modelo usamos una red DQN (Q-Net) que recibe como entrada las coordenadas y la matriz de adyacencia, retornando un arreglo del posible nodo a diriginos. Mientras que el entorno nos envía la distancia total que sera útil para realizar la recompensa relativa en cada estado, donde dicho valor es el valor a minimizar.

Además, se guarda la experiencia del modelo en memoria, como una forma de guardar el aprendizaje del modelo, y generar archivos de las mejores experiencias.

5.3. Métricas

Se tomó como métrica el camino propuesto por el modelo a lo que se llamó **average length**. El modelo buscó obtener cada vez un promedio más corto en sus distancias. Además, se obtuvo la

función de coste minimizada por el lr de la ecuación 2.5 y la ecuación 2.6, el parámetro epsilon que de forma aleatoria cambiaba el destino del siguiente nodo a lo que llamamos **exploración vs explotación**.

5.4. Resultados

Se puede ver en la figura 5.3 que las distancias se lograron minimizar a partir de la epoca 15000, mientras que en la figura 5.4 se puede ver que el costo a tenido un descenso mas lento, en ambos casos el proceso a sido lento debido al tipo de aprendizaje seleccionado.

A pesar de ello, el resultado del entrenamiento con respecto a los puntos se dió efectivamente.

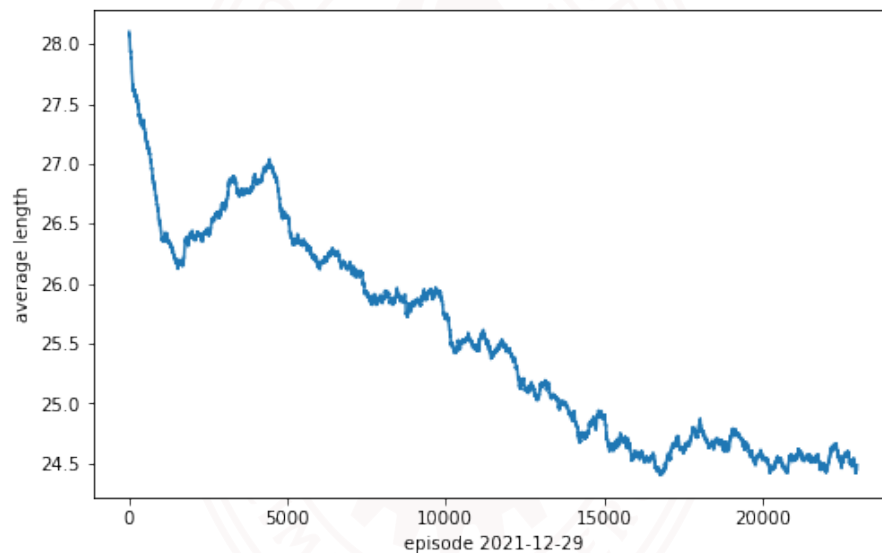


Figura 5.3: Gráfica de resultado de la distancia promedio durante las epocas

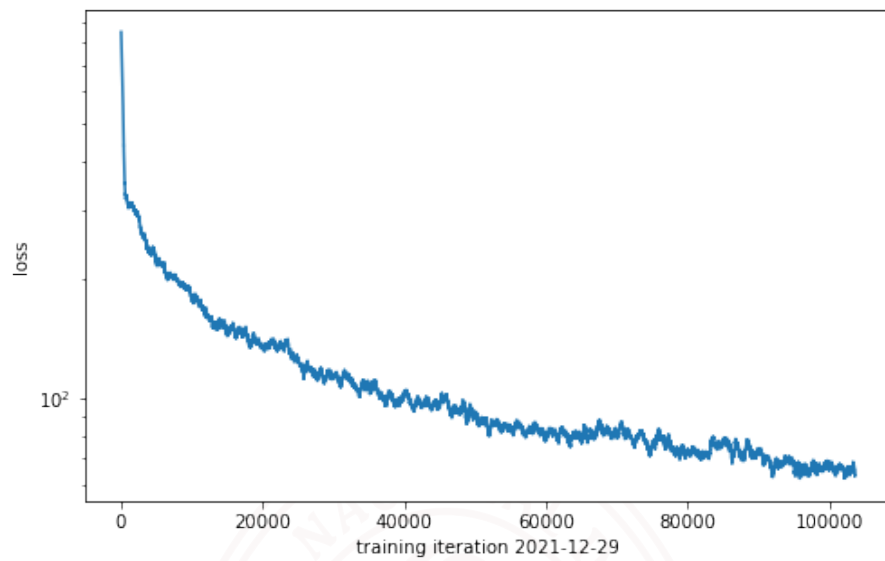


Figura 5.4: Gráfica de resultado del costo durante las épocas

También utilizando el flujo conceptual del proyecto se indican los nodos generados en formato GIS que podrán ser visualizadas en un mapa:

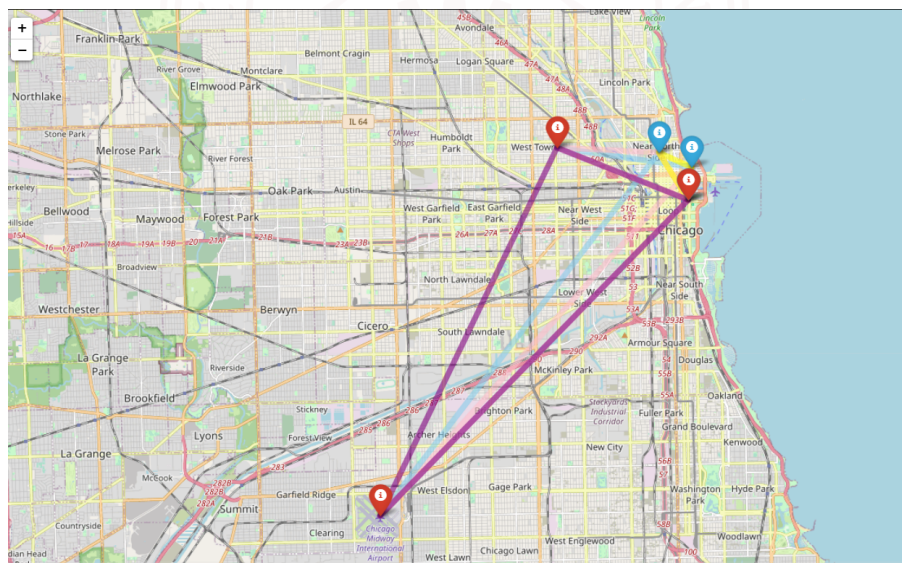


Figura 5.5: Ilustración de los nodos generados en QGIS

Usamos el modelo Q-net para retornar el arreglo de nodos en un grafo, después de la ejecución de la arquitectura propuesta:

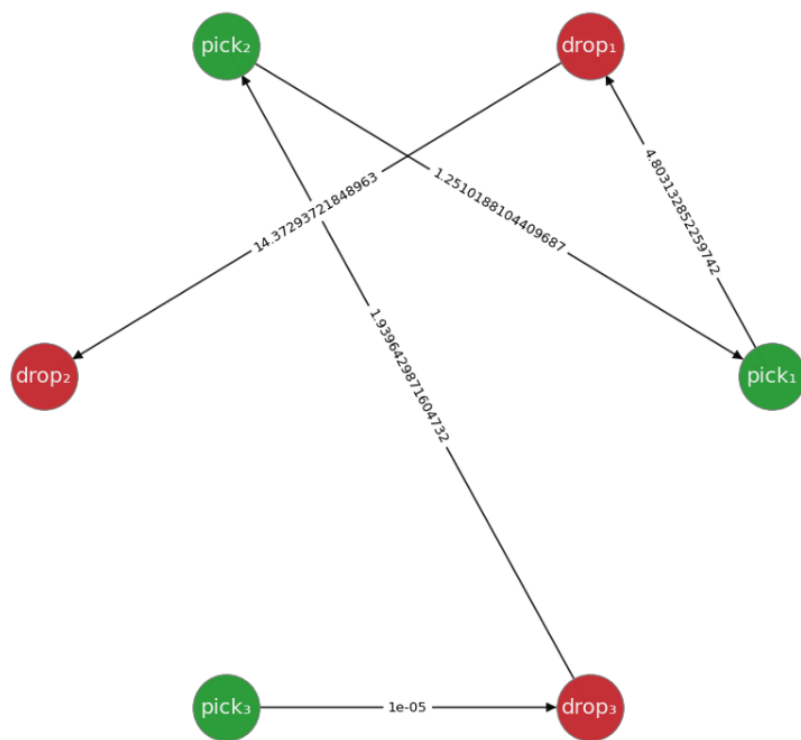


Figura 5.6: Ilustración del grafo con el camino propuesto por el modelo QNet usando el grafo pick-drop

Conclusiones

6.1. Conclusiones

- Se diseñó una arquitectura que permite obtener solución al problema de enrutamiento para viajes compartidos usando los conceptos de aprendizaje por refuerzo y los patrones de viajes compartidos.
- Se formuló el grafo dirigido necesario que entienda el agente, mostrando resultados satisfactorios.
- La arquitectura mostró una disminución en las distancias recorridas por el modelo vs el flujo tradicional de viaje idéntico .
- Se pudo identificar los parámetros más comunes para enrutar viajes y los pesos necesarios para obtener la ruta ideal de viaje compartido.

6.2. Trabajos futuros

- Se busca aumentar los parámetros de decisión en los pesos del grafo, además de incluir la posición del auto en el tiempo que es pedida la ruta.
- Implementar un modelo que pueda solucionar los 4 patrones de viajes compartidos.
- Realizar un modelo escalable que ofrezca resultados reales para la vida cotidiana.

Bibliografía

- Almasan, P., Suárez-Varela, J., Badia-Sampera, A., Rusek, K., Barlet-Ros, P. & Cabellos-Aparicio, A. (2020). Deep Reinforcement Learning meets Graph Neural Networks: exploring a routing optimization use case.
- Braverman, A., Dai, J. G., Liu, X. & Ying, L. (2019). Empty-Car Routing in Ridesharing Systems. *Oper. Res.*, 67, 1437-1452.
- Chicago Data Portal. (2013). *Boundaries - Community Areas (current)* [Online]. <https://data.cityofchicago.org/Facilities-Geographic-Boundaries/Boundaries-Community-Areas-current-/cauq-8yn6/data>
Accedido: 2021-12-05
- Chicago Data Portal. (2019). *Update of Chicago Taxi Trips DataSet* [Online]. <http://dev.cityofchicago.org/open%20data/data%20portal/2019/07/01/taxi-dataset-relaunch.html>
Accedido: 2021-12-05
- Chicago Portal. (2021). Taxi Trips [Online]. <https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew>
Accedido: 2021-12-29
- Cristóbal, E. (2021). This Is The Future Of Ridesharing: What You Need To Know About Lyft And Uber Now [Online]. <https://www.forbes.com/sites/christopherelliott/2021/04/17/this-is-the-future-of-ridesharing-what-you-need-to-know-about-lyft-and-uber-now/?sh=5b4d81686916>
Accedido: 2021-12-29
- Dailey, D., Loseff, D. & Meyers, D. (1999). Seattle smart traveler: dynamic ridematching on the World Wide Web. *Transportation Research Part C: Emerging Technologies*, 7(1), 17-32. [https://doi.org/https://doi.org/10.1016/S0968-090X\(99\)00007-8](https://doi.org/https://doi.org/10.1016/S0968-090X(99)00007-8)
- Haliem, M., Mani, G., Aggarwal, V. & Bhargava, B. (2021). A Distributed Model-Free Ride-Sharing Approach for Joint Matching, Pricing, and Dispatching Using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems*, 1-12. <https://doi.org/10.1109/TITS.2021.3096537>
- Hugh-jones, M. (1997). Beyond mapping: Concepts algorithms, and issues in GIS. *Preventive Veterinary Medicine*, 29, 345-347.

- Instituto Nacional de Estadística e Informática. (2020). *Parque automotor en circulación a nivel nacional, según departamento (2011-2019)* [Online]. https://www.inei.gob.pe/media/MenuRecursivo/indices_tematicos/cap20025_3.xlsx
Accedido: 2021-10-17
- Khan, A. & Lapkin, A. (2020). Searching for optimal process routes: A reinforcement learning approach. *Computers and Chemical Engineering*, 141, 107027. <https://doi.org/https://doi.org/10.1016/j.compchemeng.2020.107027>
- Lyft. (2019). *Introducing Shared Saver, Our Most Affordable Ride* [Online]. <https://www.lyft.com/blog/posts/introducing-shared-saver>
Accedido: 2021-10-19
- MarketWIN. (2018). *Estudio de tráfico y tendencias de movilidad urbana en Lima* [Online]. <http://willardmanrique.pe/wp-content/uploads/2021/02/Estudio-de-Trafico-y-Tendencias-de-Movilidad-Urbana-2018.pdf>
Accedido: 2021-10-15
- Microsoft. (2020). Spatial Data (SQL Server) [Online]. <https://docs.microsoft.com/en-us/sql/relational-databases/spatial/spatial-data-sql-server?view=sql-server-ver15>
Accedido: 2021-12-29
- Mnih, V., Kavukcuoglu, K. & Silver, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518, 529-533. <https://doi.org/https://doi.org/10.1038/nature14236>
- Morency, C. (2007). The ambivalence of ridesharing. *Transportation*, 34(2), 239-253. <https://doi.org/https://doi.org/10.1007/s11116-006-9101-9>
- Rahili, S., Riviere, B., Olivier, S. & Chung, S.-J. (2018). Optimal Routing for Autonomous Taxis using Distributed Reinforcement Learning. *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, 556-563. <https://doi.org/10.1109/ICDMW.2018.00087>
- Shaheen, S., Cohen, A. & Bayen, A. (2018). The Benefits of Carpooling. *UC Berkeley: Transportation Sustainability Research Center*. <https://doi.org/https://doi.org/10.7922/G2DZ06GF>
- SohuNews. (2017). *Site carpooling aterrizó en shenzhen, para abrir la era del carpooling 2.0!* [Online]. https://m.sohu.com/a/164857652_653684?_trans_=010004_pcwzy
Accedido: 2021-10-19
- Uber. (2018). *Uber economy carpooling* [Online]. <https://www.uber.com/us/zh/ride/express-pool/>
Accedido: 2021-10-19