SOAT-Greedy: Asignación Óptima de Ambulancias en Bogotá usando Aprendizaje por Refuerzo

Valentina Cepeda Vega Daniel Lasso-Jaramillo Diana Pinilla Alarcón

Mario Velásquez Semanate

1. Introducción

Para 2022, la mitad de los accidentes de tránsito en Bogotá requirieron de atención de emergencia de una ambulancia. De acuerdo con datos de la Secretaría de Movilidad hubo 25,451 accidentes de tránsito, de los cuales el 50,6 % tuvieron de gravedad algún muerto o herido (Tabla 1). En casos de accidentes de gravedad, para garantizar una respuesta oportuna, es importante reducir los tiempos de respuesta de emergencias. El estándar internacional menciona que el tiempo de respuesta de las ambulancias ante emergencias médicas debe ser de 8 minutos, sin embargo, en diversos países de Latinoamérica supera los 30 minutos (Sánchez et al., 2020). En centros urbanos y ciudades como Bogotá, la congestión de tráfico es un obstáculo para garantizar la respuesta oportuna de las ambulancias. Por su parte, según reportes realizados en medios, los cobros realizados al SOAT (Seguro Obligatorio para Accidentes de Tránsito) por parte de las ambulancias por atender heridos de accidentes de tránsitos podría estar generando una "guerra" entre operadores de ambulancias, que afecta la atención de pacientes en todo tipo de emergencias médicas (El-Espectador, 2016).

Tabla 1: Gravedad accidentes de tránsito en Bogotá (2022)

Gravedad	Frecuencia	Porcentaje
Solo daños	12,557	49.3 %
Con heridos	12,353	48.5 %
Con muertos	541	2.1 %
Total	25,451	100.0 %

En este contexto, este documento propone un modelo de distribución óptima de ambulancias para atender de manera tiempo-eficiente los incidentes de tráfico en Bogotá a lo largo de un día, y, evitando la competencia entre ambulancias por un paciente. Esto tiene como objetivo dar una guía a los proveedores de servicios de emergencias para usar recursos existentes y dar respuesta de manera más eficiente. Para ello, este problema se formulará como un bandido multi-armado (MAB) para aprender la configuración óptima. Este trabajo comienza simulando un entorno donde los incidentes de tránsito con heridos son simulados a partir la base de datos de Histórico de Siniestros de Bogotá de la Secretaría Distrital de Movilidad. Asimismo, usando teoría de grafos, obtenemos el camino más corto entre dos ubicaciones para estimar el tiempo de respuesta promedio (*shortest-path*).

Para resolver este problema se usaron tres algoritmos: ε- Greedy, Upper Confidence Bound y Gradient Bandit Algorithm. Como resultados principales se obtuvo que ε- Greedy tuvo el mejor desempeño, con un tiempo de atención de 5,8 minutos promedio por accidente. Además, se encontró que con 3 ambulancias ubicadas entre 4 puntos de despacho al suroccidente de Bogotá se puede atender una demanda de 18 accidentes al día, y cubrir un área de aproximadamente 10 kilómetros cuadrados de acuerdo con las simulaciones.

Otros documentos han intentado aproximarse a proponer una configuración óptima para servicios de emergencias. Por ejemplo, Jankovič and Jánošíková (2021) muestran, a partir de un modelo de optimización de asignación de ambulancias en Prešov, Eslovaquia; que es mejor distribuir de manera más uniforme las ambulancias en vez de tenerlas concentradas en unos puntos. Por su parte,Bains et al. (2021) desarrollan una intervención en la que generan un sistema piloto de asignación centralizada de ambulancias en San Francisco, California. Además de redistribuir las ambulancias, también las asignan a distintos hospitales para evitar el sobre flujo de pacientes en un mismo hospital. Encuentran resultados favorables al sistema, y que una mayor distribución de las ambulancias es más eficiente. Por su parte, Bains et al. (2021) desarrollan un modelo de asignación y optimización de ambulancias conociendo, parcialmente, las demandas por el servicio. El modelo toma como inputs el promedio y desviación estándar de servicios solicitados por zonas para determinar cuántas ambulancias tener en cada base - hospital.

Nuestra aproximación se asemeja a la adoptada por Allen et al. (2021), quienes utilizaron *Deep Reinforcement Learning* (Deep RL) para ofrecer una aproximación innovadora al problema de las ambulancias. En este sentido, la principal contribución de este documento es aplicar un marco de análisis similar al empleado por Allen et al. (2021) para el problema de asignación de ambulancias, pero adaptado a una zona de Bogotá y aplicado al caso de accidentes de tránsito. De igual manera, otra contribución de este documento es ofrecer una alternativa para

los proveedores de servicios de emergencias y de seguros en caso de accidentes de tránsito en Bogotá, para optimizar la distribución de estos servicios a lo largo de la ciudad y mejorar la atención médica a pacientes ante estas situaciones.

La organización de este documento se desarrolla de la siguiente manera: la sección dos describe la zona de estudio y las fuentes de los datos. En la sección tres, se presenta el problema y el ambiente creado para la simulación de los datos. La sección cuatro describe la metodología y los algoritmos utilizados. Finalmente, las secciones 5 y 6 presentan los resultados encontrados y las conclusiones.

2. Datos

Para construir el entorno se usaron datos reales de una zona específica de Bogotá. A partir de datos históricos se simuló la ubicación de los accidentes y el tiempo entre accidentes, y se determinó la ubicación de los hospitales. La zona geográfica escogida cubre principalmente la localidad de Kennedy, Puente Aranda, Teusaquillo, Los Mártires y Santa Fé. Se seleccionó esta región porque es una de las áreas con mayor número de accidentes. Para representar el mapa de esta zona como un grafo no dirigido, se usó el paquete de OSMNX (Boeing, 2017). En este grafo, los enlaces representan las calles y los nodos corresponden a las intersecciones de la ciudad. Igualmente, utilizando información de OpenStreet Maps se crean pesos en los enlaces correspondientes al tiempo de viaje, obtenido a partir del cálculo entre los límites de velocidades según el tipo de calle y las distancias reales de las mismas. Finalmente, usando la función *nearest_node* se asignaron los hospitales, puntos de despacho de las ambulancias y los accidentes al nodo más cercano dentro del grafo, lo que permite la computación de las rutas dentro de la red vial.

Del mismo modo, para simular la distribución espacial y temporal de los accidentes, se usaron datos históricos georreferenciados del Sistema Integrado de Información sobre Movilidad Urbana (SIMUR), de acceso público. Solo se consideraron incidentes con heridos o muertes (graves), y que ocurrieron los viernes de todos los meses del 2022. Se escogió ese día específico para controlar por la variedad de patrones entre diferentes días de la semana. En la Figura 1, se visualizan los accidentes graves ocurridos en los últimos viernes de enero, junio y diciembre en el grafo de la zona seleccionada.

La Figura 2 muestra el número de accidentes de cada localidad. Las localidades con mayor incidencia de accidentes fueron Kennedy y Puente Aranda, las cuales en el 2022 tuvieron 250 y 150 accidentes con muertos y/o heridos, lo que justifica la elección de la zona para el análisis.

Para estimar la distribución espacial y, posteriormente, simular nuevos accidentes, se usó el método de Kernel Density Estimation (KDE). El parámetro de ancho de banda se determinó usando 5-fold cross-validation. En

Figura 1: Accidentes último viernes de cada mes del 2022

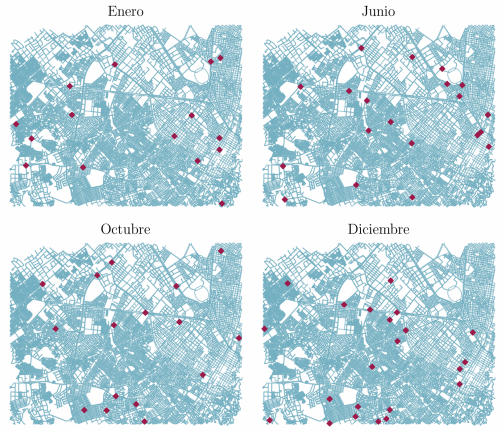
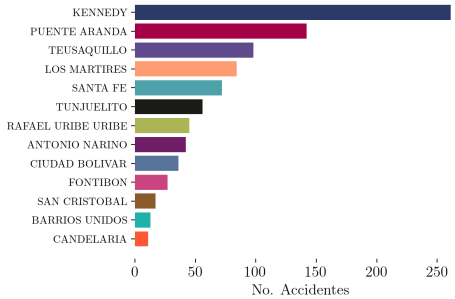
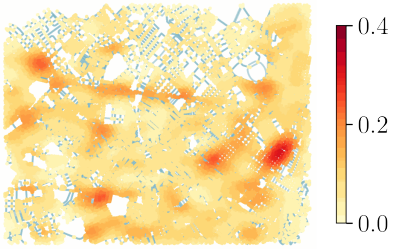


Figura 2: Número de accidentes por localidad en el 2023



la Figura 3 se presenta un mapa de calor que ilustra la función de densidad de probabilidad estimada de accidentes en cada nodo del grafo. Por otro lado, se usó una distribución negativa exponencial para simular el tiempo entre accidentes. La tasa de frecuencia de accidentes en 1 día se obtuvo al dividir las 24 horas entre el número de accidentes promedio por día según los datos (resultando en 18 accidentes).

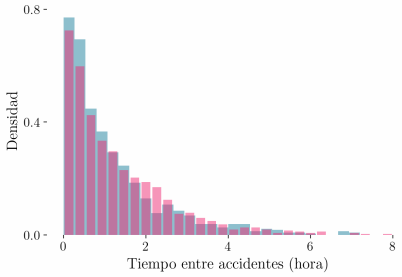
Figura 3: KDE Hotspots: Densidad de accidentes



En la Figura 4 se compara la densidad del tiempo entre accidentes de los datos reales (barras azules) con los datos simulados mediante la distribución negativa exponencial (barras rosadas).

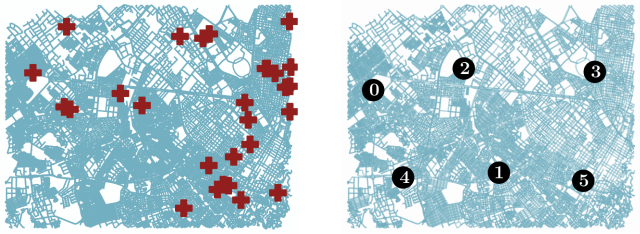
Por otro lado, la ubicación de los hospitales se obtuvo a partir de los datos georreferenciados de las Instituciones Prestadoras de Salud en Bogotá, disponibles en Datos Abiertos Bogotá. La Figura 5 (lado izquierdo) muestra los hospitales, resaltando una mayor concentración de hospitales en el oeste. Finalmente, para determinar las coordenadas de los puntos de despacho, se buscó que estuvieran uniformemente ubicados. Para esto, se empleó

Figura 4: Histograma del tiempo entre accidentes. Barras azules corresponden a la densidad de datos reales y barras rosadas a la densidad de datos simulados según dist. exponencial.



el algoritmo de K-means Clustering sobre los nodos del grafo para encontrar centroides que representaran 6 puntos de despacho. En la Figura 5 (lado derecho) se muestran los puntos de despacho obtenidos. Nótese que este método logra ubicar los puntos de despacho de manera uniforme en la región establecida.

Figura 5: Hospitales (izq.) y puntos de despacho (der.) disponibles en la zona geográfica.



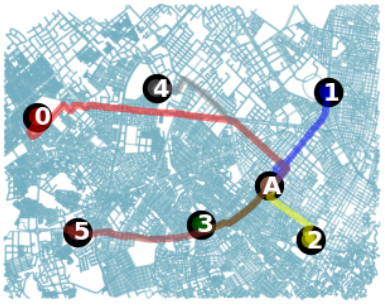
3. Problema

Se considera un entorno compuesto por un grafo de la zona geográfica de Bogotá delimitada por las coordenadas $4{,}659802°N$, $4{,}576742°S$, $-74{,}171380°W$, $-74{,}064810°E$. Para facilidad del lector, esto corresponde a una zona cuadrada entre la calle 19 con carrera 10 (NE), hasta Ciudad de Cali en la calle 43 sur con carrera 99c (SW). Cada episodio representa solo un día de atención ambulatoria para atender accidentes de tráfico en esta área. El agente es un proveedor de servicios que decide cómo distribuir las ambulancias $i \in B = \{1, 2, \ldots, b\}$ a los puntos de despacho $d \in M = \{0, 1, \ldots, m-1\}$. El espacio de acciones es discreto y está definido como todas las configuraciones posibles $\mathcal{A} = \{(d_i)_{i\in B} | d_i \in M, d_1 \leq \ldots \leq d_b\}$, donde $|\mathcal{A}| = \binom{b+m-1}{m}$[1]. Por ejemplo, con 3 ambulancias y 6 puntos de despacho, hay 56 acciones posibles. Una de ellas es la configuración $(0, 5, 5) \in \mathcal{A}$ que indica que una ambulancia es asignada al punto de despacho 0 y las otras dos ambulancias al punto de despacho 5. Además, nótese que las configuraciones $(5, 0, 5)$ y $(5, 5, 0)$ están excluidas del espacio de acciones para evitar considerar configuraciones repetidas, dado que las ambulancias son homogéneas. En cada episodio se simulan k accidentes $C = \{1, 2, \ldots, k\}$, que se muestrean espacialmente de acuerdo al modelo de KDE, y, temporalmente, de acuerdo a la distribución negativa exponencial. Además, en el grafo hay hospitales $h \in H$ que pueden atender a los pacientes de los accidentes.

La dinámica del entorno en cada episodio es la siguiente: El agente inicia realizando una acción $a \in \mathcal{A}$. Cada accidente $c \in C$ ocurre en el tiempo s_c y la ambulancia más rápida en llegar al accidente se despacha para atenderlo, la cual se denota por $i^* = \arg\min_{i\in B} l_{ic}$, donde l_{ic} corresponde al tiempo de llegada de la ambulancia $i \in B$ al accidente $c \in C$. El trayecto de la ambulancia comienza desde su punto de despacho asignado d, viaja hasta el incidente (en x_{dc} minutos), traslada a los pacientes al hospital más cercano (en x_{ch} minutos) y, finalmente, retorna a su punto de despacho original (en x_{hd} minutos). Una vez completado el recorrido, la ambulancia vuelve a estar disponible después de w minutos. El tiempo de llegada de la ambulancia que se despacha i^* al accidente c se calcula como $l_{i^*c} = \max(g_{i^*}, s_c) + x_{dc}$. El momento cuando la ambulancia vuelve a estar disponible después de atender al accidente es $g_{i^*} = l_{i^*c} + x_{ch} + x_{hd} + w$, inicializando g_{i^*} en 0 al inicio de cada episodio. El tiempo de respuesta del accidente es la diferencia entre el tiempo de llegada de la ambulancia al lugar del accidente menos el tiempo cuando ocurrió el accidente $r_c = l_{i^*c} - s_c$.

Para obtener el tiempo que se gastan las ambulancias desde sus puntos de despacho hasta los incidentes, se usa el tiempo que recorre la ambulancia durante la red vial a la velocidad máxima, utilizando la ruta definida por el *shortest-path* en el grafo desde el punto de despacho hasta el accidente.

Figura 6: Representación de los *shortest-paths* entre los puntos de despacho y un accidente



La Figura 6 representa el proceso descrito anteriormente. Para un accidente representado por **A**, en distintos colores se presentan los caminos asociados a los *shortest-paths* que tomaría una ambulancia en cada uno de los 6 puntos de despacho para trasladarse al accidente. Así mismo, para computar estos caminos se utiliza como peso en el grafo el *travel time* o tiempo de viaje que tiene en cuenta la velocidad a la que se puede transitar cada calle, por eso, se evidencia que la ruta desde el punto de despacho 0, 4 y 1 usa vías principales con mayores velocidades. Sin embargo, nótese que existe la posibilidad de que la ambulancia del punto de despacho más cercano no se asigne al incidente, ya que también se tiene en cuenta la disponibilidad de la ambulancia en un momento dado.

[1] Esta formula se encuentra utilizando la técnica de *stars and bars* para problemas combinatorios.

Finalmente, recompensa del episodio $t \in T$ se define como el negativo del tiempo promedio de respuesta de los accidentes[2] $R = -\sum_{c=1}^{k} r_c/k$. El objetivo del agente es encontrar la distribución de ambulancias óptima de forma que se maximice la suma de recompensas a lo largo de los T episodios.

4. Reinforcement Learning

El problema anterior se puede reformular como un problema de bandido multi-armado estacionario, donde se busca balancear la exploración de nuevas acciones para encontrar la configuración óptima y la explotación de acciones cuyo valor estimado es alto. El valor estimado de la acción $a \in \mathcal{A}$ en el episodio t está dado por $Q_t(a)$ y se define como la recompensa promedio obtenida al seleccionar la acción a hasta el momento t:

$$Q_t(a) = \begin{cases} \frac{\sum_{i=1}^{t=1} R_i \mathbb{I}_{A_i=a}}{\sum_{i=1}^{t=1} \mathbb{I}_{A_i=a}} & \text{si } \sum_{i=1}^{t=1} \mathbb{I}_{A_i=a} \neq 0 \\ v_0 & \text{d.l.c} \end{cases}$$

Donde v_0 es el valor inicial. Además, la actualización incremental al elegir la acción a en el episodio t es:

$$Q_{t+1}(a) = Q_t(a) + \frac{1}{n}(R_t - Q_t(a)) \quad (1)$$

Los métodos que vamos a usar para resolver el problema de despacho óptimo son:

- ε **- Greedy:** En cada episodio $t = 1, 2, \ldots$ el algoritmo selecciona la acción con la recompensa más alta (basado en la información conocida hasta el momento) con probabilidad $1 - \varepsilon$ y selecciona una acción aleatoria con probabilidad ε (Kuleshov and Precup, 2014). La probabilidad de escoger a en el paso t se define por:

  $$P(A_t = a) = \begin{cases} (1-\varepsilon) + \varepsilon/k & \text{si } a = \arg\max_{a \in \mathcal{A}} Q_t(a) \\ \varepsilon/k & \text{d.l.c} \end{cases}$$

  El agente toma la acción A_t, observa la recompensa R_t y actualiza $Q(A)$ según la Ecuación 1.

- **Upper Confidence Bound (UCB):** Este algoritmo balancea exploración y explotación al seleccionar el brazo con el límite confianza superior más alto (Mohan, 2023). La estimación por UCB contiene dos términos: la recompensa estimada y el intervalo de confianza. El intervalo de confianza es proporcional a la raíz del logaritmo del episodio t e inversamente proporcional al número de veces en el que se ha seleccionado la acción $N_t(a)$. Para escoger la acción A_t se define como:

  $$A_t = \arg\max_{a \in \mathcal{A}} Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}}$$

  El agente toma la acción A_t, observa la recompensa R_t y actualiza $Q(A)$ según la Ecuación 1.

- **Gradient Bandit Algorithm (GBA):** Este algoritmo busca aprender preferencias $H_t(A)$ que influyen positivamente en la probabilidad de que la acción A sea elegida en el episodio t:

  $$P(A_t = a) = \frac{e^{H_t(a)}}{\sum_{b=1}^{n} e^{H_t(b)}}$$

  El agente toma la acción A_t, observa la recompensa R_t, actualiza el baseline $\bar{R}_{t+1} = \bar{R}_t + (1/n)(R_t - \bar{R}_t)$, y actualiza las preferencias según:

  $$H_{t+1}(A_t) = H_t(A_t) + \alpha(R_t - \bar{R}_{t+1})(1 - P(A_t))$$

  $$H_{t+1}(a) = H_t(a) - \alpha(R_t - \bar{R}_{t+1})P(a) \quad \forall a \neq A_t$$

  donde α representa el tamaño del paso para actualizar las preferencias sobre las acciones, dependiendo de qué tan diferente es la recompensa a comparación del baseline.

5. Resultados

A continuación, mostramos los resultados experimentales donde comparamos el desempeño y solución obtenida de los algoritmos de ε - Greedy, UCB y GBA. Los parámetros usados para simular el entorno fueron $b = 3$ ambulancias, $m = 6$ zonas de despacho, $k = 18$ accidentes en cada episodio y un tiempo de espera w de 5 minutos.

Primero, se calibraron los hiperparámetros para cada algoritmo con el fin de encontrar el mejor ajuste para cada modelo. Para el caso de ε - Greedy se procede a calibrar el ε, correspondiente a la tasa de exploración. Este parámetro captura el dilema entre exploración y explotación: si es muy alto el agente selecciona la acción con mayor valor estimado menos frecuentemente y si es bajo es posible que se explote una estrategia subóptima. Para el caso de UCB, se calibra la constante de la amplitud del intervalo de confianza (c). Si c es muy alto, el algoritmo busca que todas las acciones se visiten con una frecuencia similar, generando una mayor exploración. Para el caso de GBA se revisó el parámetro α. Entre menor sea el valor de este parámetro, mayor exploración realiza el algoritmo porque las preferencias y probabilidades se actualizan más lentamente.

Con este fin, se realizó un ejercicio clásico de *grid-search*; pero, por temas computacionales, se limitó a 1500 episodios (t) en 10 experimentos (e). Se escogieron los mejores hiperparámetros según los que maximizaran la recompensa promedio por episodio, a lo largo de 10 experimentos dada por $\frac{\sum_{e=1}^{E}\sum_{t=1}^{T} R_{t,e}}{T*E}$.

La Tabla 2 presenta los resultados de la recompensa promedio para cada set de hiperparámetros explorados. En particular, para ε - Greedy un $\varepsilon = 0{,}025$ es el que entrega un mejor tiempo de respuesta promedio con -5.88 minutos. Para UCB un $c = 0{,}05$ ofrece el mejor tiempo de este algoritmo con -5.99 minutos. Finalmente, se encuentra que para GBA un $\alpha = 1/4$ ofrece el mejor tiempo de atención con -6.2 minutos.

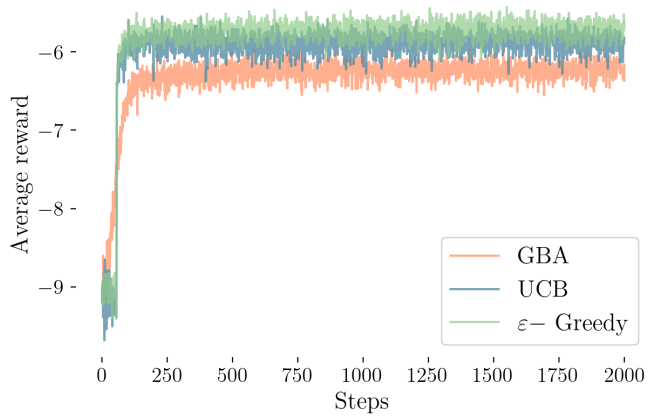
[2]Por simplicidad, se realiza el supuesto que la ambulancia puede movilizarse al límite de velocidad de la vía (i.e. no hay embotellamientos de tráfico).

Tabla 2: Hiperparámetros encontrados para cada método de *Reinforcement Learning*

$\varepsilon - Greedy$		UCB		GBA	
ε	Tiempo promedio	c	Tiempo promedio	α	Tiempo promedio
0.010	-5.993	**0.05**	**-5.995**	1/32	-7.112
0.025	**-5.884**	0.95	-6.034	1/16	-6.435
0.050	-5.888	2	-6.000	1/8	-6.504
0.075	-8.234	4	-6.140	**1/4**	**-6.273**
0.100	-6.118			1/2	-6.383

Tras identificar los mejores hiperparámetros se ejecuta cada algoritmo durante 2000 episodios (t) en 300 experimentos (e) para obtener los resultados globales. La Figura 7 presenta un resumen de las recompensas promedio entre experimentos (e) para cada episodio (t) entre los distintos algoritmos.

Figura 7: Negativo del tiempo de respuesta promedio para cada método a lo largo de 2000 episodios



Tanto en la Figura 7 como en la Tabla 3 se evidencia que el algoritmo de ε-Greedy es el que presenta la mayor recompensa promedio, tanto al tenerse en cuenta todos los episodios (t) como únicamente los últimos 500. En este caso, se tiene que el tiempo de respuesta promedio de los últimos 500 episodios del algoritmo de ε-Greedy es de 5.71 minutos.

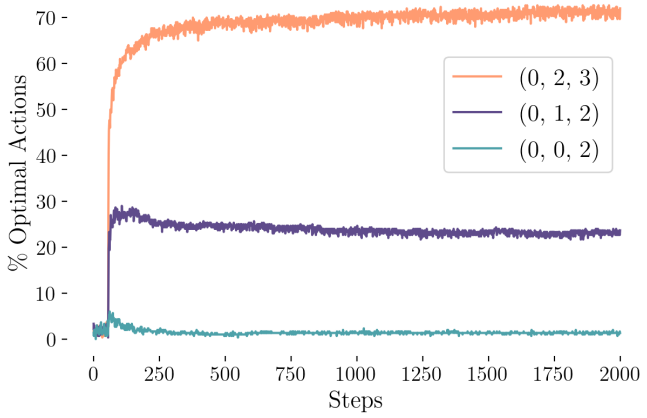
Tabla 3: Resumen del desempeño de los métodos en términos del negativo del tiempo de espera promedio

Algoritmo	**Recompensa Promedio en 2000 episodios**	**Recompensa Promedio en los últimos 500 episodios**
ε - Greedy	**-5.822**	**-5.712**
UCB	-5.980	-5.878
GBA	-6.332	-6.229

Asimismo, se evidencia en la Tabla 3 que tanto ε-Greedy como UCB obtuvieron una recompensa promedio más alta que la del algoritmo de GBA. Lo anterior se puede deber a que la recompensa por cada acción tiene una varianza alta dado que, como se ve en la Figura 3, hay diversos hotspots en la zona de estudio. Esto ultimo genera que cuando los accidentes cambian de una zona a otra la recompensa varíe de manera significativa. Los algoritmos de ε-Greedy y UCB permiten mayor exploración con lo que se encuentran con acciones que generan tiempos de respuesta promedio más bajos.

Por otro lado, se tiene que el algoritmo de ε-Greedy genera mejores recompensas promedio que UCB. Esta ventaja de ε-Greedy se puede entender a partir de la exploración continua que hace el algoritmo a lo largo de los episodios, que le permite explorar más el espacio de acciones y encontrar la acción óptima. Esto es particularmente importante en este entorno donde hay acciones subóptimas con recompensas promedio similares a las de la óptima. Estas acciones subóptimas, posiblemente, tienen un intervalo de confianza que se cruza con el de la acción óptima, lo que genera que el algoritmo de UCB converja a una acción subóptima. Es decir, dado que, en el largo plazo, ε-Greedy permite más exploración puede encontrar la acción óptima —más aún cuando hay un espacio de acciones grande—, mientras que UCB converge a una acción subóptima.

Figura 8: Distribución óptima (ε - Greedy)



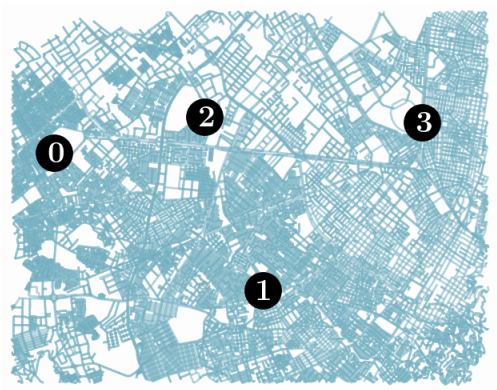
Al analizar las acciones seleccionadas por ε-Greedy a lo largo de cada episodio (t) se tiene que la política óptima es elegir la acción (0, 2, 3) un 71 % de las veces y la acción (0, 1, 2) un 23 % de las veces, como se evidencia en la Tabla 4. El hecho de que la estrategia óptima sea aleatorizar entre ambas acciones se puede explicar debido a que existen zonas de la ciudad donde es más o menos probable que los incidentes ocurran. Asimismo, se evidencia, que la estrategia óptima no incluye acciones donde se asigna más de una ambulancia a cada punto de despacho.

Para comparar, se toman los valores de los últimos 500 episodios (t) para evidenciar que incluso con más episodios, el algoritmo de ε-Greedy va a tener una recompensa promedio superior Figura 8. Los puntos de despacho de las acciones óptimas se pueden ver en la Figura 9.

Tabla 4: Porcentaje de selección de acciones óptimas

Acción	Selección Promedio en 2000 episodios	Selección Promedio en los últimos 500 episodios
(0, 2, 3)	67.223 %	71.077 %
(0, 1, 2)	23.294 %	23.099 %

Figura 9: Puntos de despacho de la distribución óptima



6. Conclusiones

Este documento presentó la aplicación de un problema de asignación modelado como un bandido multi-armado estacionario para encontrar la distribución óptima de puntos de despacho de ambulancias para la atención de accidentes de tránsito en Bogotá. Para ello, se utilizaron datos reales de la ubicación de IPS y de accidentes de tránsito, que fueron simulados para crear la dinámica de atención en cada uno de los escenarios. Para simplificar, se utilizó un área delimitada de Bogotá (aproximadamente el 25 % de la ciudad), con únicamente 6 puntos de despacho. Asimismo, para resolver este problema se utilizaron tres algoritmos: ε- Greedy, UCB y GBA.

Como resultados principales, se encontró que los algoritmos UCB y ε- Greedy obtuvieron recompensas promedio más altas que GBA. Esto podría explicarse a que estos dos algoritmos permiten mayor exploración, pues la recompensa por cada acción tiene una varianza alta debido a la presencia de diversos *hotspots* de accidentes en la zona de estudio. Sin embargo, ε- Greedy tuvo el mejor desempeño, con un tiempo de atención 5,8 minutos promedio por accidente. Esto puede deberse a que este algoritmo permite mayor exploración en el largo plazo, en comparación con UCB y GBA.

Así mismo, se encontró que con 3 ambulancias ubicadas entre 4 puntos de despacho al suroccidente de Bogotá se puede atender una demanda de 18 accidentes al día; y, de acuerdo a las simulaciones, cubrir un área de aproximadamente 10 kilómetros cuadrados. Igualmente, el algoritmo seleccionó como zonas óptimas: i) Universidad Nacional, ii) Avenida 30 de Mayo con carrera 36, iii) el barrio Villa Nelly y iv) el barrio Marsella, de las 6 posibles zonas.

Como posibles extensiones se puede modelar el problema como un escenario no estacionario para capturar patrones de tráfico intenso o estacionales, donde tanto la frecuencia como la ubicación de los accidentes puedan variar según el tiempo. Igualmente, se pueden considerar más zonas de despacho (+ de 6) para mejorar la precisión de la ubicación, aunque, el espacio de acciones crece de forma exponencial. Asimismo, el problema de bandido multi-armado se puede extender a un Proceso de Decisión de Markov (MDP), donde el agente pueda volver a decidir la zona de despacho de la ambulancia cada vez que termina de trasladar al paciente al hospital. Eso permitiría usar métodos de aproximación de funciones o de actor crítico para lidiar con un espacio más amplio de acciones y un entorno más complejo. Por su parte, también se podría complementar esta aplicación utilizando datos a tiempo real o promedios diarios con tráfico y tiempos de desplazamientos entre zonas.

7. Bibliografía

Allen, M., K. Pearn, and T. Monks (2021): "Developing an OpenAI Gym-compatible framework and simulation environment for testing Deep Reinforcement Learning agents solving the Ambulance Location Problem," *arXiv preprint arXiv:2101.04434*.

Bains, G., A. Breyre, R. Seymour, J. C. Montoy, J. Brown, M. Mercer, and C. Colwell (2021): "Centralized Ambulance Destination Determination: A Retrospective Data Analysis to Determine Impact on EMS System Distribution, Surge Events, and Diversion Status," *Western Journal of Emergency Medicine*, 22, 1311, publisher: California Chapter of the American Academy of Emergency Medicine (Cal/AAEM).

Boeing, G. (2017): "OSMnx: New Methods for Acquiring, Constructing, Analyzing, and Visualizing Complex Street Networks," *Computers, Environment and Urban Systems*.

El-Espectador (2016): "Así es el paseo de la muerte de las ambulancias," Section: Más regiones.

Jankovič, P. and Jánošíková (2021): "Ambulance locations in a tiered emergency medical system in a city," *Applied Sciences*, 11, 12160, iSBN: 2076-3417 Publisher: MDPI.

Kuleshov, V. and D. Precup (2014): "Algorithms for multi-armed bandit problems," *arXiv preprint arXiv:1402.6028*.

Mohan, S. (2023): "A brief overview of the Multi-Armed Bandit in Reinforcement Learning," .

Sánchez, S., F. Bedoya, F. Giraldez, and A. Calatayud (2020): "Más congestión, menos tiempo de respuesta ante emergencias," Section: Transporte.