



Trabajo Final de Simulación de Sistemas

Simulación microscópica del tráfico

Alumnos

Pedro López , 60711, Valentino Riera Torraca. 60212

Docente

Parisi, Daniel Ricardo

16 de febrero 2024

Índice

1. Introducción	2
2. Modelo	2
2.1. Intelligent Driver Model (IDM)	2
2.1.1. Aceleración libre	4
2.2. Intersecciones	6
3. Implementación	8
3.1. Clases	8
3.2. Flujo de Simulación	9
3.3. Output	9
4. Simulaciones	10
4.1. Modelo a simular	10
4.2. A*	11
4.3. Parámetros fijos	11
4.4. Parámetros variables	12
4.4.1. Variación en la cantidad de vehículos	12
4.4.2. Variación de velocidad deseada	12
4.4.3. Variación de costo de giro	12
4.5. Observables	12
4.5.1. Proporción de retraso	12
4.5.2. Velocidad promedio	13
4.5.3. Proporción Velocidad promedio sobre Velocidad deseada	13
4.5.4. Proporción de viaje detenido	14
4.5.5. Promedio de Porcentaje de vehículos generados	14
4.5.6. Promedio de momento de generación del último vehículo	14
5. Resultados	15
5.1. Videos	15
6. Análisis de Resultados	31
6.1. Variación en la cantidad de vehículos	31
6.2. Variación en la velocidad deseada	31
6.3. Costo de Giro	31
7. Conclusiones	32
Referencias	33

1. Introducción

A lo largo del tiempo, con el aumento de personas que utilizan vehículos móviles en el mundo, modelos de simulación computarizada de tráfico se han ido necesitando con cada vez más importancia. Debido a la complejidad del problema y todas sus aristas, han surgido muchas ramas en su estudio, siendo dos de las más importantes las simulaciones **macroscópicas**, las cuáles simulan el tráfico de forma similar a un fluido a través de obstáculos, y las **microscópicas**, en las que se distinguen agentes individuales que reaccionan a su entorno inmediato.

En este trabajo práctico exploraremos la implementación de uno de estos modelos, desarrollado por Treivber, Hennecke y Helbing en el año 200, llamado Intelligent Driver Model [2], además de presentar una extensión del modelo para las interacciones entre agentes en las intersecciones de las calles.

2. Modelo

2.1. Intelligent Driver Model (IDM)

En un modelo microscópico, cada vehículo se comporta como un agente, lo que nos hace estar en presencia de un modelo multi agente. Cada agente tiene un índice ordenado i en la calle que está circulando siguiendo al agente $i - 1$.

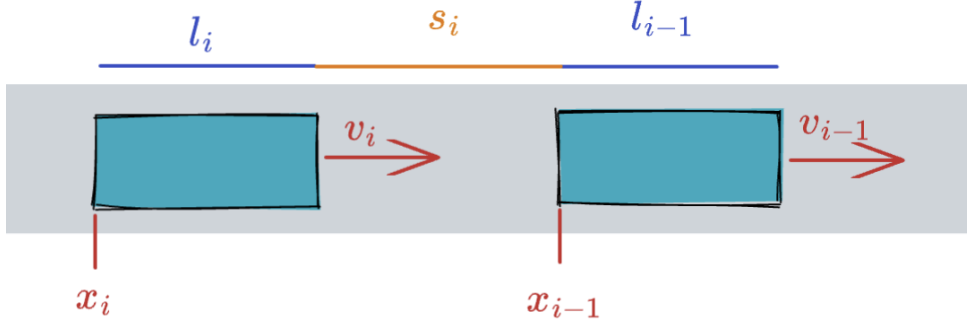


Figura 1: Representación de dos vehículos en una calle.

$$s_i = x_i - x_{i-1} - l_i \quad (1)$$

$$\Delta v_i = v_i - v_{i-1} \quad (2)$$

En la ecuación (1) s_i representa la distancia entre el frente de el vehículo i , y el extremo trasero del siguiente $i - 1$, mientras que en la ecuación (2), Δv_i representa la diferencia entre las velocidades de dos vehículos sucesivos.

En el IDM, se describe la aceleración de un vehículo como una función de las variables del mismo y las del vehículo que está delante [1]. Se describen cómo.

$$\frac{dv_i}{dt} = a_i \left(1 - \left(\frac{v_i}{v_{0,i}} \right)^\delta - \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 \right) \quad (3)$$

$$s^*(v_i, \Delta v_i) = s_{0,i} + v_i T_i + \frac{v_i \Delta v_i}{\sqrt{2a_i b_i}} \quad (4)$$

En donde denotamos:

- $s_{0,i}$: es la distancia mínima deseada entre el vehículo i y $i - 1$.
- $v_{0,i}$: es la velocidad máxima deseada del vehículo i .
- δ : es el exponente de aceleración y controla la "suavidad" de la aceleración.
- T_i : es el tiempo de reacción del conductor del vehículo i .
- a_i : es la máxima aceleración para el vehículo i .
- b_i : es la deceleración cómoda para el vehículo i .
- s^* : es la distancia deseada actual entre el vehículo i y $i - 1$.

- $v_i T_i$: es la distancia de seguridad basada en el tiempo de reacción. Es la distancia que recorre el vehículo antes de que el conductor reaccione (frenando).
- $\frac{(v_i \Delta v_i)}{\sqrt{2a_i b_i}}$: Es una distancia de seguridad basada en la diferencia de velocidad. Representa la distancia que el vehículo necesitará para reducir la velocidad (sin golpear al vehículo de adelante), sin frenar demasiado (la desaceleración debe ser menor que b_i).

Analizando s^* en la ecuación (4) podemos notar que es la suma de 3 distancias:

- La distancia mínima $s_{0,i}$.
- La distancia recorrida en el tiempo de reacción $v_i T_i$.
- La distancia segura de frenado $\frac{(v_i \Delta v_i)}{\sqrt{2a_i b_i}}$

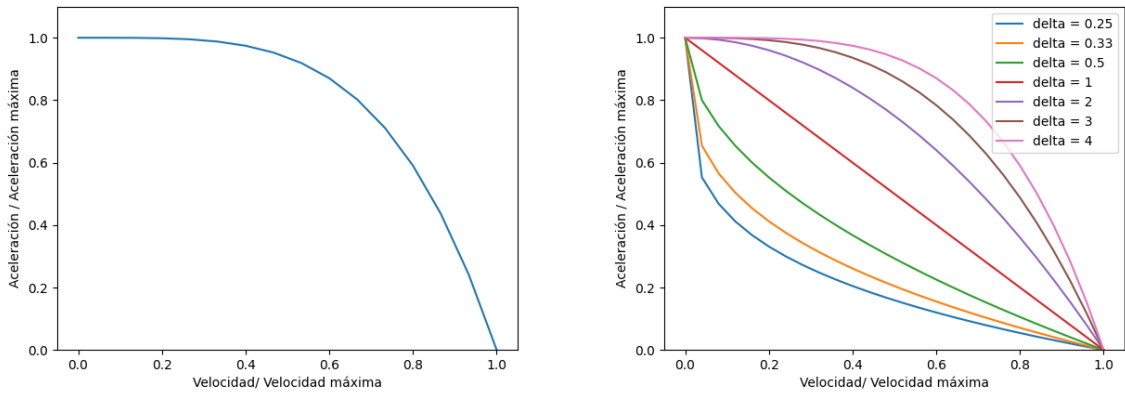
Podemos dividir la aceleración en dos términos:

$$\frac{dv_i}{dt} = a_{freeroad} + a_{interaction} \quad (5)$$

$$a_{freeroad} = a_i \left(1 - \left(\frac{v_i}{v_{0,i}} \right)^\delta \right) \quad (6)$$

$$a_{interaction} = -a_i \left(\frac{s^*(v_i, \Delta v_i)}{s_i} \right)^2 = -a_i \left(\frac{s_{0,i} + v_i T_i}{s_i} + \frac{v_i \Delta v_i}{2s_i \sqrt{a_i b_i}} \right)^2 \quad (7)$$

2.1.1. Aceleración libre



(a) Aceleración porcentual a medida que se acerca a $v_{0,i}$ (b) Curvas de aceleración con distintos valores de δ

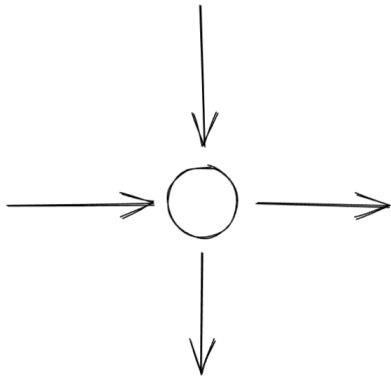
Figura 2: Curvas de aceleración respecto a la velocidad

Siguiendo la aceleración libre, en la que no hay ningún vehículo adelante, podemos ver que a medida que la velocidad v_i se acerca a la velocidad máxima $v_{0,i}$, la aceleración

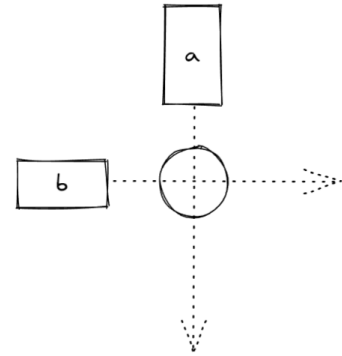
disminuye, como se puede observar en la **Figura 2a**. El exponente δ describe la "suavidad" de la aceleración en la curva a medida que se alcanza la velocidad máxima, como se puede observar en la **Figura 2b**.

2.2. Intersecciones

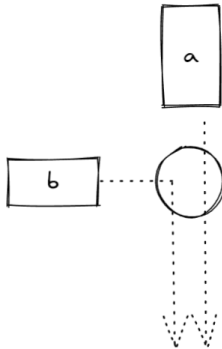
El modelo descrito anteriormente no cubre todas los aspectos de la simulación, el mismo solo denota la interacción en una calle. Nuestra propuesta es conectar estas calles por medio de intersecciones, las cuales son representadas como nodos como se ve en la **Figura 3a**. Estos nodos conectan segmentos de calles, dentro de las cuales las interacciones entre vehículos son como en el IDM. Los nodos tienen calles entrantes y salientes las cuales se asumen que están en ángulos rectos.



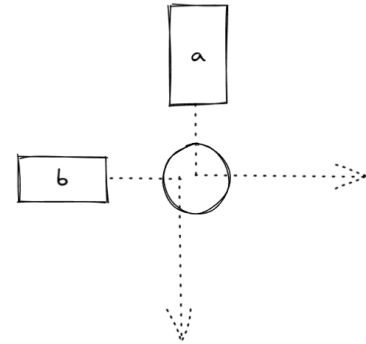
(a) Intersección sin vehículos



(b) Cruce con dos vehículos



(c) Cruce con pérdida de prioridad



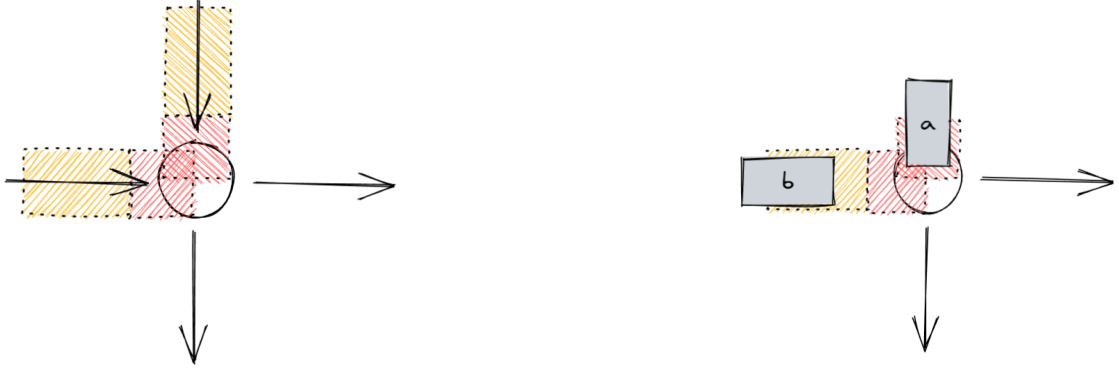
(d) Ambos vehículos doblando

Figura 3: Diferentes interacciones en una intersección

Para los casos en los que vehículos de diferentes calles quieren cruzar se estableció como regla de paso que el vehículo de la derecha siempre tiene prioridad. Esta lógica no está provista por los vehículos si no por los nodos, los cuales pueden poner stop al paso de una calle entrante para dejar pasar a la otra.

En la **Figura 3b** se puede ver un paso en el cual dos vehículos quieren continuar por el mismo sentido en el que venían y aplicando la prioridad de derecha, el nodo pondrá en stop la calle vertical y dejará pasar el vehículo **b**. En el caso de la **Figura 3c** el vehículo

b tiene la intención de doblar a la derecha, por lo que en este caso perdería su prioridad y primero pasaría el vehículo **a**. Finalmente en la **Figura 3d** ambos quieren doblar, por lo que el vehículo **b** tendrá prioridad por estar a la derecha, se pondrá en stop la calle vertical.



(a) Zonas sin vehículos

(b) Zonas mientras pasan vehículos

Figura 4: Visualización de las zonas rojas y amarillas

Para establecer el rango de interacción del nodo, se crean las llamadas zonas rojas y amarillas como se ven en la **Figura 4**. Cuando el auto líder de una calle entra dentro de las zonas es detectado por el nodo y comienza la toma de decisión de paso. El sistema de prioridad anterior entrará en efecto cuando ambos autos estén en la zona amarilla. Si un auto ya estaba por la zona roja el stop de las otras calles entrantes se prenderá y se le da prioridad de paso como se ven en la **Figura 4b**. A nivel vehículo las zonas amarillas y rojas funcionan como zona de desaceleración y frenado respectivamente.

Siendo la velocidad máxima en la zona amarilla $v_{yellow,i}$ como se define en (8), y en la zona roja tomando una aceleración hasta llegar a cero calculada como en (9)

$$v_{yellow0,i} = \alpha * v_{0,i} , \text{ donde } 0 < \alpha < 1 \quad (8)$$

$$\frac{dv_i}{dt} = -b_i \frac{v_i}{v_{0,i}} \quad (9)$$

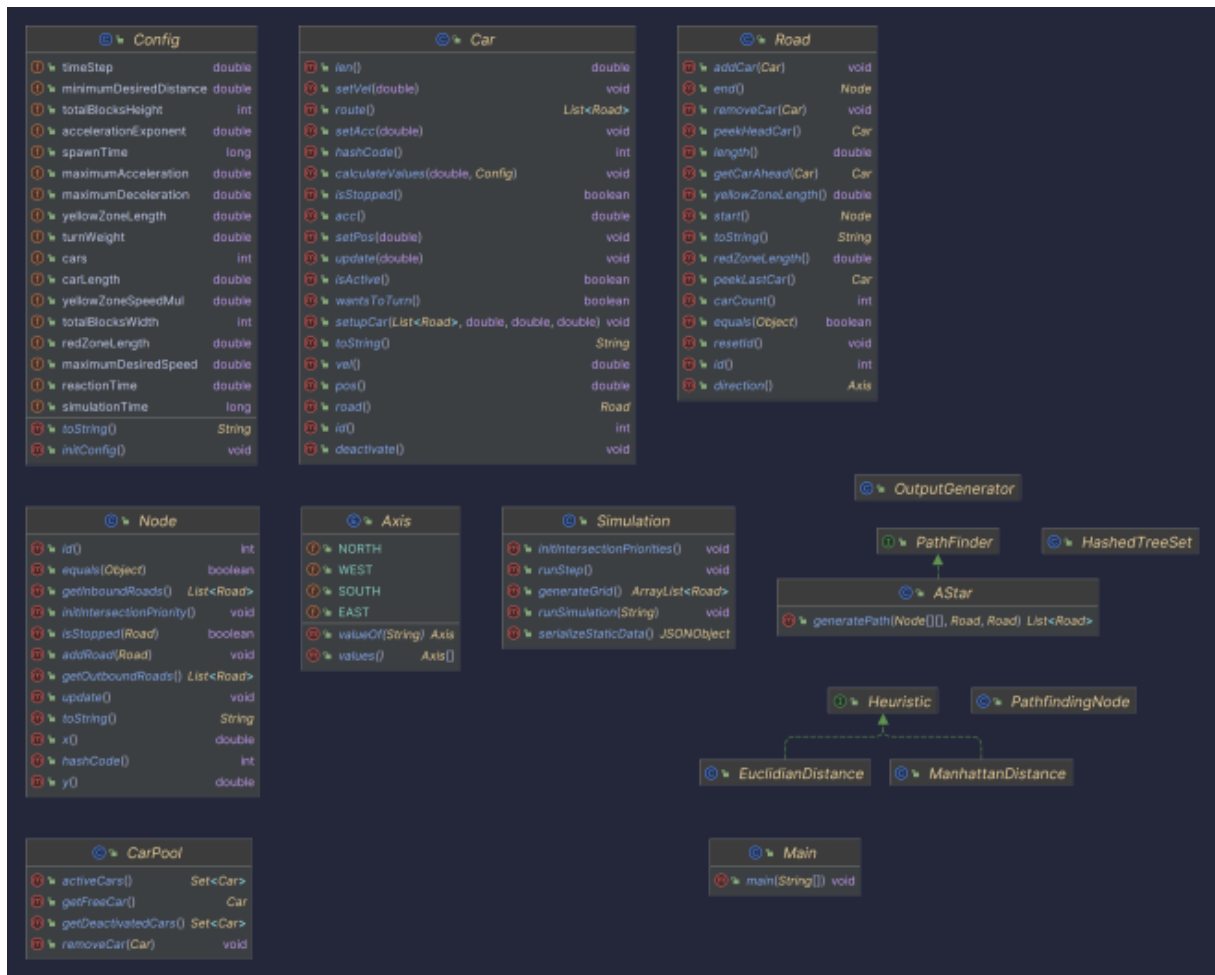


Figura 5: Diagrama de clases de la implementación del proyecto

3. Implementación

3.1. Clases

Desarrollamos el sistema utilizando un acercamiento con Programación Orientada a Objetos. El lenguaje elegido fue Java 19. En la **Figura 5** se detallan las clases y sus respectivos métodos. El código se encuentra disponible en el repositorio [3] Entre ellas podemos destacar a las más importantes:

- **Node**: Representa al nodo en una intersección. Tiene registro de las calles entrantes y salientes, así como de cuáles tienen el paso permitido y cuáles no.
- **Road**: Representa a una calle. Tiene una lista ordenada de los vehículos contenidos en ella y referencias a los nodos en sus extremos.
- **Car**: Representa al agente del vehículo. Tiene datos sobre su posición, velocidad y aceleración.
- **Simulation**: Representa el motor de la simulación. Es quién se encarga de coordinar todos los pasos del flujo de la simulación para su correcto funcionamiento.

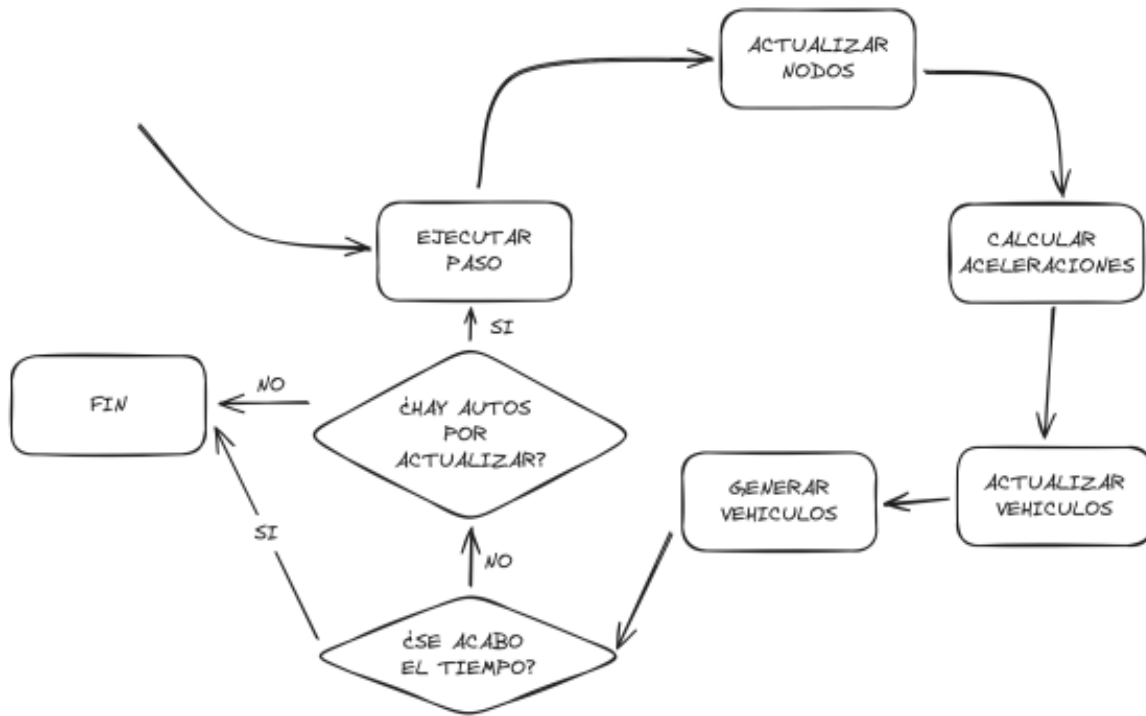


Figura 6: Flujo de Simulación

3.2. Flujo de Simulación

En la **Figura 6** podemos observar cuáles son los pasos que sigue la simulación para ejecutar un paso. Vale la pena aclarar que en el paso de **Actualización de Nodos**, al ser cada Nodo autónomo y no modificar a su entorno, puede paralelizarse dicho paso en la ejecución para aprovechar la ventaja de múltiples núcleos de CPU. Sucede lo mismo con el paso de **Cálculo de Aceleraciones**.

3.3. Output

Se generan 3 archivos de output por cada simulación. *static.json* , *snapshots.json* y *cars.json*

El primero contiene la información estática del sistema, como los parámetros, los nodos y las calles.

El segundo guarda la información de todos los vehículos activos en cada Δt , como *su velocidad*, *su aceleración*, *su posición* , *su calle actual* y *si se encuentra detenido por la zona roja*

El último contiene las rutas de cada vehículo generado y si el mismo logró llegar a destino o no.

Dentro de cada Δt

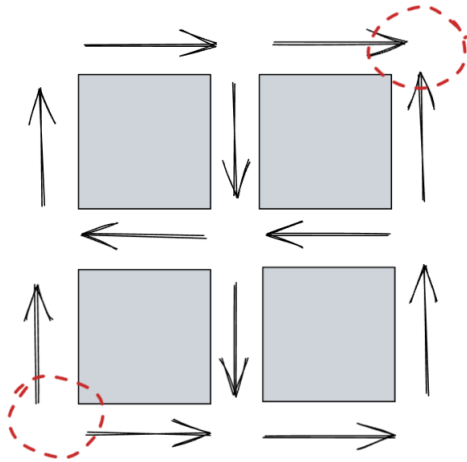
4. Simulaciones

4.1. Modelo a simular

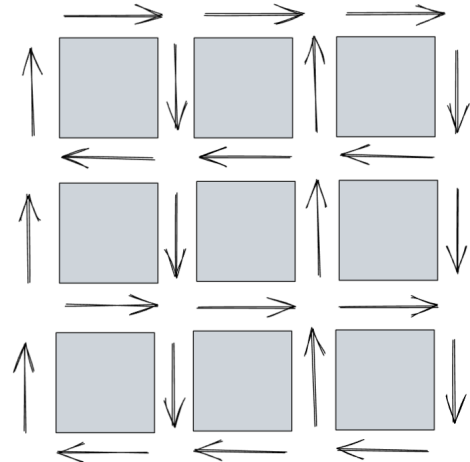
El sistema estará compuesto por un grilla de 20 x 20 nodos. Esto se manifiesta en un área de 19 x 19 cuadras. En las cuales las calles serán unidireccionales y sus direcciones estarán intercaladas.

Es importante denotar que el número de cuadras no puede ser par, ya que si lo fuera existen: un punto inaccesible y otro sin salida, como se ven en la **Figura 7a**. Al tener una cantidad impar de manzanas como se ve en la **Figura 7b** que no existe este problema, una forma fácil de comprobarlo es con la existencia del camino en la dirección de las agujas del reloj que se forma en el borde.

La simulación tendrá un tiempo máximo definido, un número N de autos que se generarán sobre las calles de los bordes del mapa y un tiempo de generación objetivo. Un Vehículo se generará en una calle vacía, en caso de no haber más calles vacías en los bordes, deberá esperar a un momento en el cual si pueda, esto puede llevar a que el tiempo de aparición real sea mayor al tiempo objetivo. Es importante aclarar que todos los vehículos serán idénticos.



(a) Grilla de 2x2 manzanas, no viable



(b) Grilla de 3x3 manzanas, viable

Figura 7: Grilla de manzanas con cantidades pares e impares

Se define estáticamente un Δt que representa un paso en el tiempo, en dicho paso se actualizarán primero todos los nodos y sus respectivas calles entrantes. Luego, se calculará la aceleración de cada auto utilizando el IDM, y una vez calculadas las aceleraciones se actualizarán las posiciones de todos los vehículos utilizando la expresión:

$$v_i(t + \Delta t) = v_i(t) + a_i(t + \Delta t) * \Delta t \quad (10)$$

$$x_i(t + \Delta t) = x_i(t) + v_i(t) * \Delta t \quad (11)$$

En caso de obtener una velocidad negativa, esta será pasada a 0.

Cada vehículo tendrá una posición inicial dentro de una carreta del borde, con una posición a una distancia mínima de cada punta de la carreta igual a una zona amarilla. La posición final tampoco puede ser dentro de la zona amarilla (ni roja).

4.2. A*

Se calculará un camino para llegar al punto destino utilizando el algoritmo A* [4] con una función de costo basada en la distancia del camino mas un posible costo por girar, para la heurística se utilizará distancia de Manhattan.

$$f(x) = g(x) + h(x) \quad (11)$$

$$g(x) = g(x - 1) + d + \omega * \gamma \quad (12)$$

$$g(0) = 0 \quad (13)$$

$$Distancia = |x_{objetivo} - x_{actual}| + |y_{objetivo} - y_{actual}| \quad (14)$$

donde d es la longitud de la calle, γ es 0 o 1 dependiendo de si sigue derecho o gira, respectivamente, y ω es el costo de realizar un giro.

4.3. Parámetros fijos

La simulación tiene un gran número de parámetros, por lo que es importante ver cuáles son los que estudiaremos y cuáles dejaremos fijos. Se decidieron dejar fijos los siguientes parámetros:

- Longitud de vehículo l_i : $5m$
- Tiempo de reacción T_i : $1,5s$
- δ : 4
- Desaceleración cómoda b : $1,67m/s^2$
- Aceleración máxima a : $0,73m/s^2$
- Distancia mínima s_0 : $2m$
- Paso del tiempo Δt : $0,2s$
- Tiempo máximo de simulación: 3600
- Tiempo objetivo de generación de vehículos: $1200s$
- Longitud Zona Roja: $15m$
- Longitud Zona Amarilla: $30m$
- Multiplicador Zona amarilla: 0,75
- Costo de Giro: 0
- Velocidad Máxima deseada: $11,11m/s^2$

4.4. Parámetros variables

Realizaremos 3 experimentos a partir de los siguientes parámetros variables:

- Cantidad de autos N
- Velocidad deseada v_0
- Costo de giro en ω

4.4.1. Variación en la cantidad de vehículos

Para este conjunto de simulaciones se modificó solamente la cantidad de vehículos en cada simulación (N). Tomando los siguientes valores:

$$N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$$

Para cada valor se realizaron 6 simulaciones, dejando $\omega = 0$ y $v_0 = 11, 11m/s$.

4.4.2. Variación de velocidad deseada

Para este conjunto de simulaciones se modificó solamente la velocidad deseada (v_0) en cada simulación, Tomando los siguientes valores:

$$v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$$

Para cada valor se realizaron 8 simulaciones, dejando $\omega = 0$ y $N = 8000$.

4.4.3. Variación de costo de giro

Para este conjunto de simulaciones se modificó el costo de giro (w) en cada simulación, tomando los siguientes valores:

$$w = \{0, 25, 50, 75\}$$

Para cada valor se realizaron 8 simulaciones, dejando $v_0 = 11, 11m/s$ y $N = 8000$.

4.5. Observables

4.5.1. Proporción de retraso

Definimos como proporción de retraso ($\bar{\tau}$):

$$\bar{\tau} = \frac{\sum_{i=0}^N \tau(i)}{N} \quad (15)$$

$$\tau(i) = \frac{t_{recorrido}(i)}{t_{min}(i)} \quad (16)$$

$$t_{min}(i) = \frac{d_i}{v_0} \quad (16)$$

En donde:

- i : Un vehículo obtenido del conjunto de todas las simulaciones
- N : Todos los autos que fueron simulados (La suma de los autos presentes en todas las simulaciones)
- $t_{recorrido}(i)$: Tiempo en el que el vehículo i realizó su recorrido
- d_i : Distancia del recorrido del vehículo i

Esto nos da un indicativo de cuánto tiempo más tardó el vehículo i en recorrer su camino frente al óptimo caso en el que el vehículo llegue a su destino con velocidad máxima constante.

4.5.2. Velocidad promedio

Definimos la velocidad promedio (\bar{v}):

$$\bar{v} = \frac{\sum_{i=0}^N \bar{v}_i}{N} \quad (17)$$

$$\bar{v}_i = \sum_t \left(v_i[t] * \frac{\Delta t}{t_i} \right) \quad (18)$$

En donde:

- i : Un vehículo obtenido del conjunto de todas las simulaciones
- N : Todos los autos que fueron simulados (La suma de los autos presentes en todas las simulaciones)
- \bar{v}_i : El promedio de velocidad del vehículo i
- t_i : Es el tiempo de recorrido total del vehículo i
- $v_i[t]$: La velocidad del vehículo i en t .

4.5.3. Proporción Velocidad promedio sobre Velocidad deseada

Definimos la proporción de como velocidad promedio (\bar{V}):

$$\bar{V} = \frac{\bar{v}}{v_0} \quad (19)$$

En donde:

- i : Un vehículo obtenido del conjunto de todas las simulaciones
- N : Todos los autos que fueron simulados (La suma de los autos presentes en todas las simulaciones)
- \bar{v}_i : El promedio de velocidad del vehículo i
- \bar{v} : El promedio de las velocidad promedio
- v_0 : La velocidad deseada

4.5.4. Proporción de viaje detenido

Definimos como proporción de viaje detenido (\bar{s}):

$$\bar{s} = \frac{\sum_{i=0}^N \bar{s}_i}{N} \quad (20)$$

$$\bar{s}_i = \sum_t \left(s_i[t] * \frac{\Delta t}{t_i} \right) \quad (21)$$

$$s_i[t] = \begin{cases} 1 & \text{si } v_i[t] < \epsilon \quad \text{o } i \text{ está en una zona roja prendida} \\ 0 & \text{en otro caso} \end{cases} \quad (22)$$

En donde:

- i : Un vehículo obtenido del conjunto de todas las simulaciones.
- N : Todos los autos que fueron simulados (La suma de los autos presentes en todas las simulaciones).
- \bar{s}_i : La proporción de viaje que estuvo detenido el vehículo i
- t_i : Es el tiempo de recorrido total del vehículo i .
- ϵ : Velocidad máxima a la que ya se considera detenido a un vehículo.
- $v_i[t]$: La velocidad del vehículo i en t .

4.5.5. Promedio de Porcentaje de vehículos generados

Definimos como porcentaje de vehículos generados ($\% \bar{C}$):

$$\% \bar{C} = \sum_i \left(\frac{n_i}{N} \right) * 100 \quad (23)$$

En donde:

- n_i : Es la cantidad de vehículos generados en la simulación i .
- N : Es la cantidad vehículos objetivo.

4.5.6. Promedio de momento de generación del último vehículo

Definimos como el promedio de momento de generación del último vehículo ($\bar{\theta}$):

$$\bar{\theta}(b) = \frac{\sum_{i=0}^S \text{Max}(b, i)}{S} \quad (24)$$

En donde:

- b : Contiene todos los tiempos de generación , agrupados por su simulación.

- S : Cantidad de simulaciones realizadas.
- $Max(b, i)$: El momento en el que el último vehículo fue generado (tiempo mas alto), para la simulación numero i .

Esto sirve para evaluar que tanto se está saturando la generación de vehículos, y así obtener valores óptimos para otras simulaciones.

5. Resultados

5.1. Videos

- 12000 Agentes: <https://youtu.be/G6Sb60wr2-4>
- 2000 Agentes: <https://youtu.be/sdwzSlq3wyA>
- $v_0 = 22,22m/s$: <https://youtu.be/fKeFkG0AJi4>
- $v_0 = 5,55m/s$: <https://youtu.be/KKL7ty7S6Lo>
- $\omega = 0$: <https://youtu.be/Oz0Ja6mWoMw>
- $\omega = 75$: <https://youtu.be/s2b1werxRHE>

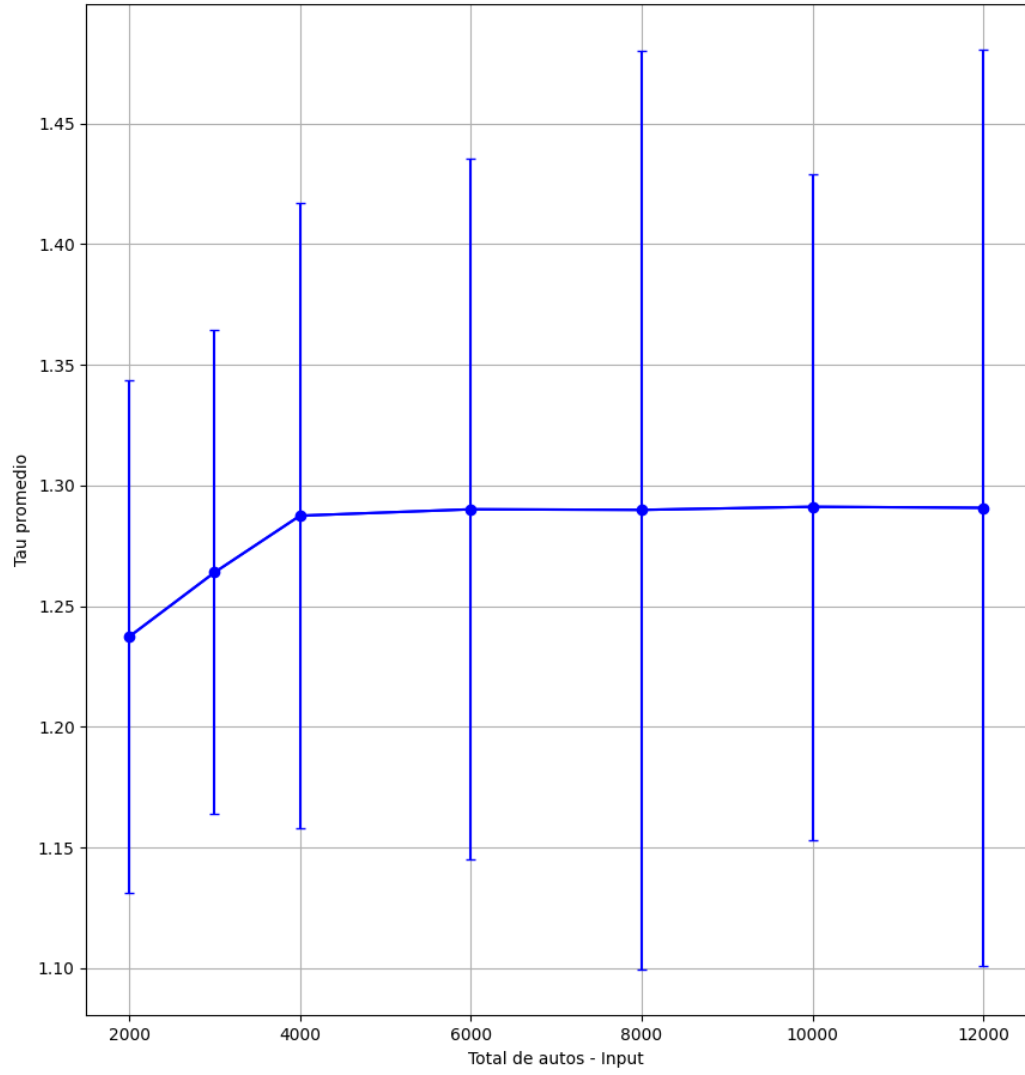


Figura 8: $\bar{\tau}$ (15) en función de la cantidad de autos objetivo, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

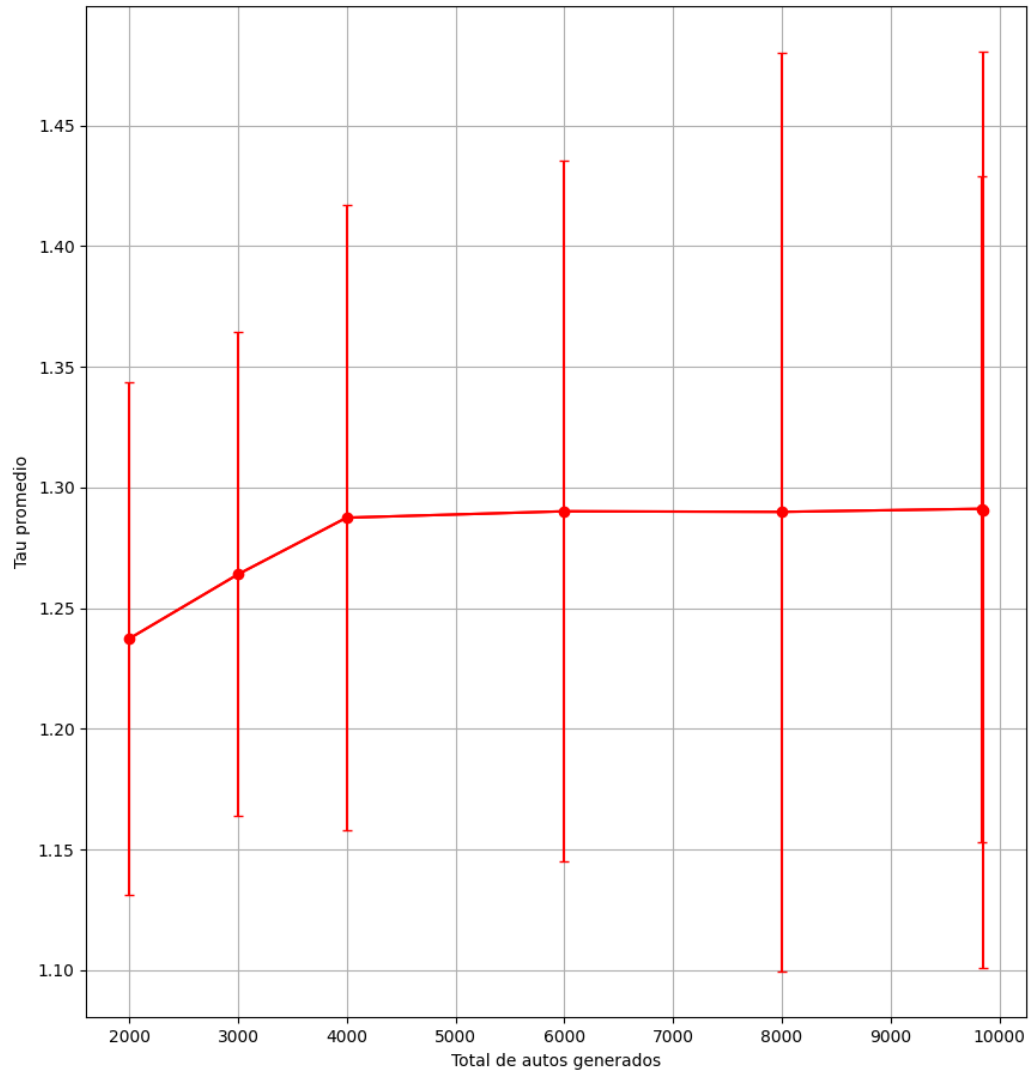


Figura 9: $\bar{\tau}$ (15) en función de la cantidad de autos realmente generados, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

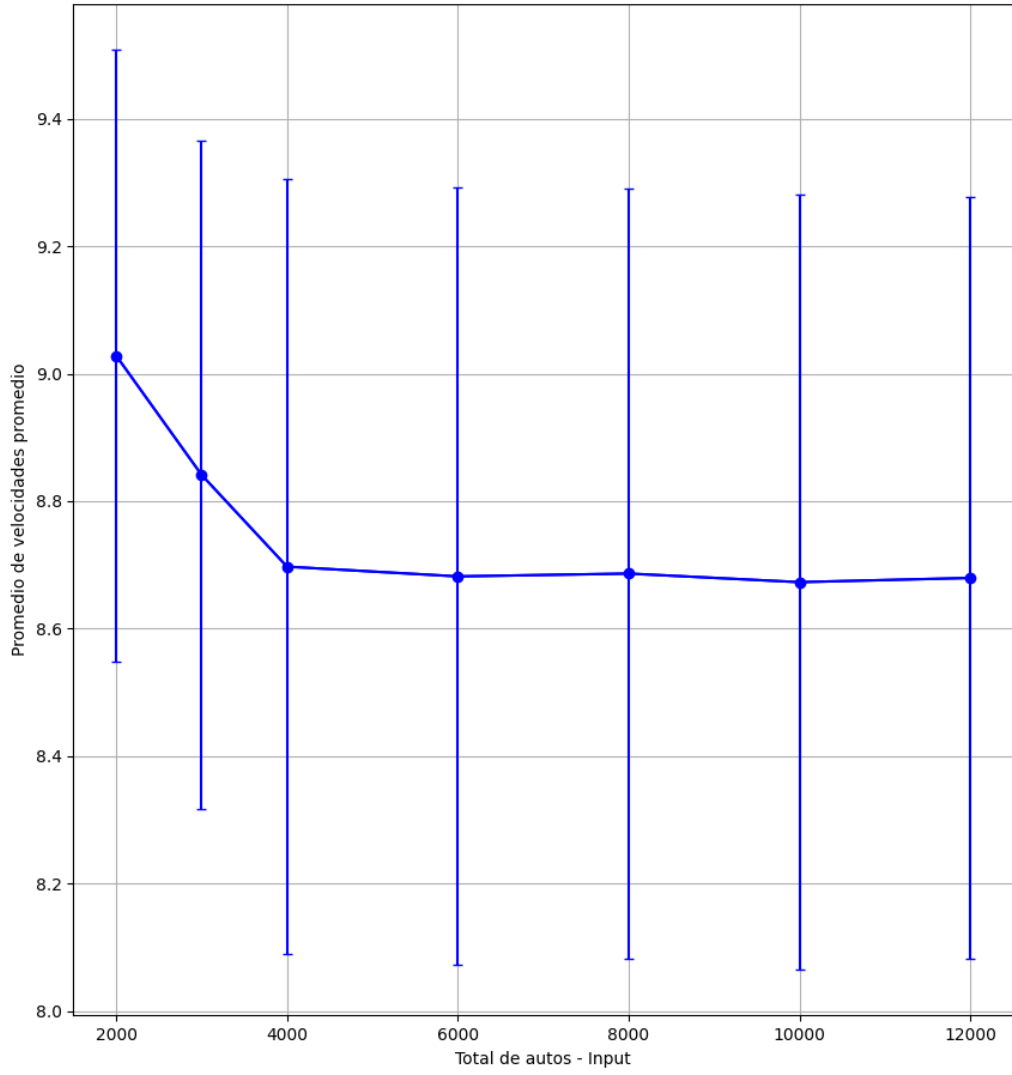


Figura 10: \bar{v} (17) en función de la cantidad de autos objetivo, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

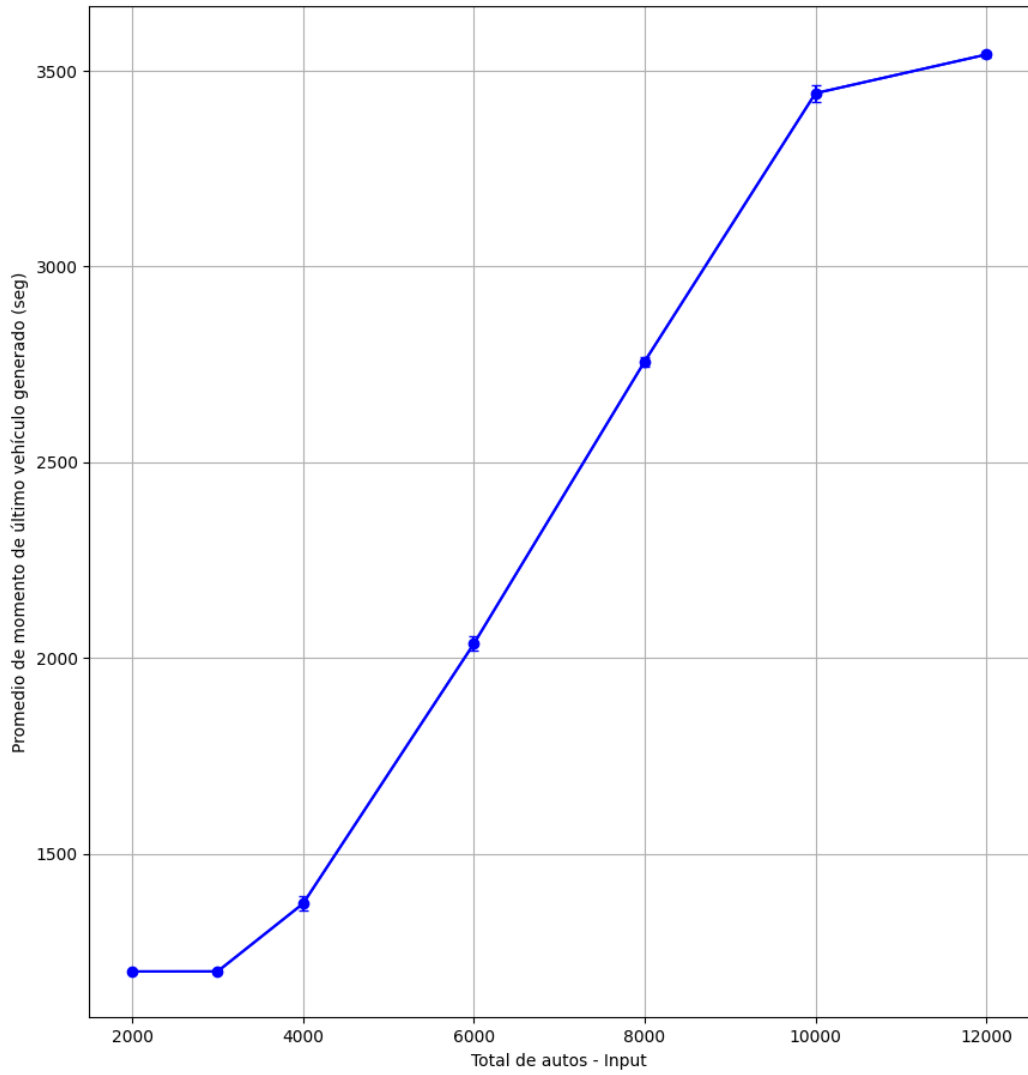


Figura 11: $\bar{\theta}$ (24) en función de la cantidad de autos objetivo, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

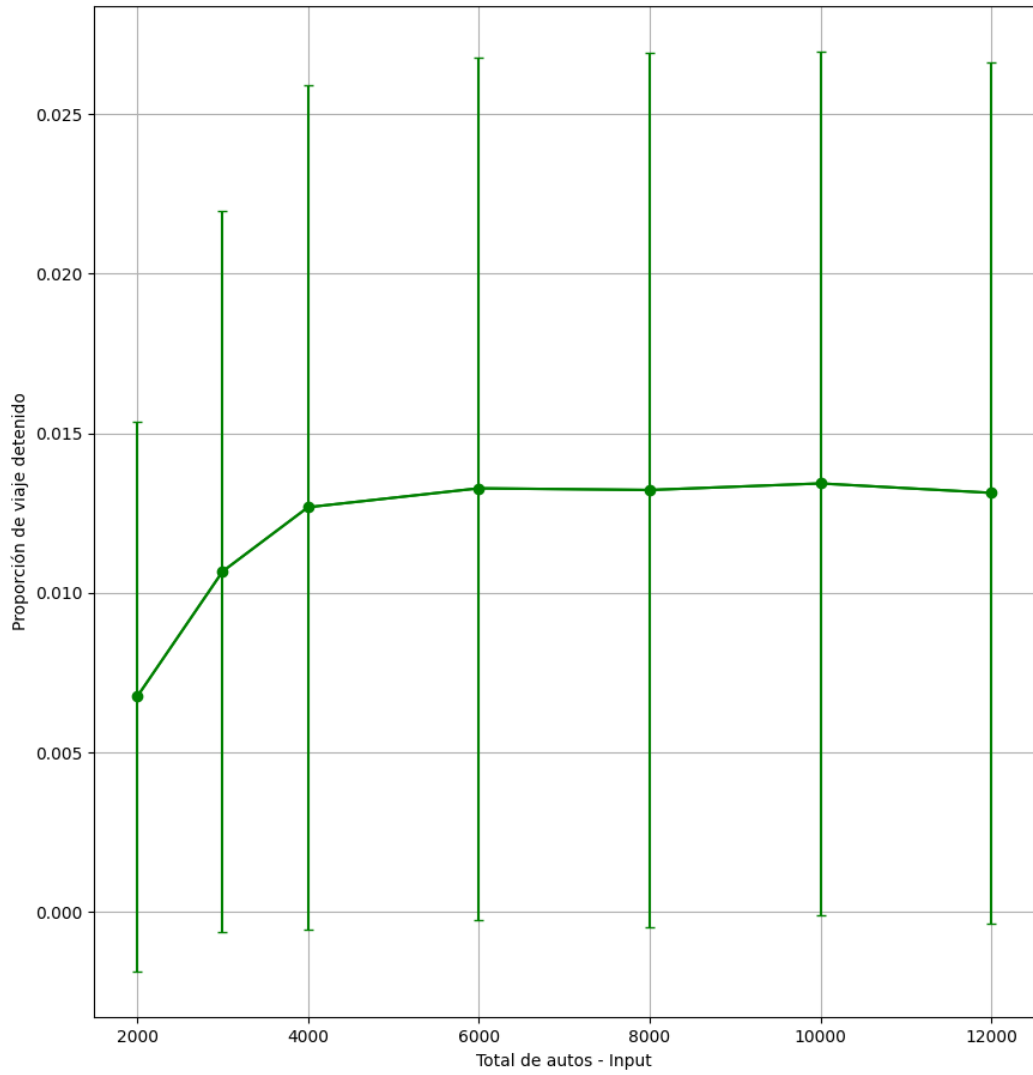


Figura 12: \bar{s} (20) en función de la cantidad de autos objetivo, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

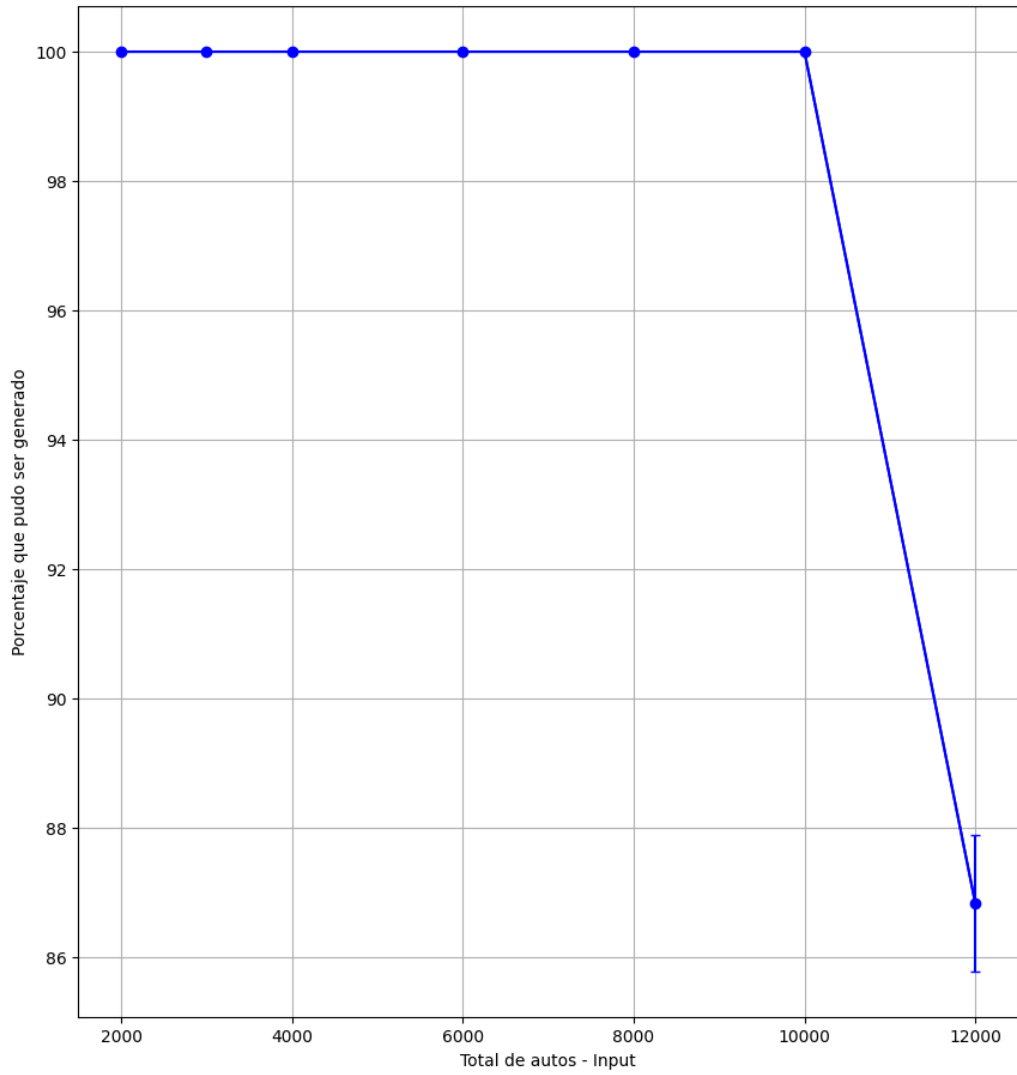


Figura 13: $\% \bar{C}$ (23) en función de la cantidad de autos objetivo, con $N = 2000, 3000, 4000, 6000, 8000, 10000, 12000$

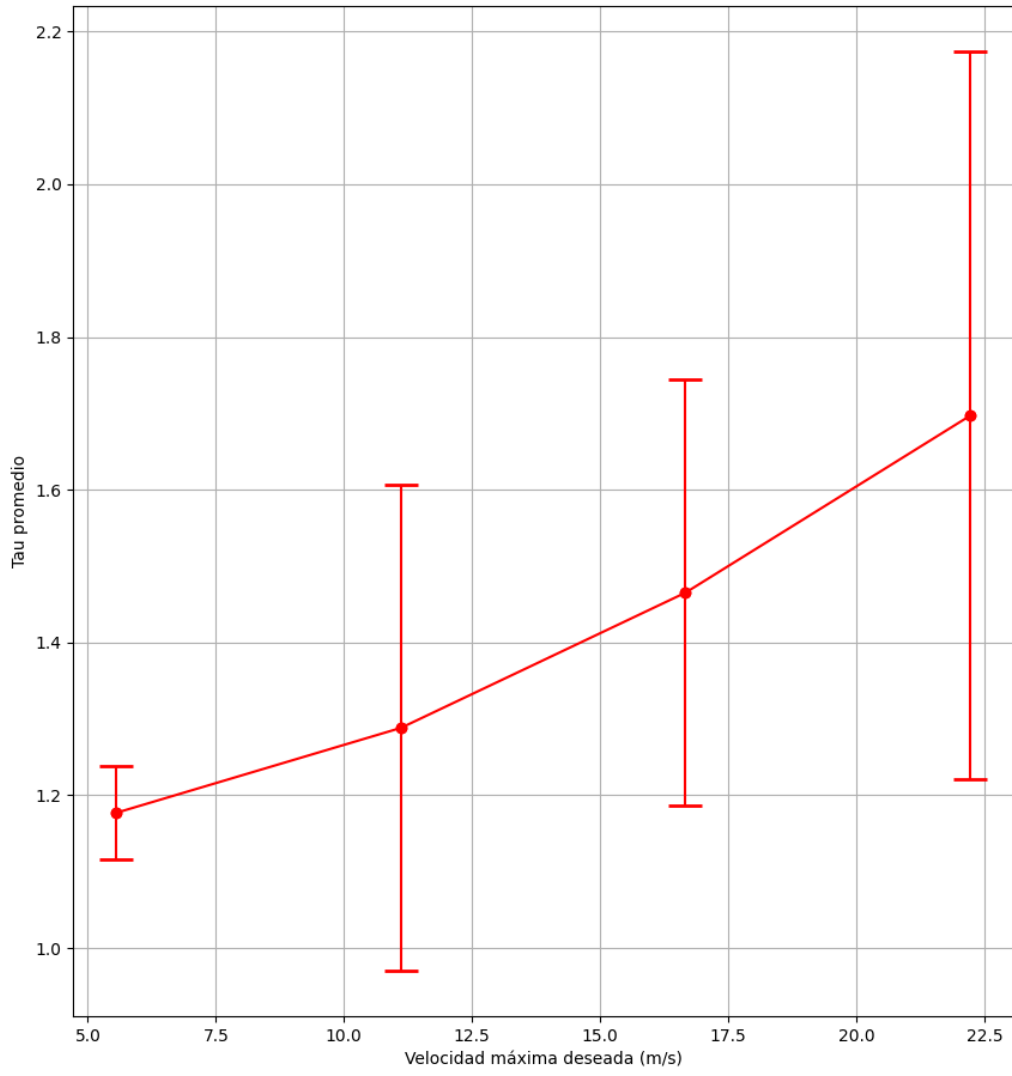


Figura 14: $\bar{\tau}$ (15) en función de v_0 , con $v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$

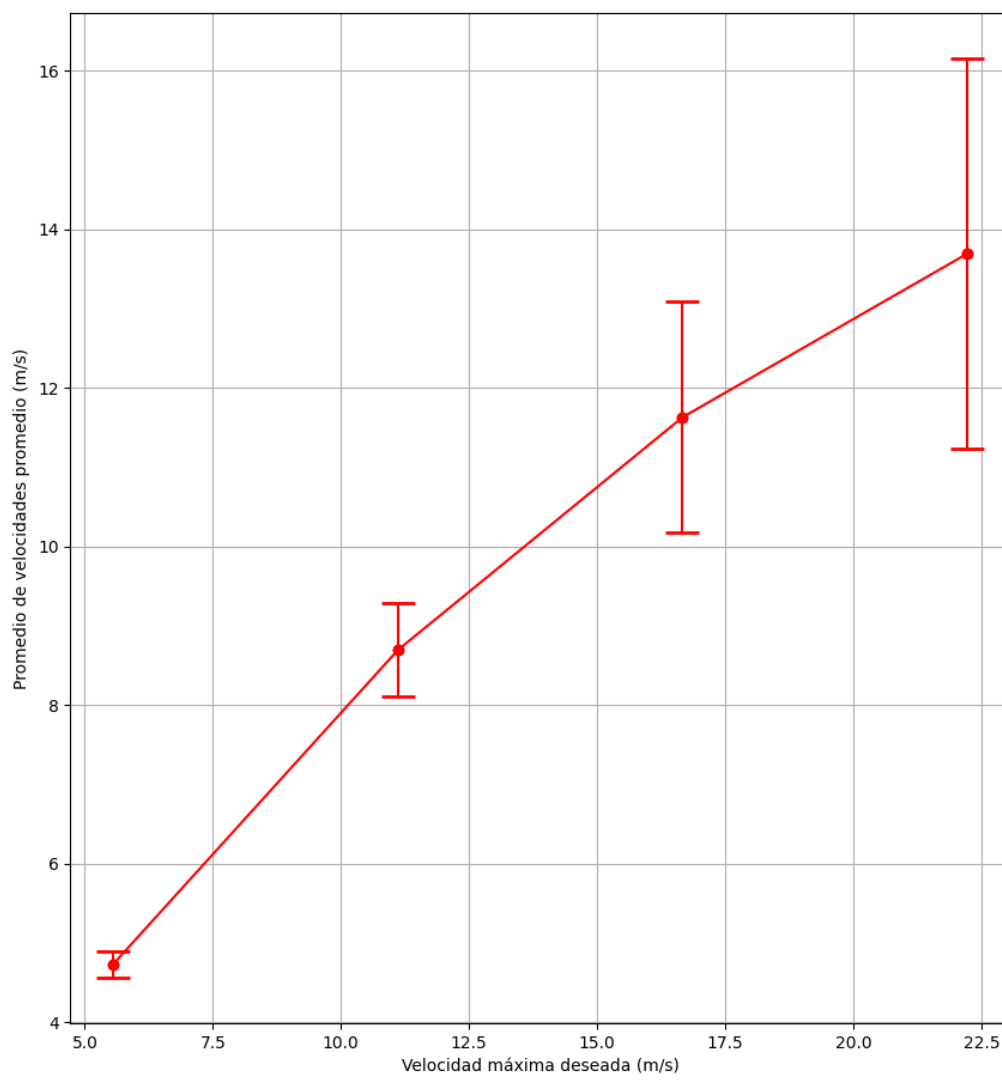


Figura 15: \bar{v} (17) en función de v_0 , con $v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$

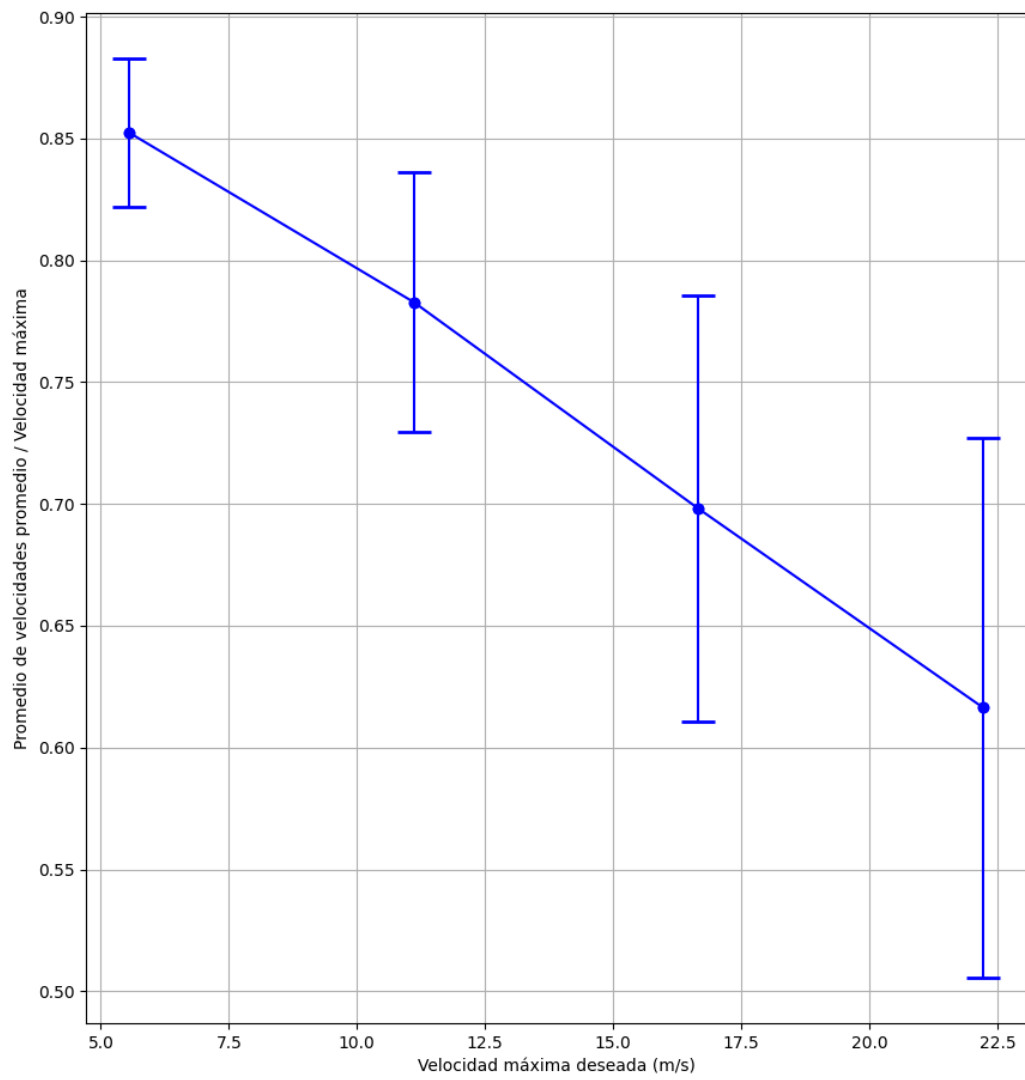


Figura 16: \bar{V} (19) en función de v_0 , con $v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$

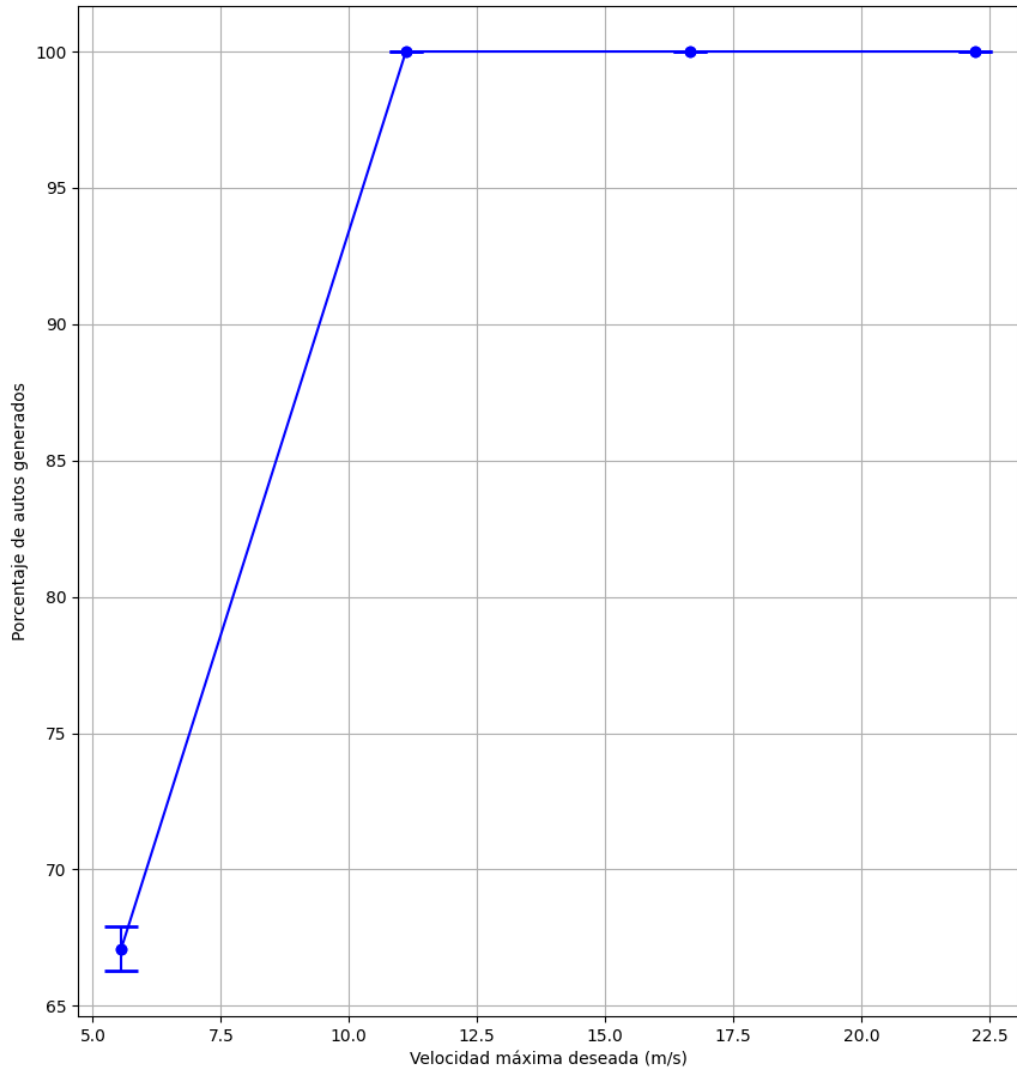


Figura 17: $\% \bar{C}$ (23) en función de v_0 , con $v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$

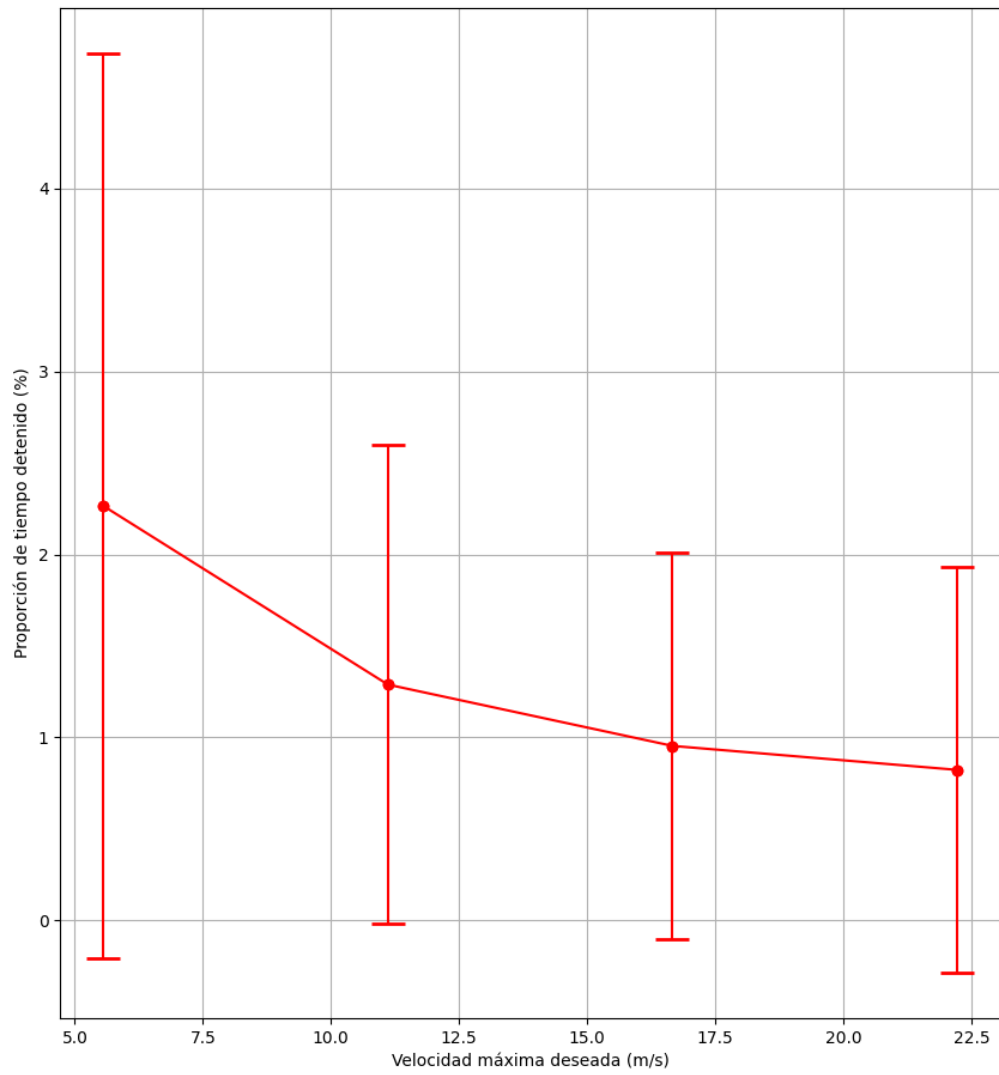


Figura 18: \bar{s} (20) en función de v_0 , con $v_0 = \{\{5,55\}, \{11,11\}, \{16,67\}, \{22,22\}\}(m/s)$

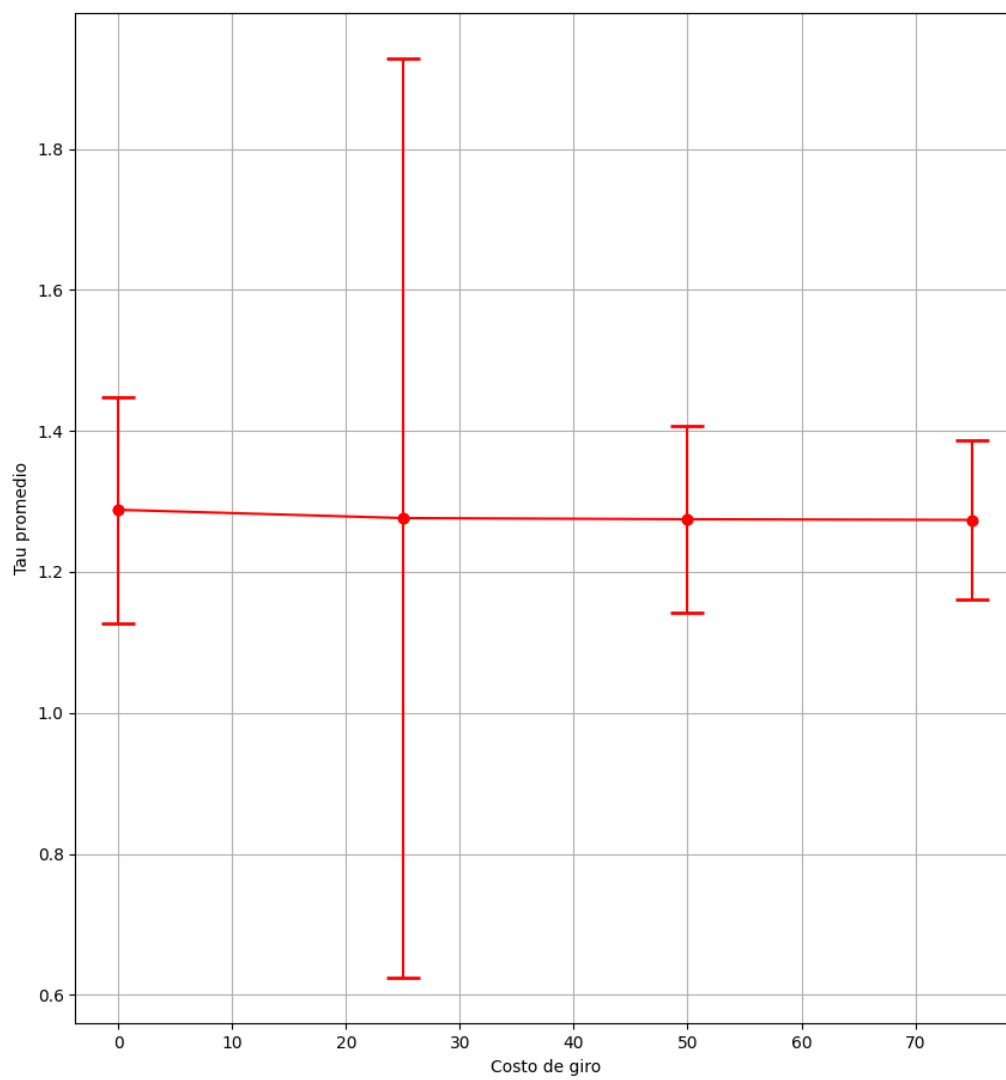


Figura 19: $\bar{\tau}$ (15) en función de w , con $w = \{0, 25, 50, 75\}$

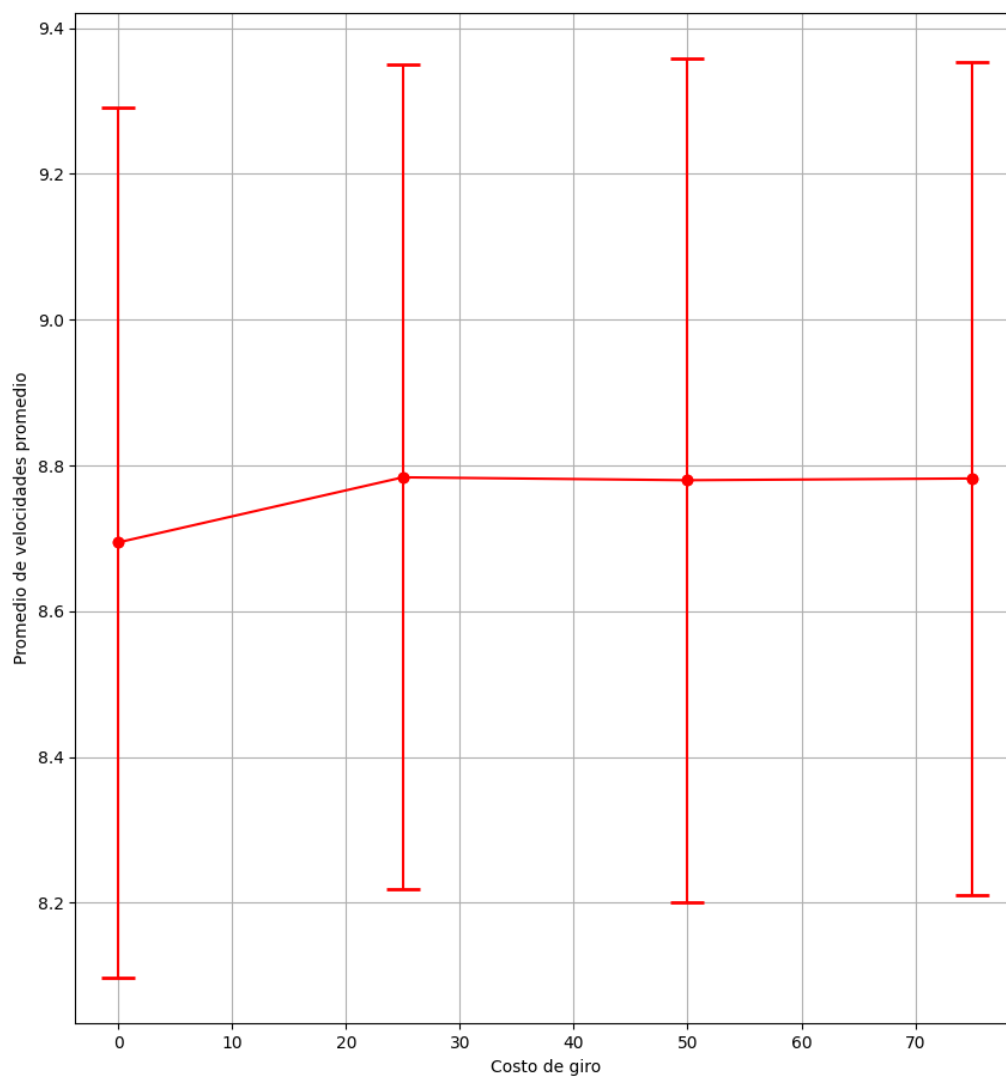


Figura 20: \bar{v} (17) en función de w , con $w = \{0, 25, 50, 75\}$

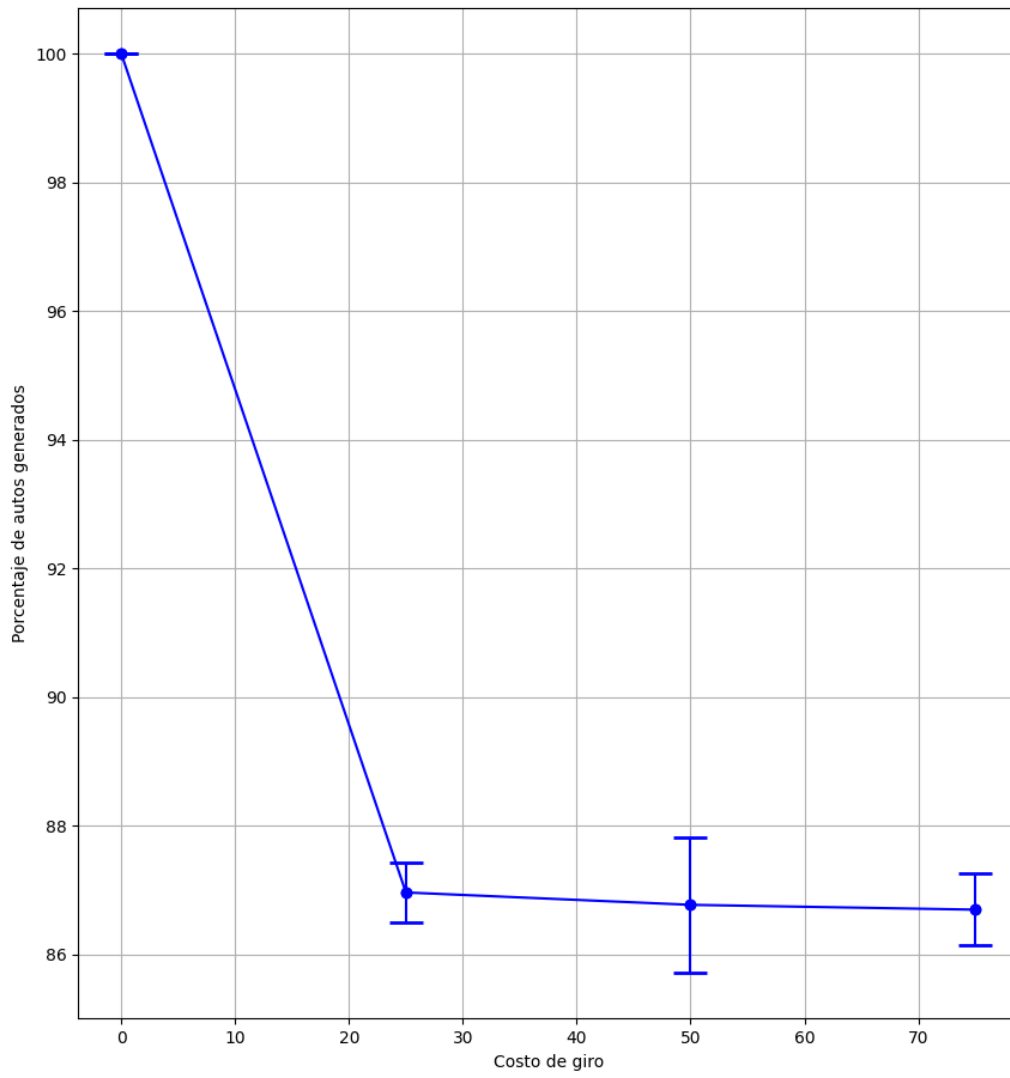


Figura 21: $\% \bar{C}$ en función de w , con $w = \{0, 25, 50, 75\}$

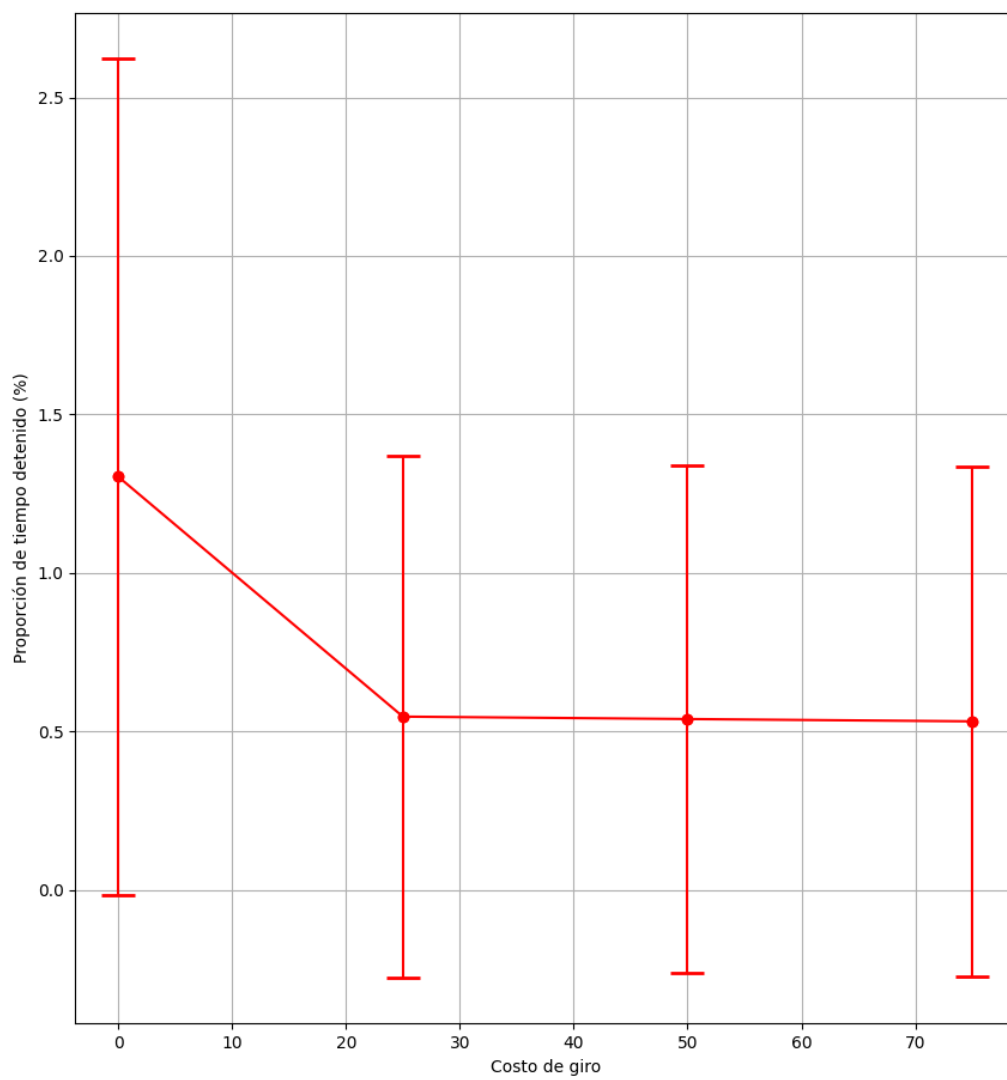


Figura 22: \bar{s} (20) en función de w , con $w = \{0, 25, 50, 75\}$

6. Análisis de Resultados

6.1. Variación en la cantidad de vehículos

Podemos observar en la **Figura 8** cómo el valor de $\bar{\tau}$ incrementa entre los 2000 vehículos y los 4000. Luego se mantiene constante. De manera contraria, podemos observar en la **Figura 10** como la velocidad promedio va disminuyendo a medida que hay más autos presentes, sugiriendo que la congestión aumenta haciendo que un vehículo tenga que frenar más por encontrarse con más autos. Otra observación a destacar es como, a medida que aumenta la cantidad total de autos, el sistema tarda más y más en generar el último. Como fue explicado antes, el objetivo de generación total es de 1200s, y podemos ver como aumenta drásticamente el tiempo en el que el último vehículo fue generado a medida que aumenta la cantidad total, mostrándonos a mayor cantidad de vehículos, menor es la probabilidad de encontrar lugar en los bordes para poder generar uno nuevo. Por último vemos, en la **Figura 11**, como los vehículos aumentan su proporción de tiempo total estando parados en relación a la cantidad, mostrándonos cierta estabilidad desde los 4000 vehículos totales. Además, podemos ver que el sistema no es capaz de trabajar con 12000 vehículos en la **Figura 13**.

6.2. Variación en la velocidad deseada

Podemos observar en la **Figura 14** como el $\bar{\tau}$ va aumentando conforme aumenta la v_0 . Analizando los promedios de velocidad \bar{v} y \bar{V} en las **Figuras 15 y 16** respectivamente, podemos ver como el promedio \bar{v} va aumentando pero el promedio \bar{V} va en descenso. En cuanto al % de tiempo detenido \bar{s} en la **Figura 18** se ve a cuanto mas velocidad deseada, menos porcentaje pasa detenido. Finalmente solo el caso de $v_0 = 5,55(m/s)$ no logra a generar los 8000 autos objetivo en la **Figura 17**.

6.3. Costo de Giro

Podemos ver primero en la **Figura 19** que no hay una influencia en el costo de giro con el valor de $\bar{\tau}$. Por otro lado, vemos un efecto tanto en la **Figura 20** como en la **Figura 21**. En la primera podemos observar como el girar menos (teniendo más peso en el costo de giro) hace que se tenga un leve incremento en la velocidad promedio. Esto solo ocurre en el paso de 0 a 25, luego se mantiene constante. Sucede algo parecido en la segunda figura, en donde podemos observar que al introducir costo de giro, no generamos el total de vehículos deseados por la simulación. Esto puede deberse a que como los conductores son más reacios a girar, se mantienen en el borde mucho más tiempo, obstruyendo el espacio para que nuevos autos puedan generarse. Podemos ver una ventaja en el costo de giro y es que, como nos indica la **Figura 22**, se pasa un tiempo menor detenido durante el viaje al evitar girar. Esta correlación parece tener sentido al pensar que al girar se pierde prioridad, mientras que al seguir derecho por un camino, sólo se pierde prioridad la mitad de las veces.

7. Conclusiones

Gracias al análisis en la variación de la cantidad de vehículos, podemos notar que éste sistema en particular, con un tiempo de simulación de 3600s y con los parámetros propuestos, tiene capacidad para generar alrededor de 10000 vehículos. Además vemos como generar muchos vehículos constantemente influye negativamente en el tiempo de viaje respecto al óptimo.

En cuanto a la variación de velocidad deseada v_0 es interesante notar como la \bar{V} va en descenso conforme v_0 aumenta, esto es coherente debido a que la aceleración máxima se mantiene constante. Aunque la proporción de tiempo frenado \bar{s} sea menor a mayor v_0 , aún teniendo la ventaja de menos velocidades nulas los vehículos no llegan a acelerar hasta v_0 tan frecuentemente. Los resultados sobre el $\bar{\tau}$ son consistentes con lo explicado anteriormente ya que se alejan de terminar las rutas en el tiempo óptimo a mayor v_0 por tener menor \bar{V} .

Por último, vemos como el costo de giro no representa una mejora en el tiempo, pero si logra hacer que menor cantidad del viaje se pase frenado. El descenso en el porcentaje de autos generados tiene un descenso cuando ω es mayor a cero, que corresponde a una menor tendencia a doblar según viendo la función de costo (12) del A*. Los vehículos se generan en los bordes por lo que al quitarle la tendencia a doblar, harán su ruta por los mismos. Ocupando el espacio y evitando la generación de nuevos vehículos.

Referencias

- [1] Bilal Himite. «Simulating Traffic Flow in Python». En: *Towards Data Science* (2021). URL: <https://towardsdatascience.com/simulating-traffic-flow-in-python-ee1eab4dd20f>.
- [2] Martin Treiber, Ansgar Hennecke y Dirk Helbing. «Congested traffic states in empirical observations and microscopic simulations». En: *Physical Review E* 62.2 (ago. de 2000), 1805–1824. ISSN: 1095-3787. DOI: 10.1103/physreve.62.1805. URL: <http://dx.doi.org/10.1103/PhysRevE.62.1805>.
- [3] Pedro Lopez y Valentino Riera. *Final SS Repositorio*. 2024. URL: <https://github.com/vriera/SS-Final>.
- [4] Wikipedia contributors. *A* search algorithm* — *Wikipedia, The Free Encyclopedia*. [Online; accessed 13-February-2024]. 2024. URL: https://en.wikipedia.org/w/index.php?title=A*_search_algorithm&oldid=1206825447.