

Scalable Simulation of Pedestrian Dynamics with the Helbing-Molnár Social Force Model Using Reticulated Geometries in retQSS

Martin Saied², Rodrigo Castro^{1,2}, and Esteban Lanzarotti^{1,2}

¹ Instituto de Ciencias de la Computación (ICC, UBA-CONICET), Ciudad Universitaria, Pabellón 0+infinito, C1428EGA, Buenos Aires, ARGENTINA

² Departamento de Computación (DC, FCEyN-UBA), Ciudad Universitaria, Pabellón 0+infinito, C1428EGA, Buenos Aires, ARGENTINA
`elanzarotti@dc.uba.ar`

Abstract. Simulating pedestrian dynamics using the social forces model of Helbing and Molnár requires evaluating interactions between all pairs of agents, resulting in a computational cost on the order of $O(N^2)$. In this work, this model is implemented in the μ -Modelica language, using the retQSS methodology which provides grid environments to spatially organize agents and restrict force calculations to local neighborhoods. The implementation is validated by reproducing the spontaneous formation of lanes in two-way corridors, verifying the linear relationship between the number of lanes and the corridor width. Based on this, a systematic study of the simulation performance is conducted, identifying configurations that reduce execution times without degrading the interesting emergent behaviors.

Keywords: QSS , Pedestrian Dynamics , Helbing , μ Modelica , Agent-Based Models.

Simulación escalable del modelo de fuerzas sociales de Helbing-Molnár usando geometrías reticuladas en retQSS

Abstract. La simulación de dinámica peatonal mediante el modelo de fuerzas sociales de Helbing y Molnár requiere evaluar interacciones entre todos los pares de agentes, lo que genera un costo computacional del orden de $O(N^2)$. En este trabajo se implementa dicho modelo en μ Modelica utilizando el entorno retQSS, que provee ambientes reticulados para organizar espacialmente a los agentes y restringir el cálculo de fuerzas a vecindades locales. Se valida la implementación reproduciendo la formación espontánea de carriles en corredores bidireccionales, verificando la relación lineal entre el número de carriles y el ancho del pasillo.

A partir de esta base, se realiza un estudio sistemático de los parámetros del simulador, identificando configuraciones que reducen los tiempos de ejecución sin degradar los comportamientos emergentes.

Keywords: QSS , Dinámica Peatonal , Helbing , μ Modelica , Modelos Basados en Agentes.

1 Introducción

El modelo de fuerzas sociales propuesto por Helbing y Molnár (Helbing & Molnár, 1995) se ha consolidado como uno de los enfoques más utilizados para describir la dinámica del movimiento de peatones en estudios aplicados. En ingeniería civil se ha empleado para derivar cargas peatonales y evaluar vibraciones de puentes (Bassoli & Vincenzi, 2021; Wei et al., 2017). En gestión de estaciones de transporte, se han propuesto esquemas de calibración de parámetros apoyados en video y visión por computadora (Li et al., 2024). En contextos de evacuación bajo estrés, se utilizaron extensiones del modelo, para generar recomendaciones concretas sobre el diseño de salidas y la mitigación de peligros (Helbing et al., 2000). También se incorporan la percepción visual limitada y los campos de visión asimétricos, mostrando mejoras tanto a nivel macroscópico como en el realismo de las colisiones (García et al., 2023). Otras introducen anticipación explícita en las interacciones, mejorando la estabilidad del movimiento en escenarios densos (Hu et al., 2023; Wang et al., 2025). Así mismo, se investigó la incorporación de propiedades heterogéneas entre agentes, incluyendo rasgos de personalidad, variaciones en la percepción o reacción al entorno (Dong et al., 2026) y también mejoras en mecanismos que toman en cuenta grupos sociales (Zhao et al., 2026), así como la respuesta frente a contraflujos y colisiones en espacios congestionados (Yang et al., 2024). Estas modificaciones se enfocan en la capacidad de reproducir *comportamientos emergentes*, que aparecen a partir de fuerzas locales entre peatones, sin que haya coordinación global. En el trabajo original se ilustran dos fenómenos: la formación de carriles en pasillos con flujo bidireccional y la oscilación del flujo peatonal en un cuello de botella (Helbing & Molnár, 1995). En general, se fue ampliando el modelo con distintas extensiones orientadas a capturar comportamientos característicos de escenarios particulares. Según el contexto, se incorporan mecanismos adicionales para reproducir patrones o comportamientos observados en la vida real.

Por otra parte, hay trabajos que se enfocan en mejorar la eficiencia computacional del modelo aplicando estrategias conocidas en la optimización de modelos de agentes. Uno de los enfoques más desarrollados es la paralelización del cálculo de fuerzas, tanto en CPU como en arquitecturas híbridas. Se han presentado implementaciones que utilizan OpenMP para distribuir la carga de simulación entre múltiples núcleos de CPU (Lobo Hernandez et al., 2016), así como esquemas híbridos CPU+GPU que logran acelerar la simulación de multitudes densas

en tiempo real (Skrzypczak & Czarnul, 2022), donde también se discute explícitamente la necesidad de reducir la complejidad algorítmica del modelo. En particular, se destaca que la evaluación directa entre todos los pares de agentes tiene un costo del orden de $\mathcal{O}(N^2)$ (Reynolds, 1987), y que limitar el cálculo de las fuerzas a vecindades cercanas es clave para lograr escalabilidad en escenarios grandes. Por ejemplo, restringir la percepción de los agentes limitando el campo visual en el modelo de fuerzas sociales no solo mejora la plausibilidad de las trayectorias, sino que también reduce el tiempo de cómputo por paso (García et al., 2023).

En este trabajo, analizaremos una versión del modelo hecha en μ -Modelica (F. Bergero et al., 2012), un subconjunto del lenguaje Modelica diseñado específicamente para interactuar con solvers QSS. Utilizaremos retQSS (Santi et al., 2022) para minimizar el tiempo de cómputo del modelo usando geometrías reticuladas con el fin de optimizar el cálculo de las fuerzas. Con esto, buscamos responder las siguientes preguntas: i) ¿Es posible implementar el modelo de fuerzas sociales en μ -Modelica y reproducir los comportamientos emergentes reportados en la literatura? ii) ¿Cuál es el impacto de integrar a la implementación el entorno de ambientes reticulados provista por retQSS? iii) ¿Es posible reducir el costo computacional asociado al cálculo de fuerzas sociales, sin perder la capacidad del modelo para reproducir comportamientos emergentes característicos?

2 Metodología

2.1 Ecuaciones de movimiento y cálculo de las fuerzas sociales

El movimiento de cada peatón se describe mediante un sistema de ecuaciones diferenciales de segundo orden. La dinámica del sistema está dada por:

$$\dot{\mathbf{r}}_\alpha = \mathbf{v}_\alpha, \quad \dot{\mathbf{v}}_\alpha = \mathbf{a}_\alpha, \quad \mathbf{a}_\alpha = \mathbf{F}_\alpha^{\text{total}}/m_\alpha$$

donde $\mathbf{r}_\alpha$ y $\mathbf{v}_\alpha$ representan la posición y la velocidad del peatón α . Estas ecuaciones se integran numéricamente utilizando el solver QSS2 (Fernández & Kofman, 2014), mientras que la aceleración $\mathbf{a}_\alpha$ se actualiza a intervalos de tiempo fijos. La aceleración de un peatón α está determinada por la suma de las fuerzas sociales que actúan sobre el mismo. Suponemos que los peatones responden de manera comparable ante obstáculos u otros peatones, independientemente de su masa, por lo que le asignamos a cada m_α un valor igual a 1.

Luego, la fuerza social total se expresa como una suma de las siguientes contribuciones:

$$\mathbf{F}_\alpha^{\text{total}} = \mathbf{f}_\alpha^{\text{d}}(t) + \mathbf{f}_\alpha^{\text{p}}(t) + \mathbf{f}_\alpha^{\text{o}}(t) \quad (1)$$

$$\mathbf{f}_\alpha^{\text{d}}(t) = \frac{1}{\tau_\alpha} (v_\alpha^0 \mathbf{e}_\alpha(t) - \mathbf{v}_\alpha(t)), \quad (2)$$

$$\mathbf{f}_\alpha^{\text{p}}(t) = \sum_{\beta \neq \alpha} \mathbf{f}_{\alpha\beta}(t), \quad (3)$$

$$\mathbf{f}_\alpha^o(t) = \sum_i \mathbf{f}_{\alpha i}(t). \quad (4)$$

$\mathbf{f}_\alpha^d(t)$ representa la tendencia del peatón α a ajustar su velocidad actual $\mathbf{v}_\alpha$ hacia su velocidad deseada $v_\alpha^0 \mathbf{e}_\alpha$. El parámetro τ_α corresponde al tiempo de relajación y controla qué tan rápido el peatón ajusta su velocidad actual a la velocidad deseada. v_α^0 es la magnitud de la velocidad que el peatón aspira a alcanzar, determinada según $\mathbf{r}_\alpha^d$, donde $\mathbf{r}_\alpha^d$ se refiere al destino final del peatón. La dirección deseada de movimiento queda definida por:

$$\mathbf{e}_\alpha(t) = \frac{\mathbf{r}_\alpha^d - \mathbf{r}_\alpha(t)}{\|\mathbf{r}_\alpha^d - \mathbf{r}_\alpha(t)\|}. \quad (5)$$

$\mathbf{f}_\alpha^p(t)$ agrupa las contribuciones de otros peatones, mientras que $\mathbf{f}_\alpha^o(t)$ representa las fuerzas con obstáculos. $\mathbf{f}_{\alpha\beta}(t)$ y $\mathbf{f}_{\alpha i}(t)$ describen el intento de mantener una cierta distancia de seguridad respecto a otros peatones β y a los obstáculos i . La fuerza de repulsión entre peatones $\mathbf{f}_{\alpha\beta}(t)$ se define como una multiplicación entre una componente angular y otra dependiente de la distancia:

$$\mathbf{f}_{\alpha\beta}(t) = w(\theta_{\alpha\beta}(t)) \mathbf{g}(d_{\alpha\beta}(t)), \quad (6)$$

donde $\mathbf{r}_{\alpha\beta} = \mathbf{r}_\alpha - \mathbf{r}_\beta$ es el vector que une a los peatones, y $\theta_{\alpha\beta}$ es el ángulo entre la dirección de movimiento del peatón α ($\hat{\mathbf{r}}_{\alpha\beta}(t)$) y la línea de visión hacia el peatón β ($\mathbf{e}_\alpha(t)$). Esto modela el hecho de que los individuos reaccionan con mayor intensidad a lo que ocurre frente a ellos que a lo que ocurre detrás. Las funciones w y $\mathbf{g}$ se definen como:

$$\mathbf{g}(r_{\alpha\beta}) = A_p \exp\left(\frac{R_\alpha + R_\beta - r_{\alpha\beta}}{B_p}\right) \hat{\mathbf{r}}_{\alpha\beta}, \quad \hat{\mathbf{r}}_{\alpha\beta} = \frac{\mathbf{r}_\alpha - \mathbf{r}_\beta}{\|\mathbf{r}_\alpha - \mathbf{r}_\beta\|}. \quad (7)$$

$$w(\theta_{\alpha\beta}) = \lambda_\alpha + (1 - \lambda_\alpha) \frac{1 + \cos(\theta_{\alpha\beta})}{2}, \quad 0 \leq \lambda_\alpha \leq 1. \quad (8)$$

El coeficiente $\lambda_\alpha \in [0, 1]$ controla el grado de anisotropía de la interacción: los valores cercanos a 1 reducen la diferencia entre lo que ocurre delante y detrás del peatón, mientras que los valores más pequeños funcionan de manera inversa. El parámetro A determina la intensidad de la fuerza repulsiva, mientras que B controla su alcance espacial, es decir, qué tan rápido decae la fuerza con la distancia. Por su parte, R_α y R_β representan los radios efectivos de los peatones, que modelan el espacio personal mínimo alrededor de cada individuo.

Finalmente, y de forma muy similar a la interacción peatón-peatón, pero sin el efecto de anisotropía, modelamos la repulsión con obstáculos a partir de una función exponencial de la distancia al obstáculo. Sea $d_{\alpha i}(t)$ la distancia desde el peatón α al punto más cercano del obstáculo i , y sea $\hat{\mathbf{r}}_{\alpha i}$ el vector normal que va desde el obstáculo hacia el peatón. Definimos entonces:

$$\mathbf{f}_{\alpha i}(t) = A_o \exp\left(\frac{R_\alpha - r_{\alpha i}}{B_o}\right) \hat{\mathbf{r}}_{\alpha i} \quad (9)$$

Aquí, A_o y B_o controlan respectivamente la intensidad y el alcance de la repulsión, específicamente con obstáculos.

2.2 Actualización de la aceleración mediante el uso del reticulado

Utilizamos el reticulado provisto por retQSS para limitar el conjunto de candidatos. Se consideran únicamente los peatones ubicados en la misma vecindad que el peatón en cuestión, descartando el resto. Definimos la vecindad en cruz, quedándonos con los peatones que están en la misma celda y las adyacentes a la izquierda, a la derecha, abajo y arriba. Asimismo, cada obstáculo se representa como un volumen cuadrado del reticulado clasificado como inaccesible. Para cada volumen considerado, se obtiene su centro y se construye la recta que une dicho centro con la posición actual del peatón. Luego, se calcula el punto de intersección entre ese rayo y el borde del cuadrado. Ese punto se utiliza como referencia para definir la dirección de repulsión. De este modo, la dirección de repulsión se aproxima mediante el vector normalizado desde el punto de intersección en el borde del volumen hacia el peatón.

3 Diseño experimental

Con el objetivo de evaluar tanto la validez del modelo implementado como el desempeño de la plataforma de simulación, se diseñó un conjunto de experimentos estructurados en torno a un escenario base y a una serie de variaciones controladas. Las simulaciones fueron realizadas en una computadora de escritorio equipada con un procesador AMD Ryzen 5 2600 de seis núcleos, 16 GB de memoria RAM y sistema operativo Ubuntu 20.04.6 LTS con kernel 5.15.0. Todos los cálculos fueron ejecutados utilizando únicamente la CPU. El software fue compilado utilizando *g++* con estándar *C++11* para el compilador MicroModelica (mmoc), *gcc/g++* con optimización *-O2* para el motor de simulación QSS y el módulo retQSS.

3.1 Escenario base: corredor bidireccional

El escenario de un corredor bidireccional se compone de dos grupos de peatones que se desplazan en sentidos opuestos. El ancho del corredor se mantiene constante a lo largo de todo su recorrido, y las dimensiones de la geometría no varían. En la Fig. 1 se ve el diseño del escenario que utilizaremos para simular estos corredores bidireccionales. El área que queda arriba o abajo de las líneas que delimitan el espacio que recorren los peatones es inaccesible. Los peatones se dividen en dos grupos, se les asigna un destino ubicado en un lado específico de la geometría y empiezan en un punto al azar. Una vez que los alcanzan, se los retorna al inicio del pasillo, conservando la misma coordenada y con la que abandonaron la geometría en el lado opuesto. Asimismo, se preservan la velocidad y la dirección que tenían previamente.

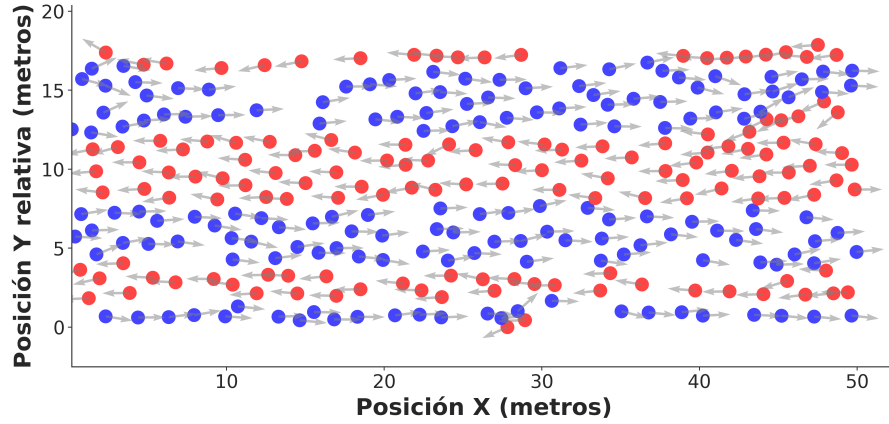


Fig. 1. Ejemplo de agrupamiento de peatones por carriles formados en un corredor de 20m con 300 peatones.

3.2 Estrategia de validación y métricas de evaluación

Para la validación inicial, nos centraremos en reproducir la evolución del número de carriles que se forman cuando se cruzan flujos de peatones en direcciones opuestas. Para lograr esto, diseñamos una heurística que nos permite identificar estos casos con suficiente precisión, de manera de no tener que basarnos únicamente en la detección a partir de visualizaciones de carriles, ya que dicho enfoque no escala bien para analizar un gran número de simulaciones. Para estimar la cantidad de carriles que se forman luego de una corrida del modelo, definimos los siguientes 3 criterios: i) **Clasificación por dirección de movimiento:** Los peatones se dividen en dos conjuntos según el signo de su componente de velocidad en el eje X, separando así los flujos que se desplazan en sentidos opuestos. ii) **Agrupamiento transversal por proximidad:** Para cada conjunto, se identifican agrupamientos espaciales en el eje X del corredor. Dos peatones se consideran relacionados si sus posiciones se encuentran a menos de una distancia en el eje X y otra distancia menor en el eje Y. Esto permite capturar carriles elongados y evita la unificación artificial de carriles consecutivos. Esta relación se cierra transitivamente: si un peatón A está relacionado con B, y B con C, entonces A, B y C conforman un mismo carril. iii) **Promedio en estado estable:** Dado que el número de carriles puede variar al inicio de la simulación o verse afectado por modificaciones aleatorias, se realiza un promedio en una ventana temporal, lo que disminuye la influencia de fluctuaciones locales o de configuraciones iniciales específicas. El resultado final es un promedio de carriles observados en una ventana de tiempo muestreada una vez cada 0.1 segundos.

Por último, para cada corrida del modelo mediremos el tiempo de usuario de manera de conocer el tiempo que tarda en ejecutarse cada simulación.

3.3 Optimización de los parámetros del sistema

Este modelo incluye diversos parámetros distribuidos entre las diferentes partes del sistema. Para cada uno de ellos, se define un valor de referencia que servirá como punto de partida para la posterior búsqueda de parámetros óptimos. Estos valores de referencia se muestran en la Tabla 1.

Table 1. Parámetros del modelo.

Símbolo	Descripción	Valor usado
ΔQ_{rel}	Error relativo de la simulación	$1e-5$
ΔQ_{min}	Error absoluto mínimo permitido	$1e-8$
M	Tamaño del reticulado de retQSS	1
dt	Intervalo entre cálculo de fuerzas	0.1 s
τ_α	Tiempo de relajación	0.5 s
$\mathbf{v}_\alpha^0$	Velocidad deseada	$1.34 \pm 0.26\text{ m/s}$
A_p	Intensidad repulsión peatón-peatón	$2.1\text{ m}^2/\text{s}^2$
B_p	Rango repulsión peatón-peatón	0.3 m
A_o	Intensidad repulsión con obstáculos	$10\text{ m}^2/\text{s}^2$
B_o	Rango repulsión con obstáculos	0.3 m
λ	Factor de percepción direccional	0.3
r	Radio de peatón	0.3 m
C_x	Ancho del corredor	2 m a 20 m
C_y	Largo del corredor	50 m
ρ	Densidad de peatones	0.3 m^{-2}

Durante la resolución numérica de modelos en μ -Modelica, los parámetros ΔQ_{rel} y ΔQ_{min} determinan la precisión con la que el integrador numérico QSS (F. M. Bergero et al., 2018) controla el error máximo relativo y absoluto (respectivamente) de las soluciones aproximadas de las ecuaciones diferenciales. Un valor más pequeño de ΔQ_{rel} implica una simulación más precisa, aunque también un mayor costo computacional, ya que el integrador deberá realizar más pasos de cálculo para mantener el error dentro de los límites especificados. ΔQ_{min} , por otro lado, define el error absoluto mínimo permitido, independientemente de la magnitud de la variable. El parámetro M controla el tamaño del reticulado espacial utilizado por retQSS, es decir, cómo se divide el dominio en volúmenes. Un reticulado más fino (una mayor cantidad de volúmenes) reduce el tamaño del vecindario local y la cantidad de agentes en dicho vecindario. Como valor de referencia inicial, se utiliza $M = 1$, lo que corresponde a un reticulado con una única celda. Este valor se adopta únicamente como punto de partida para el análisis y no como una configuración óptima. La aceleración de cada peatón se calcula como la suma de las contribuciones definidas por el modelo. Finalmente, el resto de los parámetros corresponden tanto a la dinámica de peatones como al escenario del corredor. Los valores de estos parámetros quedan fijos dado que no forman parte de la optimización de la simulación. Estos valores fueron obtenidos

del trabajo original de Helbing y Molnár y de otro estudio posterior (Helbing & Molnár, 1995; Helbing et al., 2000).

4 Resultados

4.1 Reproducción de comportamientos emergentes

Antes de introducir optimizaciones, validamos el correcto funcionamiento del modelo base implementado, analizando la aparición de carriles en el corredor bidireccional. En este experimento, se ejecutaron simulaciones de 150 segundos de tiempo simulado. Realizamos corridas con distintos anchos de pasillo, variando de 2 metros a 20 metros, con incrementos de 1 metro en cada barrido. Ejecutamos cada configuración 20 veces y calculamos el promedio de carriles obtenido durante cada simulación. En la Fig. 2 mostramos cómo se incrementa el número de carriles conforme aumenta el ancho del pasillo. Se observa que la relación sigue una tendencia lineal con un $R^2 = 0.982$. Esta tendencia coincide con lo observado por Helbing y Molnár en el trabajo original (Helbing & Molnár, 1995). Una vez comprobado que la implementación refleja los comportamientos emergentes esperados, pasaremos a determinar los parámetros óptimos mencionados anteriormente con el fin de minimizar los tiempos de ejecución.

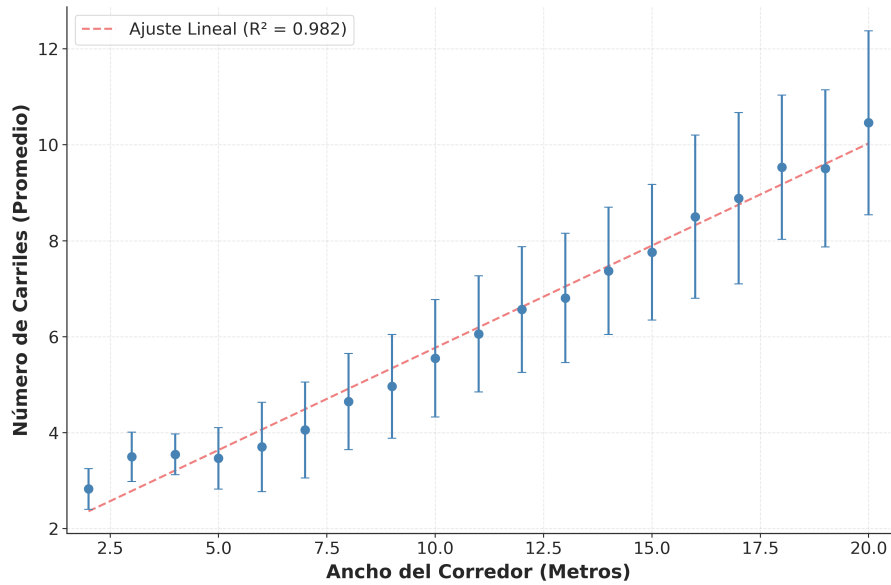


Fig. 2. Número de carriles detectados en función del ancho del corredor.

5 Optimización del ambiente de simulación

Ahora examinaremos cómo influyen los diferentes parámetros del simulador en los tiempos de ejecución y la calidad de las simulaciones. Analizamos la correcta formación de carriles en el escenario de corredor bidireccional a medida que cambiamos los parámetros que hacen que la simulación se ejecute en menos tiempo. Abordaremos la optimización de manera incremental, estudiando primero los parámetros que controlan la precisión numérica del integrador (ΔQ_{rel} y ΔQ_{min}), luego el que afecta la frecuencia de cómputo del modelo (dt) y, por último, el parámetro que regula la cantidad de celdas del entorno reticulado (M).

5.1 ΔQ_{rel} y ΔQ_{min}

Como se explicó previamente, los parámetros ΔQ_{rel} y ΔQ_{min} determinan el *quantum* a partir del cual se genera un cambio en el estado de la posición y la velocidad de los peatones. A medida que se incrementan estos valores, los estados tienen saltos más grandes en sus valores, generando simulaciones con movimientos más bruscos entre los pasos temporales de la simulación. Por convención, el valor de ΔQ_{min} es siempre 3 órdenes de magnitud menor que ΔQ_{rel} , por lo que en esta sección analizaremos únicamente el efecto de modificar ΔQ_{rel} , dado que ΔQ_{min} queda automáticamente definido en función de este último. En la Fig. 3, en los 2 paneles de la primera fila notamos dos aspectos relevantes. En el primer panel notamos que con valores de ΔQ_{rel} mayores a $1e^{-5}$, los tiempos de ejecución se mantienen similares. Además, en el segundo panel se ve que hasta valores de $1e^{-1}$ los carriles se forman correctamente. Dado que no hay mejoras apreciables en los tiempos de ejecución a partir de $1e^{-5}$ y la cantidad de carriles sigue siendo la misma que con valores más chicos, definiremos este valor como el óptimo; que cabe aclarar que es el valor por defecto sugerido por el solver de μ -Modelica.

5.2 Intervalo de actualización de motivación

El primer parámetro analizado es el intervalo de tiempo dt entre actualizaciones de la fuerza que motiva a cada peatón. Valores pequeños implican una actualización más frecuente y, por lo tanto, más precisa, pero con mayor costo computacional. A su vez, intervalos más largos reducen los tiempos de ejecución de las simulaciones a costa de la precisión en el movimiento de los peatones. Esto se debe a que si las actualizaciones de las fuerzas son demasiado espaciadas, impedimos que las fuerzas de repulsión actúen a tiempo para evitar estos comportamientos. En la Fig. 3, en los 2 paneles de la fila del medio, mostramos los resultados de variar este parámetro. Se puede ver que la mejora en los tiempos de ejecución ocurre a medida que incrementamos el dt hasta llegar a $0.2seg$, punto a partir del cual apenas se observan ganancias adicionales. Además, se observa que el número de carriles generados se degrada recién de $0.7seg$ en adelante. Dado que, usando $0.2seg$, las mejoras dejan de ser apreciables, definiremos dicho valor como el óptimo.

5.3 Tamaño de celda y definición de vecindarios

Nos resta buscar un tamaño óptimo de la celda que, en este escenario, nos dé los mejores tiempos posibles. Realizamos un análisis similar al de los anteriores, primero acortando la búsqueda al rango de valores que mantiene el funcionamiento del modelo y la generación de carriles. En los 2 paneles de la última fila de la Fig. 3 podemos observar la formación de carriles y los tiempos de ejecución en función del tamaño de las celdas. Este valor es el que disminuye a medida que aumenta el valor de M que representa la cantidad de divisiones del reticulado. La formación de carriles opera de manera adecuada hasta una distancia de 3 metros por celda, punto a partir del cual el vecindario de cada peatón se vuelve demasiado reducido para que el modelo funcione correctamente. Por encima de 3 metros por celda, observamos que ya no se aprecian cambios importantes en la cantidad de carriles para este escenario, por lo que definimos dicho valor como el óptimo.

6 Análisis de escalabilidad

Ahora realizamos un análisis acerca de qué sucede cuando aumentamos el tamaño del corredor aumentando la cantidad total de peatones manteniendo la misma densidad. Ahora que disponemos de un conjunto de parámetros óptimos para el escenario analizado, compararemos los tiempos de ejecución de tres versiones de la implementación: i) μ Modelica con solvers QSS, sin inicializar el motor de retQSS. ii) utilizando retQSS sin aplicar optimizaciones al cómputo, generando una geometría con solo un gran volumen. iii) retQSS optimizado con el tamaño del reticulado que minimiza los tiempos de ejecución al limitar el cómputo de fuerzas al vecindario de los peatones. Como se puede ver en la Fig. 4, al aplicar el reticulado en los cálculos de las fuerzas sociales, se obtienen tiempos que son entre 2 y 9 veces menores que sin dichas optimizaciones, dependiendo de la cantidad de peatones. En este caso probamos con 1000, 3000 y 5000 peatones, pero estas mejoras se hacen más grandes cuanto más aumentamos la cantidad de peatones.

7 Conclusiones

En este trabajo se ha abordado la implementación y evaluación de un modelo de simulación de movimiento peatonal basado en geometrías reticuladas, con el propósito de mejorar su eficiencia computacional. Los experimentos fueron diseñados para cuantificar las mejoras en los tiempos de ejecución, obtenidas gracias al uso de geometrías reticuladas. Los resultados obtenidos muestran que la metodología propuesta no solo reproduce adecuadamente los fenómenos emergentes esperados, sino que también introduce mejoras significativas en los tiempos de ejecución. En particular, el uso de discretizaciones espaciales reticuladas permite aprovechar de manera más eficiente los recursos computacionales, lo

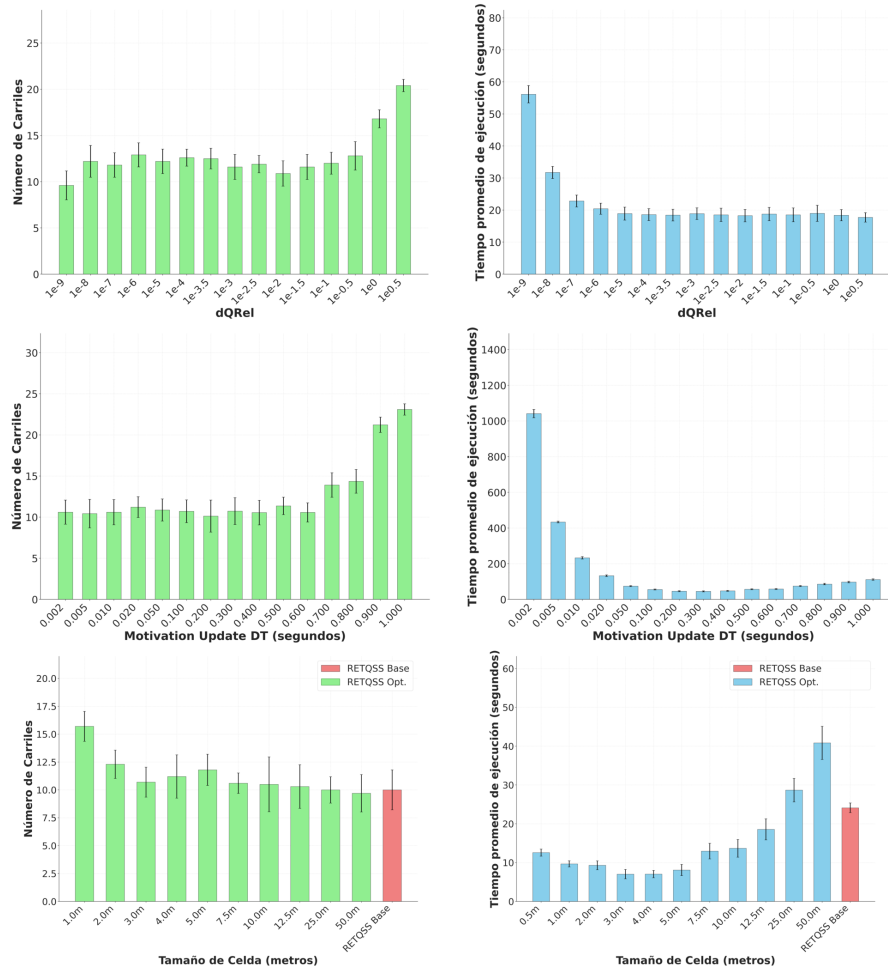


Fig. 3. Búsqueda de parámetros óptimos que permitan obtener los mejores tiempos sin que se perjudique el resultado de la simulación. Se evalúa el número de carriles detectados (columna izquierda) y el tiempo de ejecución (columna derecha), en función de los parámetros del entorno de simulación.

que resulta clave en simulaciones de gran escala o en escenarios que requieren múltiples corridas.

En el diseño experimental se estudia posteriormente el impacto de refinar el reticulado (incrementando M) sobre los tiempos de ejecución en los escenarios estudiados, manteniendo fijos el resto de los parámetros del sistema. Este análisis evidencia un compromiso entre precisión espacial y costo computacional, lo que resalta la importancia de seleccionar adecuadamente el nivel de discretización según el objetivo de la simulación. Asimismo, se analizaron puntualmente los parámetros del simulador μ -Modelica/retQSS que controlan

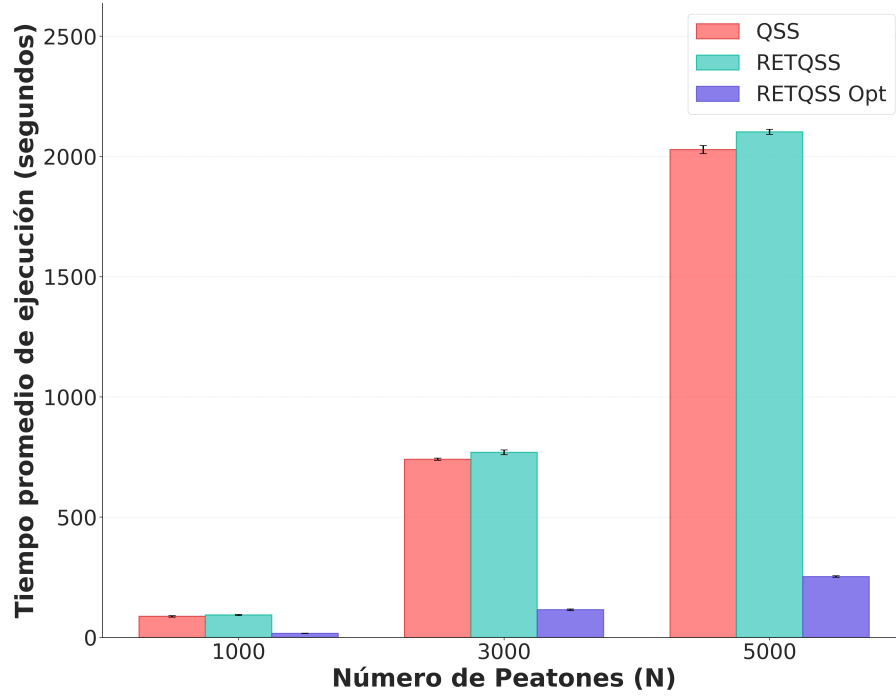


Fig. 4. Tiempos de ejecución para distinta cantidad de peatones.

la precisión numérica del modelo, identificando su influencia directa tanto en la estabilidad como en la eficiencia del modelo. Estos resultados permiten establecer criterios claros para la calibración del sistema.

Finalmente, el objetivo de este estudio fue desarrollar una serie de pasos reproducibles que nos den un valor óptimo de parámetros para otros escenarios a simular, donde las condiciones geométricas y dinámicas sociales sean potencialmente distintas a las actuales. En este sentido, el trabajo no solo aporta una implementación eficiente, sino también una metodología transferible que puede ser aplicada en futuros estudios de simulación de dinámica de partículas.

8 Disponibilidad del software

El modelo de simulación y el software retQSS pueden ser descargados del siguiente repositorio git:

<https://github.com/Martosaied/SocialForceModel-retQSS>.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

- Bassoli, E., & Vincenzi, L. (2021). Parameter calibration of a social force model for the crowd-induced vibrations of footbridges. *Frontiers in Built Environment, Volume 7 - 2021*. <https://doi.org/10.3389/fbuil.2021.656799>
- Bergero, F., Floros, X., & Fernández, J. (2012). Simulating Modelica models with a stand-alone Quantized State Systems Solver. *Proceedings of the 9th International Modelica Conference*, (76), 237–246. <https://doi.org/10.3384/ecp12076237>
- Bergero, F. M., Casella, F., Kofman, E., & Fernández, J. (2018). On the efficiency of quantization-based integration methods for building simulation. *Building Simulation*, 11(2), 405–418.
- Dong, S., Huang, P., Zhang, J., & Wang, W. (2026). An extended social force model considering pedestrian heterogeneity for ship evacuation. *Physics Letters A*, 565, 131155. <https://doi.org/https://doi.org/10.1016/j.physleta.2025.131155>
- Fernández, J., & Kofman, E. (2014). A stand-alone quantized state system solver for continuous system simulation. *Simulation*, 90(7), 782–799.
- García, A., Hernández-Delfin, D., Lee, D.-J., & Ellero, M. (2023). Limited visual range in the social force model: Effects on macroscopic and microscopic dynamics. *Physica A: Statistical Mechanics and its Applications*, 612(100), None. <https://doi.org/10.1016/j.physa.2023.128461>
- Helbing, D., Farkas, I., & Vicsek, T. (2000). Simulating dynamic features of escape panic. *Nature*, 407, 487–490. <https://doi.org/10.1038/35035023>
- Helbing, D., & Molnár, P. (1995). Social force model for pedestrian dynamics. *Phys. Rev. E*, 51, 4282–4286. <https://doi.org/10.1103/PhysRevE.51.4282>
- Hu, X., Chen, T., & Song, Y. (2023). Anticipation dynamics of pedestrians based on the elliptical social force model. *Chaos (Woodbury, N.Y.)*, 33. <https://doi.org/10.1063/5.0149995>
- Li, T., Xu, B., Lu, W., Chen, Z., Zhang, S., & Xia, F. (2024). The parameter calibration of social force model for pedestrian flow simulation based on yolov5. *Sensors*, 24(15). <https://doi.org/10.3390/s24155011>
- Lobo Hernandez, E., Luo, X., Alomía, G., Liu, N., & Zuñiga, C. (2016). How parallelization helps crowd simulation: Study of an openmp-based system, 354–357. <https://doi.org/10.1109/ICVRV.2016.66>
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *SIGGRAPH Comput. Graph.*, 21(4), 25–34. <https://doi.org/10.1145/37402.37406>
- Santi, L., Fernández, J., Kofman, E., & Castro, R. (2022). Retqss: A novel methodology for efficient modeling and simulation of particle systems in reticulated geometries. *Computer Physics Communications*, 270, 108157. <https://doi.org/10.1016/j.cpc.2021.108157>
- Skrzypczak, J., & Czarnul, P. (2022). Efficient parallel implementation of crowd simulation using a hybrid cpu+gpu high performance computing system.

- Simulation Modelling Practice and Theory*, 123, 102691. <https://doi.org/10.1016/j.simpat.2022.102691>
- Wang, J., Lv, W., Cao, S., & Guo, C. (2025). Cosforce: A force-based general pedestrian model with anticipation and reaction mechanisms. <https://arxiv.org/abs/2410.10746>
- Wei, X., den Broeck, P. V., Roeck, G. D., & Nimmen, K. V. (2017). A simplified method to account for the effect of human-human interaction on the pedestrian-induced vibrations of footbridges [X International Conference on Structural Dynamics, EUROLYN 2017]. *Procedia Engineering*, 199, 2907–2912. <https://doi.org/https://doi.org/10.1016/j.proeng.2017.09.331>
- Yang, J., Zang, X., Chen, W., Luo, Q., Wang, R., & Liu, Y. (2024). Improved social force model based on pedestrian collision avoidance behavior in counterflow. *Physica A: Statistical Mechanics and its Applications*, 642, 129762. <https://doi.org/https://doi.org/10.1016/j.physa.2024.129762>
- Zhao, X., Li, W., Mo, Z., Xue, Y., & Wu, H. (2026). Simulation of pedestrian grouping and avoidance behavior using an enhanced social force model. *Sustainability*, 18(2). <https://doi.org/10.3390/su18020746>