

Escuela Politécnica Superior

20
21

Trabajo fin de grado

Diseño y desarrollo de una plataforma de simulación de estrategias de búsqueda bioinspiradas



Carlos Gijón Granados

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C\Francisco Tomás y Valiente Nº 11

**UNIVERSIDAD AUTÓNOMA DE MADRID
ESCUELA POLITÉCNICA SUPERIOR**



Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

**Diseño y desarrollo de una plataforma de
simulación de estrategias
de búsqueda bioinspiradas**

**Autor: Carlos Gijón Granados
Tutor: Roberto Latorre**

mayo 2021

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual. La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 19 de Mayo de 2021 por UNIVERSIDAD AUTÓNOMA DE MADRID,
Francisco Tomás y Valiente, nº 1,
Madrid, 28049,
Spain.

Carlos Gijón Granados

Diseño y desarrollo de una plataforma de simulación de estrategias de búsqueda bioinspiradas

Carlos Gijón Granados

San Fernando de Henares, Madrid. 28830. Avda. Jirón nº 8.

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

A mi familia, mis profesores y mis compañeros

*Fact may not be truth, and
truth may not be factual.*

Haruki Murakami

PREFACIO

Este trabajo ha sido elaborado a partir de una motivación de origen múltiple. En primer lugar, encontramos una motivación que surge del interés científico en el campo de la psicobiología. Con el esfuerzo volcado en este documento, pretendo aportar evidencias acerca de la eficiencia en búsquedas ciegas de diferentes estrategias de búsqueda a través de un estudio teórico computacional.

Por otro lado, existe una motivación que proviene de un lado más relacionado con la ingeniería. La elaboración de este proyecto me ha permitido adquirir experiencia en el desarrollo de herramientas de simulación. Además, como resultado de este trabajo, espero ofrecer a la comunidad científica una herramienta de simulación de estrategias de búsqueda sobre entornos de características variables, con potencial de aplicación en una gran variedad de estudios teóricos de comportamiento. La forma en que se ha diseñado lo hace altamente eficiente, fácilmente customizable y, en caso de necesidad, ampliable de manera sencilla gracias a que cumple con numerosos estándares y patrones de diseño.

Espero que el resultado sea de gran interés tanto por una parte como por la otra, ya que este tipo de proyecto me parece totalmente acertado para acercar los dos polos del eterno enfrentamiento entre ciencia e ingeniería que vive la informática en nuestros días.

Carlos Gijón Granados

AGRADECIMIENTOS

Creo que el primer lugar entre mis agradecimientos debe ir hacia todo el personal docente que durante todos estos años me ha acompañado en mis estudios, sin olvidar la importancia de los docentes presentes durante mi etapa universitaria en la Escuela Politécnica Superior. Gracias por todo vuestro conocimiento.

Deseo agradecer en especial a Roberto Latorre todo el esfuerzo que ha volcado en mejorar este trabajo, cuyo planteamiento y desarrollo no habría sido posible sin su ayuda. También agradezco el punto de vista e interpretación biológicos que ha aportado Gloria Fernández-Lázaro durante su desarrollo.

Quiero dedicar un fragmento de este trabajado a dar las gracias a las personas con las que convivo de forma más cercana. Gracias a mi familia por no parar de motivar mi curiosidad científica desde siempre, y no dejar que pase un día sin hacerme reír. Gracias a mis amigos por apoyarme ante lo bueno y lo malo. Gracias a todas las personas que me demuestran la bondad que existe en la gente. Es vuestro apoyo por el que he conseguido llegado hasta aquí.

No podría terminar este apartado sin agradecer al lector que se dispone a leer el resto de este documento. Gracias.

RESUMEN

Las búsquedas aleatorias de alimento, pareja reproductiva o cualquier otro recurso necesario para la supervivencia o reproducción son inevitables en la naturaleza debido a la falta de conocimiento acerca del entorno que rodea a un animal, donde se encuentran los recursos que éste busca. Es posible observar empíricamente cómo los animales utilizan diferentes estrategias a la hora de afrontar este problema. Es nuestro objetivo en este trabajo diseñar y desarrollar un framework que nos permita analizar la eficiencia relativa de distintas estrategias de búsqueda aleatorias propuestas en la literatura y observadas en la naturaleza, enfocándonos principalmente en los conocidos como Lévy Walks.

Se comienza abordando el marco teórico sobre el que se desarrolla nuestro trabajo, explicando las evidencias que existen acerca del rendimiento de los Lévy Walks frente a otras estrategias de búsqueda. Similarmente, se realiza un análisis de las herramientas software existentes con un propósito similar al nuestro.

Tras explicar las decisiones de diseño e implementación que tuvieron lugar durante el desarrollo de nuestro simulador de estrategias de búsqueda, se expone la batería de experimentos que se realizó con ayuda de éste. Finalmente, se analizan los resultados obtenidos en los experimentos y sus implicaciones en el campo de estudio.

PALABRAS CLAVE

Forrajeo animal, búsqueda aleatoria, eficiencia, Lévy flight, simulación

ABSTRACT

Random searches for nourishment, breeding pair or any other resource necessary for survival or reproduction are inevitable in nature due to lack of knowledge about the environment that surrounds an animal, where the resources it seeks lay. It is possible to empirically observe how animal use different strategies when they have to face this problem. It is our objective in this project to design and develop a framework that allows us to analyze the relative efficiency of different random search strategies proposed in the literature and observed in nature, focusing mainly on what are known as Lévy Walks.

We start by addressing the theoretical framework around which our project is developed, explaining the evidence that exists about the performance of Lévy Walks against other search strategies'. Similarly, an analysis on other software tools with a similar purpose to ours is conducted.

After explaining the design and implementation decisions taken during our random search strategy simulator's development, the set of experiments carried out with its help is set forth. Finally, the results obtained in the experiments and their implications in its field of study are analyzed.

KEYWORDS

Animal foraging, random search, efficiency, Lévy flight, simulation

ÍNDICE

1	Introducción	1
1.1	Motivación	2
1.2	Objetivos	3
1.3	Organización de la memoria	4
2	Estado del arte	5
2.1	Lévy Flight Foraging Hypothesis	5
2.2	Modelos de movimiento animal	6
2.3	Simuladores	10
2.3.1	Netlogo	10
2.3.2	Conejos y Lobos	11
2.3.3	Populus	12
3	Diseño	15
3.1	Métricas	15
4	Desarrollo	17
4.1	Simulador	17
4.1.1	Strategies	17
4.1.2	Entities	18
4.1.3	Environment	18
4.1.4	Simulation	18
4.1.5	Monitoring	19
4.1.6	Excepciones	19
4.2	Algoritmo de Simulación	20
4.2.1	Bucle principal	20
4.2.2	Simulación por lotes y captura de excepciones	22
5	Resultados	25
5.1	Búsqueda de objetivos estáticos no perecederos en un tiempo dado	25
5.2	Búsqueda de un número dado de objetivos estáticos no perecederos	26
5.3	Búsqueda de un número dado de objetivos estáticos perecederos	29
5.4	Búsqueda de un número dado de objetivos aleatorios no perecederos	30
5.5	Búsqueda de un número dado de objetivos aleatorios perecederos	31
5.5.1	Cambio de la línea base de comparación	35

6 Conclusiones y trabajo futuro	37
6.1 Conclusiones	37
6.2 Trabajo futuro	38
Bibliografía	43
Definiciones	45
Acrónimos	47
Apéndices	49
A Fragmento de código ejemplo	51

LISTAS

Lista de códigos

4.1	Bucle principal de simulación	20
4.2	Método de actualización del entorno simulado	21
4.3	Método de actualización por época de los buscadores	21
A.1	Ejemplo del estilo del código desarrollado	51

Lista de figuras

2.1	Error inducido en el modelado de estrategias por la tasa de muestreo	7
2.2	Trayectorias ejemplo de un Lévy Walk, un movimiento Browniano y un movimiento balístico	8
2.3	Distribución de cola pesada (Cauchy) frente a distribución normal (Gaussiana)	9
2.4	Modelo de depredación lobo-oveja de Netlogo	11
2.5	Simulador “Rabbits and Wolves” (Resultados)	12
2.6	Simulador “Populus”	13
5.1	Episodios de búsqueda en 200.000 épocas (objetivos estáticos no perecederos)	27
5.2	Tiempo de búsqueda de 1.000 objetivos estáticos no perecederos	28
5.3	Tasa de éxito en búsquedas de 250 objetivos estáticos perecederos	29
5.4	Tiempo de búsqueda de 250 objetivos estáticos perecederos	32
5.5	Tiempo de búsqueda de 250 objetivos aleatorios no perecederos	33
5.6	Tasa de éxito y tiempo de búsqueda de 250 objetivos aleatorios perecederos	34
5.7	Tiempo de búsqueda y tasa de éxito para 250 objetivos aleatorios perecederos tras el cambio en la línea base de comparación	36

INTRODUCCIÓN

Cualquier animal que desarrolle su vida en la naturaleza está obligado a realizar búsquedas en su entorno. Esto se debe a la necesidad de dicho animal de asegurar su supervivencia y la de su especie a través de diferentes recursos, como pueden ser la comida o una pareja reproductiva. Como estos recursos se encuentran dispersos en el entorno que lo rodea, será entonces una tarea inevitable el ir en su busca. Es entonces cuando aparece el concepto de forrajeo. Los animales necesitan recorrer su entorno de una u otra manera para encontrar provisiones.

Sobre esta idea, la hipótesis del forrajeo a través de vuelos Lévy o *Lévy Flight Foraging Hypothesis* (LFFH) [1], se basa en la aceptación de los Lévy flights y Lévy walks como las estrategias que optimizan las búsquedas aleatorias para buscadores sin conocimiento sobre su entorno. Esta hipótesis sostiene una adaptación natural a través de la evolución, de la selección natural, del forrajeo hacia estrategias de tipo Lévy para cualquier buscador animal en la naturaleza [2]. Su formulación hace más de dos décadas [1] ha generado una gran discusión en el ámbito biológico que se extiende hasta nuestros días [3, 4].

A partir de su proposición, se han realizado numerosos estudios experimentales y teóricos sobre buscadores y cazadores animales y los patrones que siguen sus rutas durante los periodos de búsqueda de recursos. A pesar de esto, sus resultados son contradictorios, ya que podemos encontrar estudios que apoyan resultados opuestos. Así, estudios realizados sobre el albatros [1, 5], el mejillón [6], el mono araña [7], el chacal [8], diferentes depredadores marinos [9, 10], organismos microscópicos [11, 12] e incluso en el movimiento de los cazadores y recolectores de origen humano [13, 14], muestran evidencias de traslados de “cola pesada” (*heavy-tailed*), o lo que es lo mismo, de patrones Lévy, durante sus búsquedas. Sin embargo, también existen resultados que se abren a algunos de los ejemplos anteriores, como para el albatros [15], el mejillón [16, 17] o el mono araña [18].

El contraargumento principal para esta hipótesis (además de los estudios previamente mencionados), y una de las preguntas más debatidas en años recientes por la psicobiología, se encuentra en el origen de la aparición de estos patrones de búsqueda [4]. Frente a la hipótesis presentada, que asegura un origen innato y evolutivo para su aparición [2, 19], encontramos argumentaciones a favor de un origen emergente [6, 20], resultado de la interacción buscador-entorno.

Es entonces el objetivo de este trabajo estudiar a través de simulaciones computacionales las estrategias que muestren mayor eficiencia sobre diferentes **entornos** de búsqueda. De esta manera, obtendremos datos estadísticos sobre las diferencias que supone para un **buscador** un cambio en su manera de afrontar una búsqueda en el **entorno** que lo rodea y sobre si esto puede suponer una ventaja evolutiva sobre individuos que utilicen una estrategia diferente. El problema parece simple, buscar el camino más corto o de menor coste que separa al **buscador** del **objetivo**, optimizar una búsqueda dirigida. Sin embargo, en la mayoría de casos este **buscador** poseerá una pequeña cantidad o ninguna información acerca de su **entorno** y la posición de sus **objetivos**. Será necesario entonces recurrir a búsquedas no dirigidas o aleatorias. El problema reside ahora en la elección del proceso ,estocástico u ordenado, de mayor eficiencia para llevar a cabo estas búsquedas, la estrategia de búsqueda que mejor se adapte a las necesidades del **buscador** y su **entorno**. Es esta elección en la que nos centraremos durante la extensión de este trabajo.

1.1. Motivación

El tema de estudio de este trabajo es un claro ejemplo de cómo la informática puede resultar de gran utilidad para otras disciplinas de la ciencia como la potente herramienta que ésta conforma. Específicamente en nuestro campo de estudio, el de la **psicobiología**, que se ocupa del estudio del comportamiento de los animales, entre los cuales se incluye a la especie humana, la informática nos permite realizar estudios teóricos computacionales. Así, somos capaces de replicar tanto **entornos** biológicos como los protagonistas animales que se encuentran en éstos, aliviando el gran esfuerzo y coste que supondría un estudio realizado sobre sus análogos en la naturaleza. Otra ventaja de este tipo de estudios es que nos permiten simular el paso de grandes cantidades de tiempo.

De esta manera, una vez descubiertos patrones de tipo Lévy en **buscadores** naturales, somos capaces de simular estrategias de búsqueda que produzcan estos mismos patrones. Siendo así, las estrategias de búsqueda simuladas nos permitirán llegar a conclusiones estadísticas acerca de la eficiencia de sus búsquedas y si éstas podrían haber generado resultados mejores si estos mismos **buscadores** hubieran afrontado el problema de las búsquedas utilizando una estrategia diferente.

Podemos entonces realizar un estudio más a fondo sobre el impacto de las rutas trazadas sobre los **entornos** en que suelen encontrarse nuestros **buscadores**, así como someterlos a pruebas bajo **entornos** que no les son familiares, lo cual sería de la mayor complejidad llevar a cabo en la naturaleza o en un experimento tradicional.

La informática supone así la conciliación del conocimiento teórico, que nos permite abstraer las propiedades características de nuestros **buscadores** y sus rutas, con pruebas empíricas totalmente controladas y bajo los parámetros y **entornos** que sean de mayor utilidad para nuestro estudio.

1.2. Objetivos

Además del esfuerzo de investigación realizado para aportar evidencia sobre la eficiencia y posible carácter evolutivo de los **Lévy walks**, este trabajo se trata de un proyecto multidisciplinar en el que he colaborado con Gloria Fernández-Lázaro, del *Friends of Thoreau Program* del Instituto Franklin en la Universidad de Alcalá. Esto me ha permitido poner en práctica mis conocimientos acerca de las tecnologías de la información para aprender a través de la experiencia lo que significa trabajar en un proyecto junto a personas que no comparten mi formación académica.

Durante mi formación, lo habitual ha sido trabajar en un equipo formado completamente por informáticos para llevar a cabo un proyecto conjunto, donde todos los integrantes de éste poseen la misma visión y conocimientos similares. Sin embargo, me resulta especialmente interesante la oportunidad de trabajar y poner en común el progreso junto a alguien que no esté acostumbrado a la abstracción y el desarrollo de un sistema informático. De esta manera, trabajar con biólogos me permitió vivir en primera persona la comunicación necesaria para cooperar con alguien ajeno a nuestra disciplina, impulsándonos a ambos a avanzar hacia la propuesta objetivo de este trabajo de una manera más correcta y profesional, aportando los conocimientos que cada uno poseemos pero resultan totalmente distantes para el otro.

Además de esto, la parte más técnica del proyecto, el diseño e implementación de un simulador, me parece una oportunidad de lo más interesante para poner en práctica y ampliar mis conocimientos acerca de la programación y los patrones a los que estamos acostumbrados y en los que he sido formado durante estos años de grado. La naturaleza abstracta de los sistemas de información y la gran capacidad de computación de los sistemas actuales casan a la perfección con el objetivo de este trabajo, dándonos la posibilidad de simular el paso de grandes cantidades de tiempo sobre **buscadores** cuyo patrón de comportamiento bien podría tratarse del de un **buscador** natural animal.

El simulador será desarrollado en el lenguaje de programación Python, en su versión 3.8. Éste se trata de uno de los lenguajes de programación de mayor importancia en la actualidad, puesto que se ha ganado gracias a todas las facilidades que ofrece a los programadores, así como la gran cantidad de bibliotecas que la comunidad informática ofrece a todo aquel que necesite trabajar con información de distinta naturaleza y distintos grados de complejidad.

Por lo tanto, todo el esfuerzo volcado sobre la implementación del simulador será de gran utilidad personal, ya que éste me permitirá entender en un mayor grado la manera más correcta y habitual de trabajar en este lenguaje que tanto impacto puede tener en mi futuro profesional.

Por último, me resulta del mayor interés llevar a cabo el desarrollo de un framework potencialmente reutilizable. De esta manera, se cuidó minuciosamente el diseño y documentación del proyecto para asegurar que sus componentes presenten el menor nivel de acoplamiento posible, y que su funcionalidad pueda ser ampliada con facilidad. Por ejemplo, sería simple añadir una nueva estrategia de

búsqueda, al definir el framework una interfaz que deben cumplir todas las estrategias. Estoy convencido de que ésta es la manera correcta de desarrollar software de carácter científico, con la esperanza de que facilite el trabajo futuro en la materia de estudio.

1.3. Organización de la memoria

La memoria de este Trabajo Fin de Grado está dividida en cinco apartados que permitirán al lector familiarizarse con el problema en general, para después pasar a una descripción más técnica acerca de cómo se afrontó el problema, así como los resultados obtenidos a partir de la resolución de éste y la interpretación que podemos hacer sobre ellos.

De esta manera, en el **Capítulo 2** se comenzará describiendo el estado actual acerca de la LFFH. También se explican diferentes modelos presentes en la literatura que describen el movimiento animal, así como distintas herramientas ya existentes con el objetivo de realizar simulaciones computacionales de **entornos** naturales.

Después, se expone más a fondo el trabajo personal realizado para discutir acerca de la LFFH. Se hace énfasis en el origen evolutivo o emergente de los patrones Lévy sin olvidar nuestro objetivo principal, encontrar la estrategia que presente mayor eficiencia en las búsquedas aleatorias propuestas. Siendo así, como parte del **Capítulo 3**, primero se presentará el diseño del experimento a realizar, describiendo las métricas que serán recogidas durante la ejecución de las simulaciones, para más tarde describir las características funcionales del simulador a desarrollar.

Se realizará en un estudio de la parte más técnica del simulador en el **Capítulo 4**. En éste, se describe la arquitectura del simulador desarrollado, así como las decisiones de diseño e implementación que lo originaron. También se exponen los pilares principales de su funcionamiento.

A continuación, en el **Capítulo 5**, se exponen los resultados obtenidos en las diferentes simulaciones que se realizaron con el objetivo de poder comparar la eficiencia de las distintas estrategias de búsqueda estudiadas.

En el **Capítulo 6** se comentan las conclusiones a las que nos permiten llegar los resultados obtenidos. Se ofrece una discusión acerca de lo que esto podría suponer en el mundo real, en la naturaleza, eliminando la abstracción propia de las ejecuciones y llevando el trabajo de vuelta a su origen más natural. También se identifican los posibles caminos a seguir para continuar aportando evidencias teóricas al campo de investigación estudiado.

ESTADO DEL ARTE

2.1. Lévy Flight Foraging Hypothesis

El debate acerca de los *Lévy walks* permanece activo desde la investigación llevada a cabo por Viswanathan *et al.* en 1999 [1]. Este artículo mostraba la supremacía de esta estrategia bajo las circunstancias de una búsqueda aleatoria. Las búsquedas aleatorias son una situación ineludible en la naturaleza, debido al desconocimiento de la ubicación de los *objetivos* en el entorno. Es por este motivo que la publicación levantó un gran interés dentro de la comunidad científica.

La *LFFH* se basa en estos hechos para sostener una tendencia evolutiva hacia estrategias que respondan a patrones de tipo Lévy. Se establecen entonces dos actitudes al respecto.

Los argumentos a favor de la *LFFH* [2, 3, 19] defienden la aparición de este tipo de estrategias y patrones de búsqueda como resultado del proceso evolutivo. Los *Lévy walks* se presentan como óptimos, y si las estrategias de búsquedas están sujetas a la selección natural, la evolución favorecerá a aquellos individuos que utilicen estrategias mejor adaptadas a las características de su entorno. Es entonces lógico tener en consideración esta posibilidad *evolucionista*.

Por otro lado, hay quienes mantienen una postura *emergentista* opuesta a esta visión [6, 20, 21]. Ellos atribuyen la aparición de patrones de *cola pesada* a la interacción de los *buscadores* con sus *entornos*. Son por tanto los *entornos* los que favorecen la aparición o no de *Lévy walks*. Si la evolución está involucrada es para evitar que esta estrategia desaparezca [20].

El motivo por el que resulta tremendamente difícil llegar a un acuerdo, es la diferencia que existe entre *patrón* y *proceso* [4]. A pesar de la posición que se quiera adoptar acerca del origen de los *Lévy walks*, existen estudios que afirman la aparición de *patrones* de este tipo en *buscadores* animales [1, 5–14]. Sin embargo, resulta mucho más complejo afirmar el motivo por el cual se movieron de esa forma. Detrás de la trayectoria o *patrón* que siguió el animal y que podemos observar, existe un motivo o *proceso* que lo generó y el cual desconocemos.

Se han formulado numerosos frameworks [22, 23] que buscan explicación a las decisiones que toman los animales a la hora de desplazarse por su *entorno*. Se debe tener en cuenta el entorno físico

que rodea al animal, el grupo junto al que se desplaza y se coordina (de existir), sus propias decisiones y motivaciones individuales y, en general, todo estímulo al que éste se exponga. Es por tanto necesaria información acerca del *proceso* de desplazamiento a la que nos es imposible acceder, o para la cual necesitaríamos recoger una cantidad de datos empíricos inviable.

2.2. Modelos de movimiento animal

Con el desarrollo de modelos de movimiento, buscamos *procesos* teóricos que ayuden a describir los diferentes tipos de movimiento que los animales siguen durante sus búsquedas. De esta manera, podemos generar *procesos*, estocásticos o deterministas, que produzcan *patrones* que compartan las mismas características fundamentales que aquellos observados empíricamente en la naturaleza. Es gracias a estos modelos como somos capaces de desarrollar hipótesis teóricas cuyos resultados pueden ayudarnos a explicar la naturaleza y su funcionamiento.

La manera en que se formulan estos modelos de movimiento animal es a través de estudios observacionales de animales en la naturaleza durante periodos de búsqueda de recursos. De esta manera, se registran las diferentes posiciones que ocupó el animal en el *entorno* y se estudian los modelos matemáticos que mejor se ajusten a los patrones de movimiento observados. El problema reside en la frecuencia con que se miden las posiciones del animal para describir la trayectoria que siguió. Entre una medida de posición y la siguiente, ha de transcurrir un periodo de tiempo no nulo. Esto significa que no tenemos acceso a la información completa de la trayectoria que el animal siguió, sino a una aproximación utilizando las posiciones registradas. Sólo sabemos con certeza la posición que el animal ocupó en el instante t y en el instante $t + 1$, pero no podemos asegurar la trayectoria que siguió entre estos dos puntos. La *Figura 2.1* pone en evidencia el problema que supone esta limitación.

También es importante a la hora de estudiar un modelo de movimiento el concepto de “paso”. Cuando hablamos de paso, hacemos referencia al desplazamiento que un *buscador* realiza sin cambiar de dirección. Es el conjunto de estos pasos o movimientos rectilíneos consecutivos el que conforma la trayectoria de los *buscadores* estudiados.

Este concepto está relacionado con la diferencia entre un *Lévy flight* y un *Lévy walk*. Ambas estrategias presentan una característica distribución de probabilidad de *cola pesada* en la logitud de sus pasos que les da nombre, una distribución de Lévy. Sin embargo, cada una posee características únicas que las separan. Principalmente, los *Lévy flights* hacen referencia a desplazamientos aéreos y por tanto en tres dimensiones espaciales. Por su lado, los *Lévy walks* describen desplazamientos terrestres, bidimensionales. Es más, habitualmente, los *Lévy flights* son utilizados en estudios teóricos e ignoran el comportamiento del *buscador* entre un paso y otro. En su análisis no se tiene en cuenta el recorrido del *buscador*, sino que se hace énfasis en la posición inicial y final de cada paso. También colisionan en este sentido con los *Lévy walks*, cuyo análisis detalla la trayectoria del animal. A pesar

de esto, se sigue haciendo la distinción entre un paso y otro de un **Lévy walk** utilizando la dirección de su movimiento para diferenciarlos.

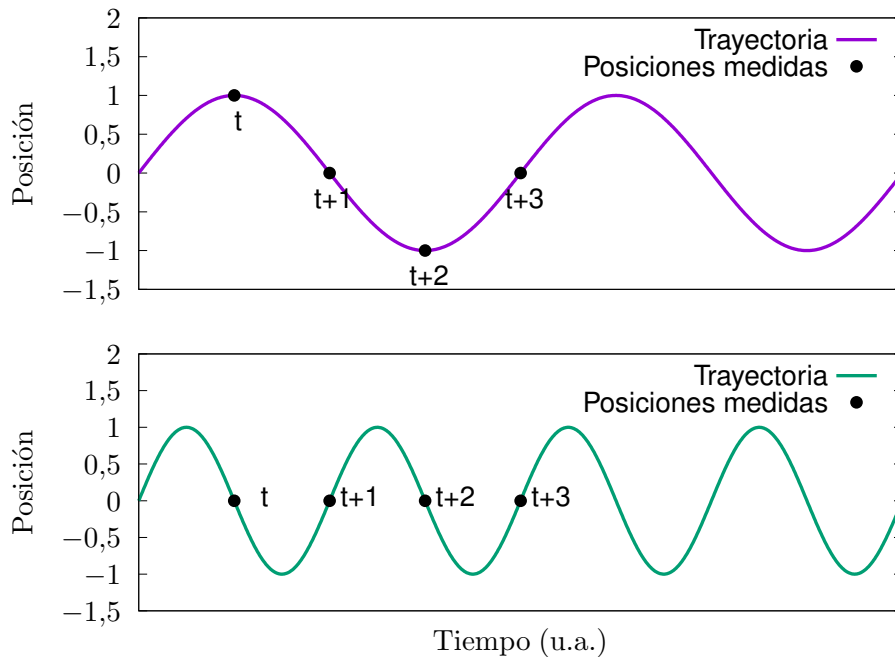


Figura 2.1: Esta figura ilustra la pérdida de información sobre la trayectoria si la frecuencia de muestreo no es adecuada. En el panel superior, las muestras describen de forma precisa el movimiento del animal. En el inferior, las muestras apuntan a que el animal permanece inmóvil mientras éste se desplaza más que el anterior. En ambas se representa un movimiento unidimensional cuya posición viene dada en el eje Y.

Un **buscador Lévy** (Figura 2.2, panel superior) alterna pasos cortos, cambiando constantemente su dirección, con pasos largos, desplazándose en línea recta en una misma dirección. Cuando sus pasos son cortos, el buscador centra su búsqueda en sus inmediaciones. Sin embargo, en lugar de continuar explorando su **entorno** de forma exhaustiva, un **buscador Lévy** decidirá eventualmente dar un “salto”, un paso considerablemente más largo. De esta forma, combina una búsqueda de ámbito local con exploración, formando parches de zonas exploradas a lo largo de su **entorno**.

Las estrategias Lévy no son las únicas que basan sus búsquedas en el balance entre una búsqueda local y la exploración. Encontramos uno de los límites de este balance en **buscadores** que priorizan la fase de búsqueda local (Figura 2.2, panel central), dejando la exploración en segundo plano. Decimos entonces que éste realiza un **movimiento Browniano**. Este movimiento está caracterizado por pasos cortos, que permiten al **buscador** descubrir sus inmediaciones a través de un **paseo aleatorio** que no introduce en su estrategia los largos pasos que caracterizan la fase de exploración de un **Lévy walk** o **Lévy flight**. Este cambio de estrategia produce una distribución normal o Gaussiana en la longitud de paso del **buscador**. La probabilidad de que se produzcan pasos anormalmente largos se reduce a cero, lo cual contrasta con las pesadas colas que manifiestan los **Lévy walks**, como se evidencia en la Figura 2.3.

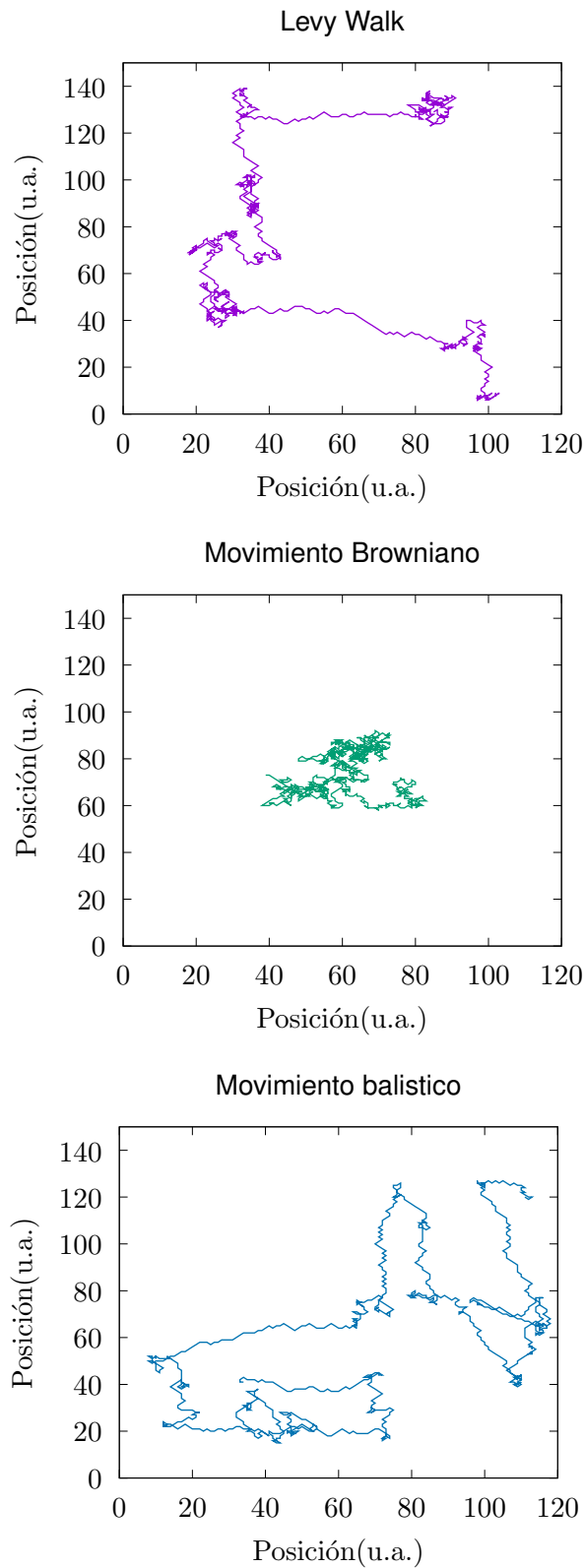


Figura 2.2: Trayectorias de ejemplo de un buscador trazando un Lévy walk (panel superior), un movimiento Browniano (panel central) y un movimiento balístico (panel inferior). Estas trayectorias se han generado utilizando el simulador desarrollado como resultado de este trabajo.

Si por el contrario encontramos un **buscador** cuya estrategia prioriza la exploración (Figura 2.2, panel inferior), sin eliminar por completo la fase de búsqueda local, decimos que éste describe un **movimiento balístico**. Se trata de una búsqueda que se centra menos en el ámbito local que un **Lévy flight** o **Lévy walk** y por tanto realiza pasos más largos. Junto al **movimiento Browniano**, define los límites de la estrategia Lévy. El **movimiento balístico** se caracteriza por su marcado desplazamiento en línea recta, mediante el cual un **buscador** realiza grandes desplazamientos a lo largo de su **entorno** omitiendo casi por completo la búsqueda local, de pasos cortos. La longitud de sus pasos no produce una distribución de probabilidad bien definida.

En esta misma línea de **paseos aleatorios**, encontramos a aquellos **buscadores** que llevan a cabo búsquedas utilizando estrategias de movimiento totalmente aleatorias (no confundir con el concepto de búsqueda aleatoria o ciega). Así, un **buscador** de este tipo, “**buscador aleatorio**” de aquí en adelante, deambulará por su **entorno** sin seguir ninguna estrategia específica, explorando de forma aleatoria sus alrededores hasta que encuentre algún **objetivo**. Los patrones que generan las rutas de estos **buscadores** aleatorios son inconsistentes.

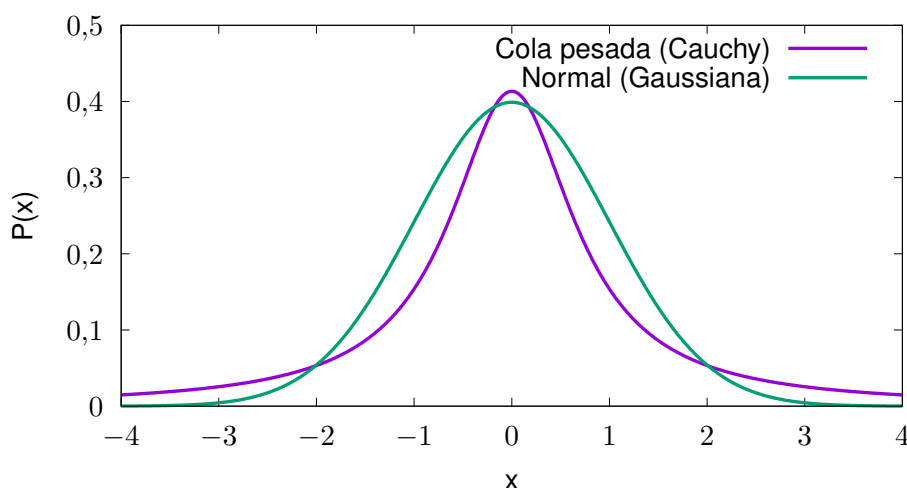


Figura 2.3: Distribución de **cola pesada** frente a la distribución normal o Gaussiana. Puesto que la distribución de Lévy es de una sola cola, se ha utilizado la distribución de Cauchy para una comparación más intuitiva.

Todos estos movimientos asumen que nuestro **buscador** no posee capacidad alguna de memoria. De esta manera, su movimiento debe describir un **paseo aleatorio**, para el cual sólo es relevante la posición ocupada en el instante actual para dar un paso hacia la posición siguiente. Para contrastar con esta idea, también incluimos **buscadores** aleatorios con memoria en nuestro estudio teórico. Éstos son capaces de recordar las últimas posiciones que ocuparon en el **entorno**. De esta manera, no volverán a visitar regiones en las que buscaron recientemente, evitando visitar espacios sin **objetivos** y favoreciendo la exploración de zonas no visitadas. Aún siendo así, sus búsquedas continúan siendo no dirigidas debido a la falta de conocimiento acerca de la posición de los recursos en su **entorno**.

Finalmente, también se valora durante el desarrollo de nuestras simulaciones la eficiencia de una estrategia exhaustiva. Esto es, un **buscador** que decida explorar paso a paso el **entorno** que lo rodea al completo de forma ordenada, sin visitar posiciones anteriores hasta haber visitado su **entorno** al completo. Este tipo de movimiento nos ayudará a entender si realmente vale la pena ejecutar una estrategia de movimiento compleja, o si lo más eficiente es en realidad una exploración secuencial del **entorno**.

2.3. Simuladores

El primer aspecto a destacar acerca de los diferentes frameworks de simulación utilizados habitualmente en estudios teóricos similares al nuestro es su propósito. A diferencia del desarrollado como objeto de este trabajo, la mayoría de estos frameworks se centra en modelos de interacción entre especies o individuos, principalmente relaciones depredador-presa. Esto es, se presentan como una herramienta de apoyo a la ecología, a la dinámica entre poblaciones que comparten un mismo **entorno**. Es en este aspecto en que difiere con el simulador desarrollado en nuestro caso, que tiene su foco puesto en la **psicobiología**, el estudio del comportamiento animal, siendo el protagonista principal la estrategia utilizada por el **buscador**.

Vamos entonces a analizar tres simuladores de modelos biológicos que congenian con el aspecto del forrajeo animal y las búsquedas aleatorias propio del estudio llevado a cabo en este trabajo.

2.3.1. Netlogo

Netlogo es un framework programable de modelado multi-agente desarrollado por Uri Wilenski [24]. Disponemos de un lenguaje de programación propio a éste que permite definir el comportamiento de las diferentes **entidades** de nuestro ecosistema. De la misma manera, podemos utilizar uno de los modelos de su numerosa librería para realizar simulaciones con parámetros configurables. Su uso es habitual en la comunidad biológica cuando se trata de realizar simulaciones como las propuestas, debido a su gran variedad de modelos ya disponibles así como el gran potencial que ofrece su personalización. En este apartado hablamos acerca de uno de sus modelos, el “Modelo de depredación lobo-oveja” (*Wolf Sheep Predation Model*) como ejemplo de lo que Netlogo nos ofrece.

Este modelo, bajo Netlogo y a través de una interfaz gráfica sencilla, nos permite realizar pruebas sobre **entornos** con distintos parámetros. Asimismo, también podemos definir algunos parámetros que modulan el comportamiento de los **cazadores** y las **presas**, los lobos y las ovejas, respectivamente. De esta manera, podemos comprobar si el ecosistema planteado se presenta como sostenible, o lo que es lo mismo, si ninguna de las poblaciones se extingue como resultado de su interacción.

En la captura de la **Figura 2.4** podemos ver los parámetros configurables del modelo tanto para el

entorno como para sus protagonistas en la parte superior izquierda de la interfaz del simulador. En la parte derecha se puede observar la representación gráfica en tiempo real del entorno. Por su lado, los resultados, también presentados en tiempo real, se reflejan en la serie temporal presente en la parte inferior izquierda, donde se presenta la dimensión de cada una de las poblaciones involucradas sobre el tiempo.

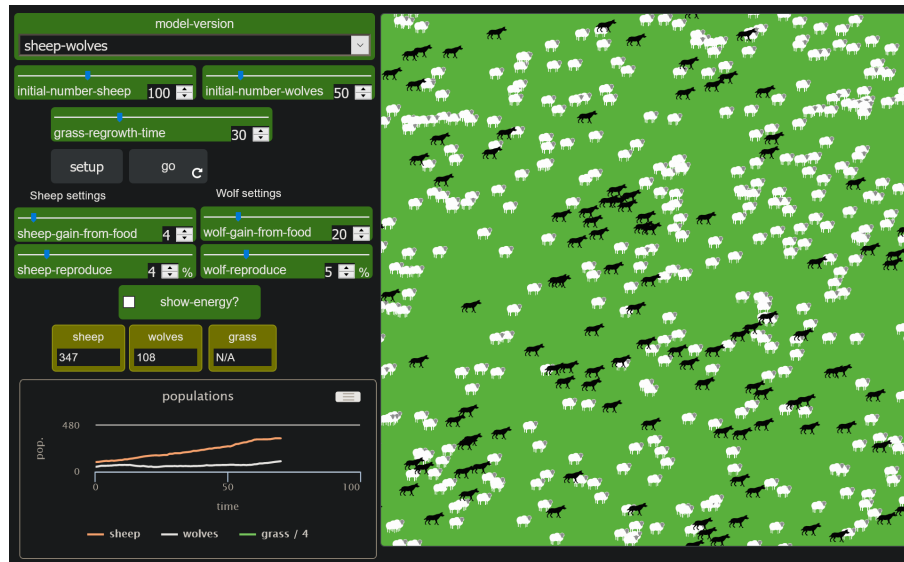


Figura 2.4: Captura de pantalla del modelo de depredación lobo-oveja de Netlogo.

El uso de Netlogo parece apropiado teniendo en cuenta la naturaleza del estudio. Sin embargo, tras valorarlo, se decidió optar por un desarrollo propio en lugar de configurar un modelo para este framework. A pesar del potencial de personalización de Netlogo, un desarrollo personal nos da acceso a un control total de lo que ocurre en nuestras simulaciones. No sólo tenemos control absoluto sobre el comportamiento de las entidades simuladas, sino que decidimos el comportamiento de todos los elementos involucrados. Así, podemos tomar decisiones acerca de qué se simula, cómo se simula, cuánta información se recoge y de qué manera. Esto da lugar a un mayor grado de optimización, versatilidad y aumenta la precisión de nuestra herramienta, el simulador.

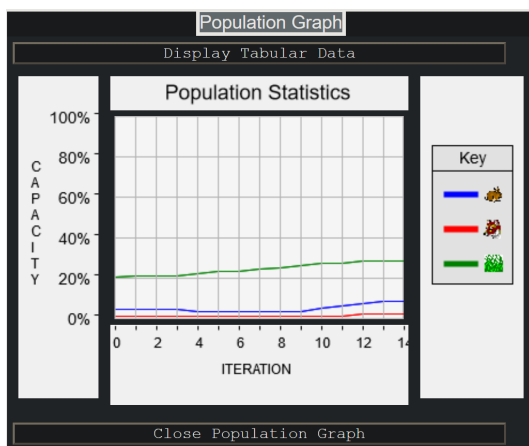
2.3.2. Conejos y Lobos

El simulador “Conejos y Lobos” (*Rabbits and Wolves*) fue desarrollado por *The Shodor Education Foundation* [25]. De manera similar al simulador que se comentó en el Apartado 2.3.1, se presenta al usuario a través de una interfaz gráfica. También su objetivo es similar al del mencionado, estudiar la estabilidad entre las diferentes poblaciones de un ecosistema. En este caso, sin embargo, el simulador sólo es capaz de simular un único modelo.

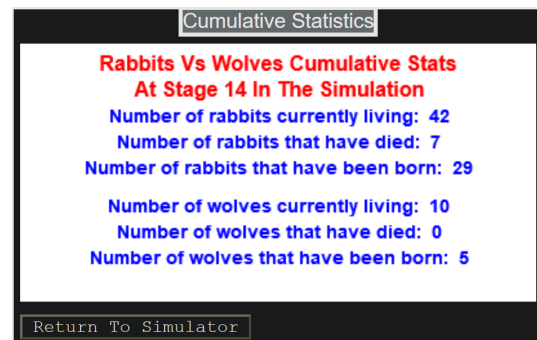
Al acceder al simulador encontramos representaciones gráficas del entorno simulado y sus casillas, así como de sus buscadores, los lobos, y sus objetivos, los conejos. Somos capaces de visualizar

también los parámetros configurables de la ejecución así como sus resultados en tiempo real a través de un menú *pop-up*, como se muestra en la Figura 2.5.

El principal problema que encontramos en este simulador es su bajo nivel de personalización. A pesar de permitirnos configurar algunos parámetros acerca del entorno y sus entidades, no somos capaces de cambiar el modelo simulado ni la estrategia de búsqueda utilizada por los buscadores. En nuestro caso, estamos interesados en analizar las diferencias entre distintas estrategias de búsqueda. Así, debemos ser capaces de diseñar nuestros propios modelos para éstas, además de tener control sobre el entorno en el que vive nuestro buscador. Necesitamos seleccionar las características y detalles de nuestras simulaciones y los datos que éstas generan de una manera que el simulador que acabamos de describir no nos permite.



(a) Gráfico de poblaciones



(b) Estadísticas acumuladas

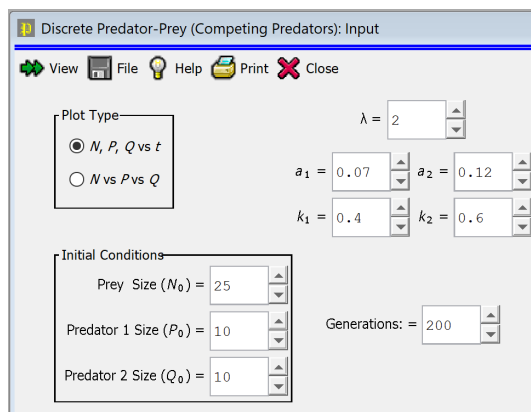
Figura 2.5: Capturas de pantalla del simulador “Conejos y Lobos”. A la izquierda, se muestra el gráfico de poblaciones. A la derecha, las estadísticas acumuladas de las poblaciones que están siendo simuladas.

2.3.3. Populus

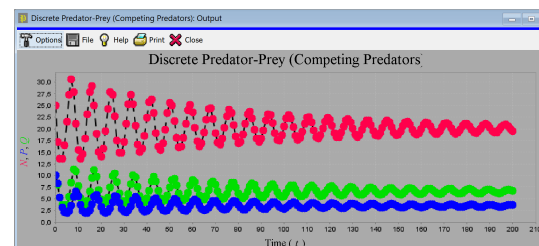
El *College of Biological Sciences* de la Universidad de Minnesota pone a nuestra disposición el último simulador que vamos a presentar en este apartado, “Populus” [26]. La diferencia principal con los dos simuladores anteriores es su presentación. Este programa no utiliza una representación gráfica en tiempo real de su entorno sino que, por el contrario, toma un enfoque centrado en los resultados. Esto es, sus modelos son teórico-matemáticos, sistemas de ecuaciones. Esto significa que sus simulaciones no son procedurales, sino que son deterministas. De esta forma, se nos ofrece una amplia gama de modelos matemáticos de ecosistemas biológicos, con la opción de modificar todos sus parámetros. Así, tras seleccionar el modelo que deseamos estudiar y ajustar sus detalles, se nos muestra una gráfica que refleja los resultados obtenidos, como podemos ver en la Figura 2.6.

A través de esta herramienta somos capaces de realizar simulaciones de diferentes modelos teóricos que se describen en la literatura. Por ejemplo, encontramos el modelo de Nicholson-Bailey [27], un modelo que asume búsqueda no aleatoria descrito por Robert M. May en 1978 [28] y un modelo en que **cazadores** deben competir entre sí por las **presas** desarrollado por Hassell en 1978 [29], entre muchos otros. Así, podemos observar de forma sencilla las consecuencias de cambiar el valor de sus parámetros. También permite la opción de cargar nuestros propios modelos para simularlos.

Este simulador presenta un enfoque alternativo al tomado en nuestro proyecto. En lugar de utilizar modelos matemáticos, en nuestro caso decidimos llevar a cabo simulaciones computacionales de caminantes aleatorios bidimensionales. Siendo así, damos vida en un **entorno** compartido a nuestro **buscador** y sus **objetivos** y simulamos el paso del tiempo en éste. De esta manera, observamos los cambios que producen las variaciones aleatorias en los procesos de las **entidades** de la simulación, en lugar de obtener aproximaciones matemáticas y deterministas de éstos. Somos capaces entonces de obtener un mayor nivel de detalle en la interacción **cazador-presa**, generando y almacenando información acerca de todos los estados en los que se encontró el **entorno** durante el proceso de simulación.



(a) Configuración de parámetros



(b) Gráfica resultado

Figura 2.6: Capturas de pantalla de “Populus” (2.3.3). En la Figura 2.6(a) podemos ver el proceso de configuración de parámetros del modelo seleccionado. En la Figura 2.6(b), la gráfica que se genera como resultado de su ejecución.

DISEÑO

El objetivo de esta investigación es encontrar qué estrategia de búsqueda se presenta como la más eficiente para un animal cuyo fin es obtener recursos necesarios para su supervivencia o reproducción de un **entorno** que desconoce completamente. Este objetivo se intentará alcanzar mediante un estudio teórico computacional basado en la simulación de distintas estrategias de búsqueda en **entornos** bidimensionales.

De una manera más específica, este trabajo se centra en la estrategia de forrajeo de los **Lévy walks** frente a otras estrategias como son la búsqueda exhaustiva, la búsqueda aleatoria, la búsqueda aleatoria con memoria, el **movimiento Browniano** y el **movimiento balístico**, descritas en el **Apartado 2.2**. Sobre estas estrategias, se realizarán pruebas en **entornos** de búsqueda con distintas características y con **objetivos** que varían en comportamiento, para determinar cuáles de éstas se adaptan mejor a cada uno de los distintos escenarios y cuantificar las diferencias presentes en su eficiencia y eficacia según diferentes métricas. Por ejemplo, se utilizarán tanto **objetivos estáticos** como dinámicos, así como **objetivos perecederos** y no **perecederos**.

3.1. Métricas

Será a través de las métricas recogidas como podremos verificar de forma teórica si los **Lévy walks** realmente presentan algún tipo de superioridad frente a otras estrategias cuando se trata de llevar a cabo búsquedas aleatorias. En caso afirmativo, es también nuestro objetivo descubrir las condiciones bajo las cuales esto es cierto. En cualquier caso, seremos capaces de observar qué estrategias producen mejores resultados según las características que presente el **entorno** que rodea al **buscador** animal que las utilice.

El simulador de búsquedas bioinspiradas recopilará toda aquella información relevante para la evaluación de la estrategia que el **buscador** esté utilizando. Se recogerán datos acerca de la trayectoria del **buscador**, la cantidad de energía o tiempo restante de vida sin consumir alimentos que posee en cada momento, y se llevará un registro de aquellas épocas en que la búsqueda ha resultado en un acierto, es decir, las épocas en las que el **buscador** ha encontrado un **objetivo**.

Asimismo, quedarán también registrados datos acerca de los **objetivos** de la búsqueda. Tenemos acceso a la trayectoria que siguió cada uno, así como a información acerca de los periodos de tiempo en que cada **objetivo** existió en el **entorno** junto con la causa de su desaparición, o lo que es lo mismo, si fue encontrado y consumido o se agotó su **tiempo de vida**. También conocemos el tiempo restante para que un recurso regenerable vuelva a poder ser consumido.

Adicionalmente, estas primeras métricas crudas, sin procesar, nos permitirán llegar a fuentes de información más complejas una vez sean trabajadas *a posteriori* tras terminar las simulaciones. De esta manera, podemos conocer el tiempo medio que un **buscador** necesita para encontrar un **objetivo** en el **entorno** cuando utiliza una estrategia de búsqueda específica, así como el porcentaje de éxito en búsquedas con **objetivos** volátiles. También nos permiten conocer los **episodios de búsqueda** completados por un **buscador** durante las simulaciones. Un **episodio de búsqueda** se define como el periodo de tiempo que transcurre desde que un **buscador** comienza una búsqueda hasta que encuentra un **objetivo**.

DESARROLLO

4.1. Simulador

El simulador con el que se ha llevado a cabo el estudio teórico desarrollado en este trabajo fue programado en el lenguaje Python, en su versión 3.8. Los siguientes apartados se identifican con los paquetes Python que componen el simulador. Su diseño busca ofrecer una herramienta software bien estructurada, reusable, ampliable y mantenible, gracias al uso de patrones de programación, buenas prácticas y una documentación descriptiva.

Todos estos paquetes y los ficheros que los componen se encuentran rigurosamente documentados con ayuda de la herramienta "pdoc" [30]. Todas las clases implementadas, así como sus métodos, han sido comentadas utilizando el estilo recomendado por Google [31]. En el [Apéndice A](#) encontramos un ejemplo del estilo que cumple todo el código desarrollado como parte del simulador.

4.1.1. Strategies

Pese a que todos los módulos son esenciales para el correcto funcionamiento del simulador, éste es quizá el principal y de mayor importancia. Esto es así puesto que es dentro de éste donde reside nuestro objeto de estudio principal, los modelos de movimiento animal. Asimismo, también incluye estrategias para el posicionamiento inicial (al comienzo de una simulación o cuando un **objetivo** se regenera tras ser consumido) de **entidades** en el **entorno**.

Siendo así, encontramos las implementaciones de los **Lévy walks**, el **movimiento Browniano**, el **movimiento balístico**, la búsqueda exhaustiva, la búsqueda aleatoria y la búsqueda aleatoria con memoria. Todas las estrategias mencionadas son configurables, siendo posible cambiar sus parámetros para adaptarlas a nuestras necesidades. Por ejemplo, podemos elegir la cantidad de memoria que posee nuestro buscador cuando lleva a cabo una búsqueda aleatoria con memoria. Tanto es así, que el **movimiento Browniano** y el **movimiento balístico** son en realidad **Lévy walks** configurados con los parámetros adecuados.

Para su desarrollo, se ha aplicado el patrón de programación de Estrategia (*Strategy* [32]). Se de-

fine una clase abstracta *Strategy* que posee un método abstracto *apply*. Este método abstracto será implementado por toda clase hija de *Strategy* como manera de definir la estrategia específica que representan. Así, encontramos dos implementaciones principales para este patrón en el simulador, *MovementStrategy* y *PlacementStrategy*, para las estrategias de movimiento y aquellas que se encargan de dar posición a toda nueva **entidad** que aparezca en el **entorno**, respectivamente. De esta manera, no sólo aseguramos transparencia para las **entidades** que quieran hacer uso de cualquiera de las estrategias (de movimiento o posicionamiento) implementadas, sino que además ofrecemos gran facilidad para añadir nuevas estrategias al simulador sin necesidad de modificar más código que la nueva estrategia a implementar.

4.1.2. Entities

El paquete descrito a continuación se encarga de administrar toda la información relevante a las **entidades** que viven en el **entorno** de nuestras simulaciones. De esta manera, sus clases describen a los **buscadores** y **objetivos**. No sólo continen en sus atributos los datos que los describen, sino que también los mantienen coherentes y en el estado correcto durante el proceso de simulación.

Existe también en este paquete una clase abstracta que ofrece una implementación base para una **entidad**. Ésta implementa funcionalidad básica como es la actualización de su posición cada época a través de una de las estrategias del paquete *strategies* (Apartado 4.1.1). Así, si se requiere la implementación de algún otro tipo de **entidad** más especializada, sólo es necesario definir una subclase con la funcionalidad y datos específicos a ese tipo de **entidad**, como se hizo con las clases *Seeker* (buscador) y *Target* (objetivo).

4.1.3. Environment

Este paquete y su clase principal, *Environment*, recoge los parámetros y características del **entorno** sobre el cual se realizan las simulaciones. Además, también posee una colección de todas las **entidades** presentes en cada simulación y se encarga de su coordinación e interacción. Siendo así, marca el comienzo y fin de cada época, además de informar a las **entidades** de posibles obstáculos o **entidades** colindantes.

4.1.4. Simulation

Las simulaciones podrían llevarse a cabo utilizando únicamente los paquetes descritos en los apartados anteriores y las clases que continen. Sin embargo, se incluye en el simulador este paquete junto con su clase principal *Simulator* a modo de contenedor que envuelve la funcionalidad de las simulaciones y simplifica su ejecución, a modo de *Facade* [32].

De esta manera, este paquete ofrece una interfaz configurable para realizar las simulaciones que necesitemos de una forma sencilla. Así, su clase principal, tras ser debidamente configurada, se encargará de realizar todas las inicializaciones pertinentes para comenzar el proceso de simulación. Además, también contiene los mecanismos necesarios para capturar excepciones, finalizar las simulaciones atendiendo a diferentes modos de simulación y realizar simulaciones por lotes. Las ejecuciones por lotes y la captura de excepciones se explican con mayor detalle en el [Apartado 4.2](#).

4.1.5. Monitoring

Este paquete, a pesar de no contener funcionalidad necesaria para la ejecución de las simulaciones, sí se presenta como necesario para comparar la eficiencia de diferentes estrategias de búsqueda. Las clases pertenecientes al paquete *monitoring* son las encargadas de “observar” a las *entidades* simuladas, recogiendo datos acerca de su estado y volcándolos en diferentes ficheros.

De esta manera, se hace uso del patrón *Observer* [32]. Existe una clase abstracta *Observer* que implementa a un observador con un sujeto genérico que será extendida por subclases especializadas en observar y documentar cambios en los datos de diferentes componentes del simulador. Así, se facilita la implementación de nuevos observadores y se asegura una interfaz de observación genérica y fácil de integrar en las clases sujeto observadas. Además, también se ha incluido una subclase abstracta de *Observer*, *EntityObserver*, que define la implementación de observadores de *entidades*. Ofreciendo este tipo de subclases abstractas intermedias, somos capaces de asegurar un resultado final ordenado y que sigue una lógica común aún cuando algún otro desarrollador extienda la funcionalidad de nuestro código.

Aparte de un código bien estructurado, para estudiar la eficiencia de las estrategias de búsqueda propuestas es muy importante la cantidad de datos disponibles por tratarse de procesos estocásticos. Siendo así, la eficiencia toma un primer plano en las decisiones de diseño de nuestro simulador. Para mejorar la eficiencia del simulador, este paquete hace uso de hilos de ejecución para escribir los datos registrados en ficheros. De esta manera, los observadores tienen a su disposición un búfer y un hilo. Estos observadores escribirán en el búfer los datos que desean volcar en fichero. Será el hilo el que, asíncronamente, irá leyendo el búfer y lo vacíe, escribiendo la información solicitada en los diferentes ficheros. Así, logramos observar y documentar los diferentes estados del simulador y sus componentes manteniendo el gasto de recursos al mínimo.

4.1.6. Excepciones

Debemos imponer ciertas reglas lógicas a los *entornos* que simulamos. Estos *entornos* son representaciones teóricas de *entornos* naturales y deben, por tanto, adherirse a ciertas reglas comunes. Por ejemplo, resulta imposible que dos *entidades* ocupen exactamente el mismo lugar en el *entorno*. No

todos los posibles estados del **entorno** son aceptables y, haciendo uso de excepciones, podemos asegurar que nuestras simulaciones cumplen las características necesarias para llevar a cabo el estudio propuesto.

Para asegurar que todas las posibles excepciones definidas por los desarrolladores pueden ser controladas correctamente, se define una interfaz común a todas éstas. Esta interfaz se trata de una subclase de *Exception*, de nombre *UserError*, que contiene un mensaje que explica por qué se generó. De esta manera, todas las excepciones que desarrollemos serán subclases de *UserError*, asegurando que todas contienen un mensaje de la misma naturaleza. Este diseño aumenta notablemente la mantenibilidad del simulador desarrollado, poniendo a disposición del usuario una forma sencilla de definir nuevas excepciones y de asegurar que todas ellas poseen una estructura común.

4.2. Algoritmo de Simulación

4.2.1. Bucle principal

A pesar de poder actuar de forma independiente utilizando las diferentes estrategias de búsqueda para navegar por su **entorno**, las **entidades** necesitan actuar acompañadas. Ésta es la función que realiza el bucle principal de nuestras simulaciones, el cual se encarga de coordinar las actualizaciones del **entorno**, así como de comprobar si la simulación ha llegado a su fin. Los módulos diseñados nos permiten mantener un bucle principal sencillo, como se muestra en el **Código 4.1**. El carácter desacoplado de los diferentes componentes se ve reforzado si observamos la pila de ejecución completa involucrada en cada iteración de este bucle.

Código 4.1: Fragmento de la clase *Simulation* donde se muestra el bucle principal de las simulaciones. El método *update* de la clase *Environment*, en el que se actualiza el estado de las **entidades** simuladas se muestra en el **Código 4.2**.

```

1  def __simulation_loop(self):_
2      """Main_simulation_loop."""_
3
4      while(not self.end_simulation):_
5          self.environment.update()_
6          self.epoch_+=_1_
7          self.__check_sim_end_condition()_

```

El primer método a analizar es *Environment.update*. Éste es el encargado de actualizar el **entorno** simulado en cada época. De esta manera, como se muestra en el **Código 4.2**, informa a las **entidades** de la llegada de una nueva época. En línea con el bucle principal, este método tampoco posee carga algorítmica alguna, sino que sirve para difundir la llegada de una nueva época a las **entidades** protagonistas de nuestras simulaciones.

Código 4.2: Fragmento modificado de la clase *Environment*. Este método es el encargado de actualizar, durante cada época, el estado de las entidades simuladas. El código se ha modificado para eliminar algunas llamadas a funciones privadas para facilitar su comprensión. En el código original se realizan comprobaciones adicionales para asegurar que las entidades permanecen dentro de los límites del entorno.

```

1  def update(self):
2      """Updates the state of the Environment's Entities.
3
4      Targets are updated first so that the Seeker is able to scan
5      the Environment properly.
6      """
7
8      # Update Targets
9      for target in self.targets:
10         target.next_epoch()
11         target.notify()
12
13     # Update Seeker
14     seeker.scan_environment(self)
15     seeker.next_epoch()
16     seeker.notify()

```

Código 4.3: Método *Seeker.next_epoch*. Realiza la actualización pertinente al buscador o *Seeker* cada época de simulación transcurrida. Debe ser, por lo tanto, llamado tras cada época.

```

1  def next_epoch(self):
2      """To be called as a callback function each epoch passed.
3
4      Returns:
5      int tuple: Entity's new position in the Environment.
6      """
7
8      if self.energy != constant.UNLIMITED:
9          self.energy -= 1
10
11     if not self.consumed_target:
12         self.move()
13
14     return self.position

```

El siguiente paso en la pila de ejecución es el método que se encarga de actualizar el estado de las **entidades**. Se ofrece una implementación abstracta con nombre *Entity.next_epoch* que cada subclase de *Entity* debe definir acorde a sus atributos. Encontramos en el **Código 4.3** la implementación para un **buscador** (clase *Seeker*) a modo de ejemplo. Este método comienza a introducir lógica en el simulador en forma de una máquina de estados de las **entidades**.

El componente final de la pila de ejecución es el método *MovementStrategy.move*. Es aquí donde encontramos el núcleo de los procesos independientes que conforman las **entidades**. Las estrategias de búsqueda son el objeto principal del estudio realizado y las encargadas de mover a las **entidades** a lo largo de su **entorno**. Cada **entidad** se encarga de actualizar su posición llamando a este método de forma independiente y transparente para el resto del simulador. La llegada de la nueva época es notificada a la **entidad** y ésta devuelve su nueva posición en el **entorno**.

Es este comportamiento transparente y la independencia de las entidades lo que caracteriza al diseño realizado para nuestro simulador. De esta manera, la lógica que habilita al simulador recae principalmente en las **entidades** y sus estrategias.

4.2.2. Simulación por lotes y captura de excepciones

La mayor parte de las estrategias de búsqueda bajo estudio, así como la posición inicial en el **entorno de buscador y objetivos**, son procesos estocásticos. Los experimentos ejecutados con nuestro simulador son procesos no deterministas. A pesar de utilizar la misma configuración de partida, obtendremos variaciones de diferente magnitud al llevarlos a cabo. Es por esto que deberemos realizar varias ejecuciones por cada escenario estudiado con fin de obtener estadísticas más precisas. Para facilitar esta tarea, se ofrece una utilidad como parte del simulador que nos permite definir el número de ejecuciones a realizar bajo una misma configuración.

Sin embargo, esta utilidad no sólo nos permite realizar ejecuciones por lotes a modo de planificador de tareas. Nuestro propósito es simular millones de épocas, cuya ejecución puede prolongarse durante varios días dependiendo de la potencia de computación que tengamos a nuestra disposición. Por este motivo, es importante automatizar el proceso de monitorización, para lo cual se ofrece un mecanismo de captura de excepciones además de recolectar metainformación acerca de cada una de las simulaciones ejecutadas. Este mecanismo se ilustra en el **Algoritmo 4.1**.

De esta manera, es posible que las ejecuciones permanezcan desatendidas hasta que se complete el lote de simulaciones. Una vez finalizadas, obtendremos archivos con metainformación junto a los datos resultado, tanto acerca del lote como acerca de cada una de las simulaciones individuales. Podremos de esta manera comprobar el porcentaje de simulaciones del lote que se han ejecutado correctamente, así como el número de simulaciones totales ejecutadas. Se refleja también la configuración común a todas las simulaciones como son la estrategia de búsqueda utilizada, el tamaño de

entorno y el modo de simulación. También se registra el número de épocas y el número de **objetivos** encontrados en cada una de las simulaciones del lote.

Ya que esta información se almacena siempre con el mismo formato, es posible automatizar su lectura y análisis. Así, podremos repetir las simulaciones que hayan terminado a causa de algún error imprevisto, además de comparar el rendimiento de las distintas estrategias de búsqueda utilizando el número de épocas simuladas o el número de **objetivos** encontrados.

```
1 Input: Tamaño del lote batch_size  
2 for iteration  $\leftarrow$  0 to batch_size do  
3   try :  
4     Inicializar simulación;  
5     Notificar comienzo a Observadores;  
6     Bucle principal de simulación;  
7   catch e  $\leftarrow$  Exception :  
8     error_string  $\leftarrow$  "e.name: e.message";  
9     simulation_status  $\leftarrow$  ERROR;  
10  finally :  
11    Notificar fin a Observadores;  
12  end  
end
```

Algoritmo 4.1: Este pseudocódigo muestra el algoritmo implementado para asegurar una ejecución por lotes exitosa utilizando capturas de excepciones y recolección de metainformación.

RESULTADOS

5.1. Búsqueda de objetivos estáticos no perecederos en un tiempo dado

En estas primeras simulaciones realizamos una comparativa de la eficiencia de las estrategias en estudio a través de los **episodios de búsqueda** que cada una es capaz de completar en 200.000 épocas de simulación. Cada simulación consiste en un único **buscador** que utilizará la misma estrategia de búsqueda hasta el fin de las 200.000 épocas.

Durante este y todos los demás experimentos, comprobamos la eficiencia de las estrategias ante **entornos** cuadrangulares, repitiendo cada experimento sobre **entornos** de distintas dimensiones. Siendo así, en las gráficas con los resultados que se muestran a continuación se utiliza el lado del cuadrado que conforma el **entorno** para definir el tamaño de éste. Con objetivo de acentuar el cambio que supone la variación del tamaño del **entorno** en los resultados, se han utilizado **entornos** no toroidales. Los **buscadores** no podrán superar los límites impuestos por el tamaño de su **entorno**. Por el contrario, en un **entorno** toroidal, un **buscador** podría navegar desde cualquier lado de su **entorno** al lado opuesto a éste. Este primer grupo de simulaciones nos permitirá hacernos una idea del comportamiento general de cada estrategia en función del tamaño del **entorno**, fijando una línea base para la comparación, ya que es de esperar que la eficiencia en comparación con otras estrategias varíe bajo **entornos** de distinto tamaño.

Durante este experimento, sólo existirá un **objetivo** disponible al mismo tiempo en el **entorno**. Se trata de un **objetivo estático** con **tiempo de vida** ilimitado. Una vez éste sea encontrado por el **buscador**, aparecerá instantáneamente un nuevo **objetivo** de las mismas características en una posición aleatoria del **entorno**.

En las gráficas que se muestran a continuación encontramos los resultados medios tras repetir el experimento cinco veces por estrategia y tamaño de **entorno**. La **Figura 5.1** muestra el rápido descenso en el número de **episodios de búsqueda** completados en relación al aumento del tamaño del **entorno** para todas las estrategias. La restrictiva limitación impuesta sobre el tiempo de simulación afecta a

todas las estrategias de búsqueda en un grado similar.

Los resultados obtenidos no se alejan de lo esperado. Al comienzo, para entornos pequeños, son la estrategia exhaustiva, la aleatoria, la aleatoria con memoria y el movimiento Browniano las que encuentran un mayor número de objetivos. Éstas son estrategias que se centran en el ámbito local. La eficiencia relativa de los Lévy walks y el movimiento balístico aumenta con el tamaño de entorno. La estrategia exhaustiva se mantiene como una de las más eficientes a lo largo de todo el experimento. Para entornos de lado mayor a 250, la limitación impuesta sobre las épocas de simulación es demasiado estricta para realizar una comparación ecuánime.

5.2. Búsqueda de un número dado de objetivos estáticos no perecederos

En este segundo experimento, hacemos un cambio en la variable dependiente. En este caso, en lugar de fijar el tiempo de simulación, fijamos el número de episodios de búsqueda que los buscadores deberán completar. De esta manera, medimos el tiempo en épocas de simulación que tarda cada una de las estrategias en completar 1.000 episodios de búsqueda. Con este cambio en la estrategia de simulación nos aseguramos de disponer de suficientes datos para realizar una mejor comparación a través de todos los tamaños de entorno simulados.

De forma similar al experimento descrito en el Apartado 5.1, repetimos este experimento para entornos de distintos tamaños, los cuales continúan siendo cuadrangulares y no toridales. Asimismo, también encontramos un único objetivo estático, que reaparece sin demora una vez consumido y cuyo tiempo de vida es ilimitado.

En este caso, se realizaron cuatro repeticiones de cada simulación por estrategia y por tamaño de entorno. Siendo así, observamos en la Figura 5.2 una clara división de las estrategias de búsqueda en dos grupos para la mayoría de tamaños de entorno simulados. En general, encontramos por un lado a la estrategia aleatoria, el movimiento Browniano y la estrategia aleatoria con memoria como las menos eficientes. Son el movimiento balístico, los Lévy walks y la estrategia exhaustiva las estrategias de búsqueda que menor crecimiento en su tiempo de búsqueda muestran en relación al crecimiento del entorno. Sin embargo, para los entornos de menor tamaño, los Lévy walks y sobre todo el movimiento balístico presentan una gran desventaja frente al resto de estrategias. En los entornos más pequeños, el movimiento balístico alcanza un tiempo de búsqueda de aproximadamente el doble que el resto de estrategias y los Lévy walks se igualan a las estrategias que finalmente presentan la menor eficiencia.

A pesar de que el movimiento balístico, los Lévy walks y la estrategia exhaustiva presentan un rendimiento similar, cabe destacar que la estrategia exhaustiva mantiene el mejor rendimiento dentro de las estrategias estudiadas independientemente del tamaño de entorno.

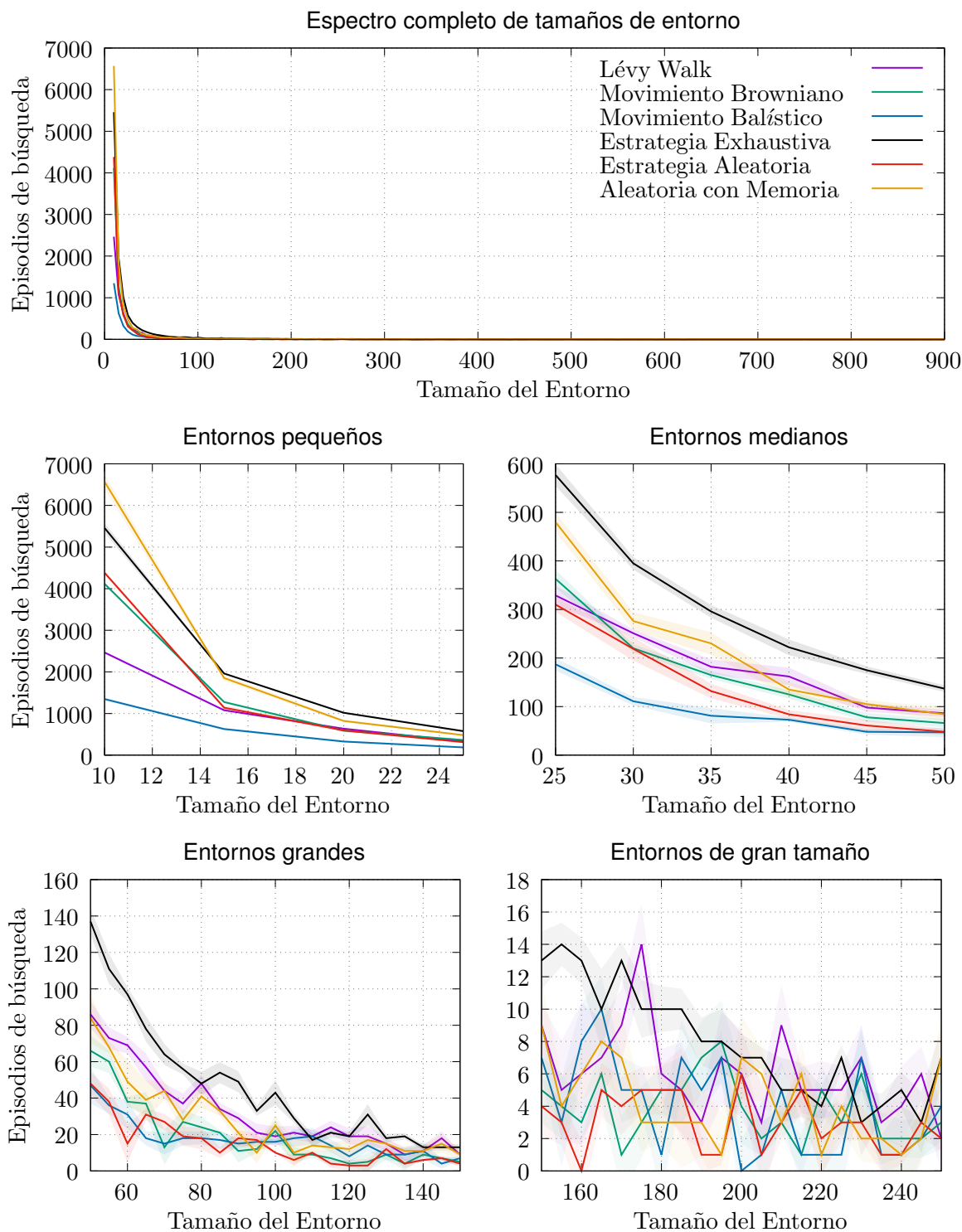


Figura 5.1: Número medio de episodios de búsqueda de todas las estrategias analizadas de cinco simulaciones con una duración fija de 200.000 épocas. El panel superior muestra el espectro completo de tamaños de entorno, mientras que los paneles inferiores muestran el detalle para determinados tamaños. El número de episodios de búsqueda de todas las estrategias estudiadas decae rápidamente cuando el tamaño de entorno comienza a aumentar.

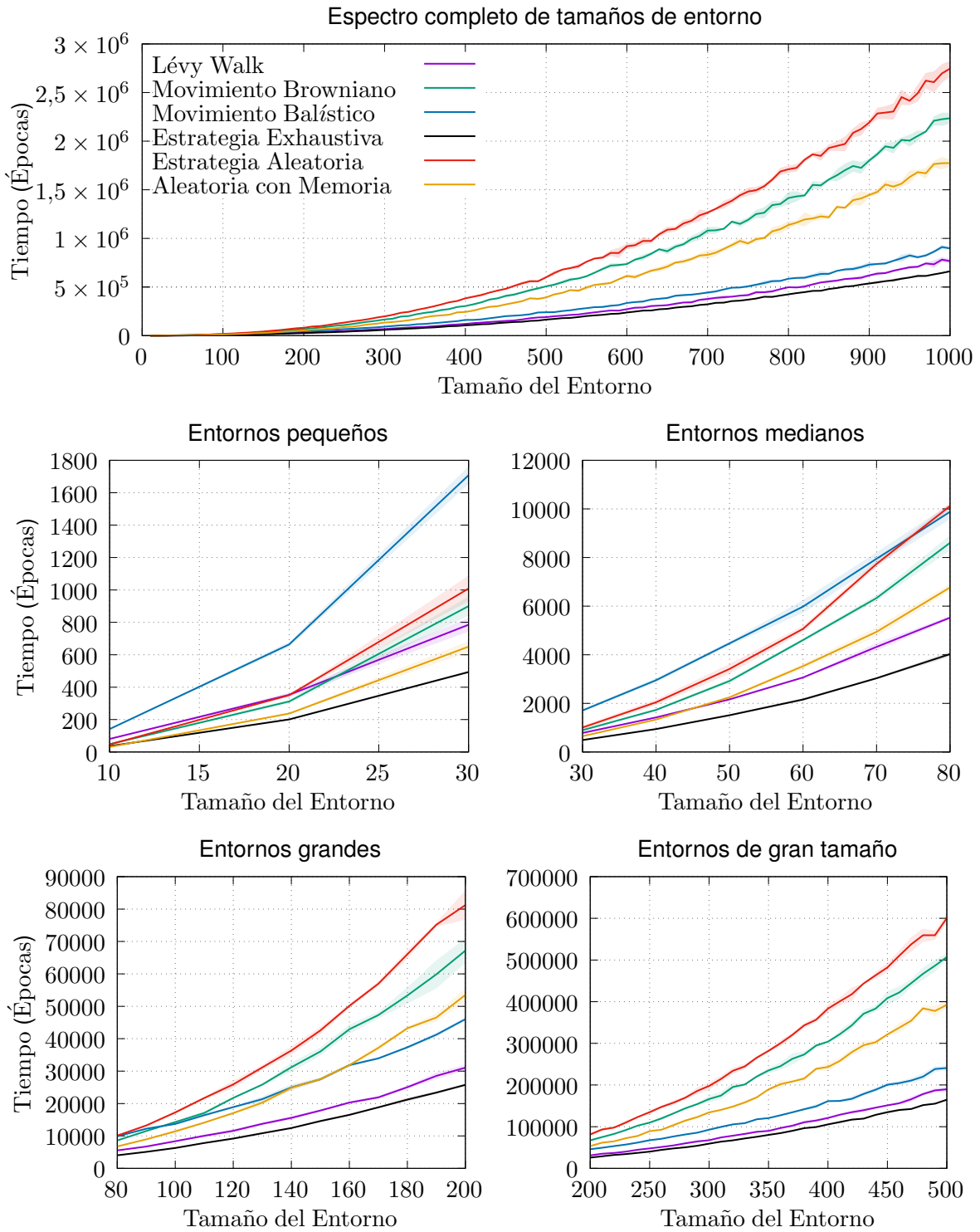


Figura 5.2: Tiempo medio de búsqueda de 1.000 objetivos estáticos y no perecederos de todas las estrategias analizadas. Se muestra el promedio de cuatro iteraciones y el tiempo de búsqueda se ha normalizado para mostrar el tiempo medio en encontrar un solo objetivo. Para la mayor parte de tamaños de entorno, observamos una división de las estrategias en dos grupos. Por un lado, tenemos a la estrategia aleatoria, el movimiento Browniano y la estrategia aleatoria con memoria. Por otro, mostrando mayor eficiencia, el movimiento balístico, los Lévy walks y la estrategia exhaustiva.

5.3. Búsqueda de un número dado de objetivos estáticos percederos

Durante los experimentos descritos en el [Apartado 5.1](#) y en el [Apartado 5.2](#), hemos trabajado con objetivos no percederos. Vamos ahora a introducir una limitación en el tiempo de vida de los objetivos de las búsqueda. De esta manera, vamos a repetir el experimento del [Apartado 5.2](#), esta vez con objetivos percederos. Así, mediremos el tiempo en épocas de simulación que tarda cada estrategia de búsqueda en completar 250 episodios de búsqueda, en entornos de tamaño variable con objetivos estáticos y percederos. Debido a que tratamos con objetivos percederos, aparece una nueva métrica de interés analítico, la tasa de éxito. Medimos, así, el número de objetivos encontrados sobre todos los que aparecieron en el entorno durante la simulación.

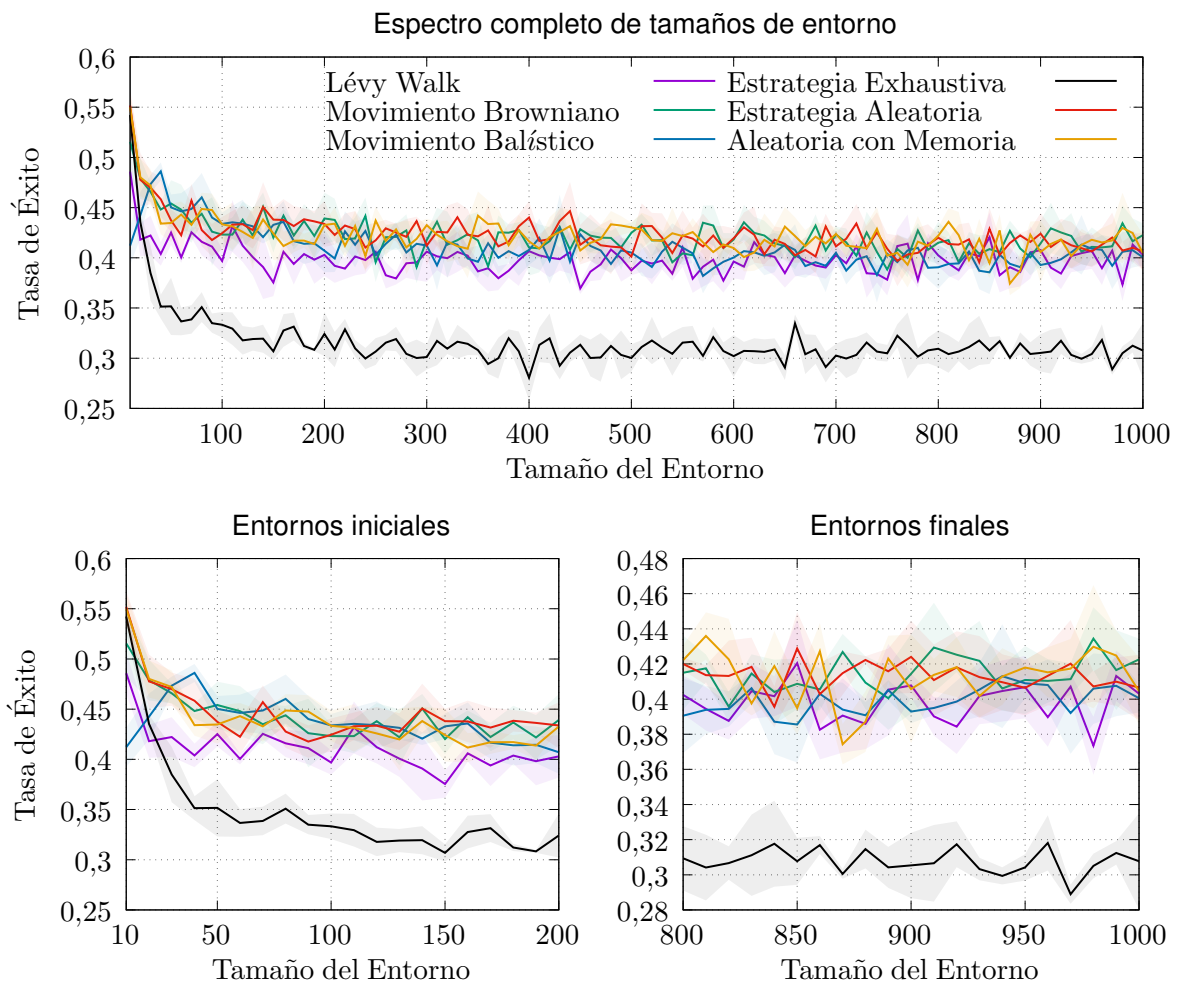


Figura 5.3: Tasa de éxito en la búsqueda de 250 objetivos estáticos percederos de todas las estrategias analizadas, promedio de cuatro simulaciones. La tasa de éxito es similar para todas las estrategias excepto para la exhaustiva, para la cual resulta aproximadamente un 10 % menor.

Para asegurar que la restricción impuesta al **tiempo de vida** de los **objetivos** es equivalente para todas las estrategias de búsqueda, el valor fijado en estas simulaciones se ha seleccionado relativo a cada estrategia y es diferente para cada tamaño de **entorno**. Así, el **tiempo de vida** será igual a la mitad de la línea base de comparación para cada estrategia y tamaño de entorno (**Apartado 5.2**). Esto contrasta con el experimento del **Apartado 5.5.1**, donde se fija un mismo **tiempo de vida** para todas las estrategias de búsqueda.

De nuevo, los **entornos** son cuadrangulares y no toroidales y sólo existe un **objetivo** a la vez en el **entorno**, que reaparecerá sin demora en una posición aleatoria de éste cuando sea consumido. También se realizaron cuatro repeticiones de este experimento, cuyo promedio compone los resultados mostrados en la **Figura 5.3** y la **Figura 5.4**.

Casi todas las estrategias presentan una tasa de éxito similar a excepción de la estrategia exhaustiva, como observamos en la **Figura 5.3**. La tasa de éxito de la estrategia exhaustiva se encuentra aproximadamente un 10 % por debajo de la tasa de éxito del resto de estrategias, las cuales difieren entre sí en un 3 % aproximadamente. Destacan los **Lévy walks** por encontrarse ligeramente. En la **Figura 5.4** encontramos el tiempo de búsqueda requerido por cada estrategia para completar los 250 **episodios de búsqueda**. De forma similar a lo observado en el caso de **objetivos no perecederos** (**Figura 5.2**), encontramos dos agrupaciones principales. La estrategia aleatoria, el **movimiento Browniano** y la estrategia aleatoria con memoria presentan los peores resultados. El **movimiento balístico** junto con la estrategia exhaustiva y los **Lévy walks** ven su tiempo de búsqueda mucho menos afectado por el aumento en el tamaño del **entorno** que las tres anteriores. Sin embargo, a diferencia de lo que ocurría en el caso anterior, los **Lévy walks** son la estrategia que mejores resultados muestra a largo plazo de todas las analizadas.

5.4. Búsqueda de un número dado de objetivos aleatorios no perecederos

Con este experimento, buscamos explorar las diferencias que supone en la búsqueda utilizar **objetivos móviles**, en contraste con los **objetivos estáticos** simulados hasta ahora en los experimentos de los **Apartados 5.1, 5.2 y 5.3**. Aunque el simulador está preparado para que los **objetivos** puedan moverse por el **entorno** siguiendo estrategias de movimiento complejas, nos centramos aquí en **objetivos** que se mueven siguiendo la estrategia más simple. Así, vamos a medir el tiempo de búsqueda necesario para que cada estrategia de búsqueda complete 250 **episodios de búsqueda** de **objetivos aleatorios**, en primera instancia no **perecederos**.

Los **entornos** utilizados siguen siendo cuadrangulares, no toroidales y de tamaño variable. En esta ocasión no se limita el **tiempo de vida** de los **objetivos**, ya que buscamos unas primeras métricas acerca de búsquedas con **objetivos aleatorios**. También vuelve a existir un máximo de un **objetivo** al mismo

tiempo en el **entorno**. Cuando éste sea consumido, se seleccionará una posición aleatoria del **entorno** para que aparezca otro **objetivo** con las mismas características.

De esta manera, se realizaron cuatro iteraciones del experimento, cuyos promedios resultado se muestran en la **Figura 5.5**. Los resultados obtenidos se asemejan a los estudiados en el **Apartado 5.2**, con **objetivos estáticos** no **perecederos**. En esta ocasión también encontramos dos grupos de estrategias según el tiempo de búsqueda en completar los 250 **episodios de búsqueda**, pero esta vez todas las estrategias necesitan un tiempo menor en general para completar el experimento. Así, se vuelven a agrupar la estrategia aleatoria, el **movimiento Browniano** y la estrategia aleatoria con memoria como las menos eficientes, presentando los mejores resultados el **movimiento balístico**, los **Lévy walks** y, por encima de todas ellas, la estrategia exhaustiva.

La estrategia exhaustiva vuelve a presentar los mejores resultados a lo largo de todo el espectro de tamaños de **entorno** simulados como ya ocurría en el experimento del **Apartado 5.2**.

5.5. Búsqueda de un número dado de objetivos aleatorios perecederos

De forma análoga a lo realizado en el **Apartado 5.3**, vamos a repetir el experimento descrito en el **Apartado 5.4** añadiendo esta vez una limitación al **tiempo de vida** de los **objetivos**. De esta manera, medimos el tiempo de búsqueda y la tasa de éxito de todas las estrategias de búsqueda bajo estudio para completar 250 **episodios de búsqueda**, con **objetivos aleatorios perecederos**.

Utilizamos **entornos** cuadrangulares, no toroidales y de tamaño variable. Continúa existiendo la limitación de un **objetivo** al mismo tiempo en el **entorno**, y no existe demora entre la consumición de un **objetivo** y la aparición de otro en una posición aleatoria de su **entorno**.

La elección del **tiempo de vida** de los **objetivos** se realizó de la misma manera que en el **Apartado 5.3**, utilizando la mitad del promedio de tiempo invertido en la búsqueda de un **objetivo**. Este **tiempo de vida** se calcula para cada estrategia por separado y por tamaño de **entorno**.

Los resultados promedio de cuatro iteraciones de este experimento se muestran en la **Figura 5.6**, la cual incluye tanto el tiempo de búsqueda como la tasa de éxito del total de los 250 **episodios de búsqueda**. Los resultados observados en el tiempo de vida se han simplificado por ser similares a los observados en el experimento del **Apartado 5.4**. Se mantienen las dos agrupaciones de estrategias y la estrategia exhaustiva sigue liderando la eficiencia a través de todos los **entornos** simulados. Respecto a la tasa de éxito, los resultados para la mayoría de estrategias son semejantes a los observados en el **Apartado 5.3**. Sin embargo, en el caso de la estrategia exhaustiva se observa una gran mejora. Su tasa de éxito ha mejorado en casi un 15% respecto a este experimento, con **objetivos estáticos perecederos**, y se sitúa ahora en primer lugar respecto a esta métrica.

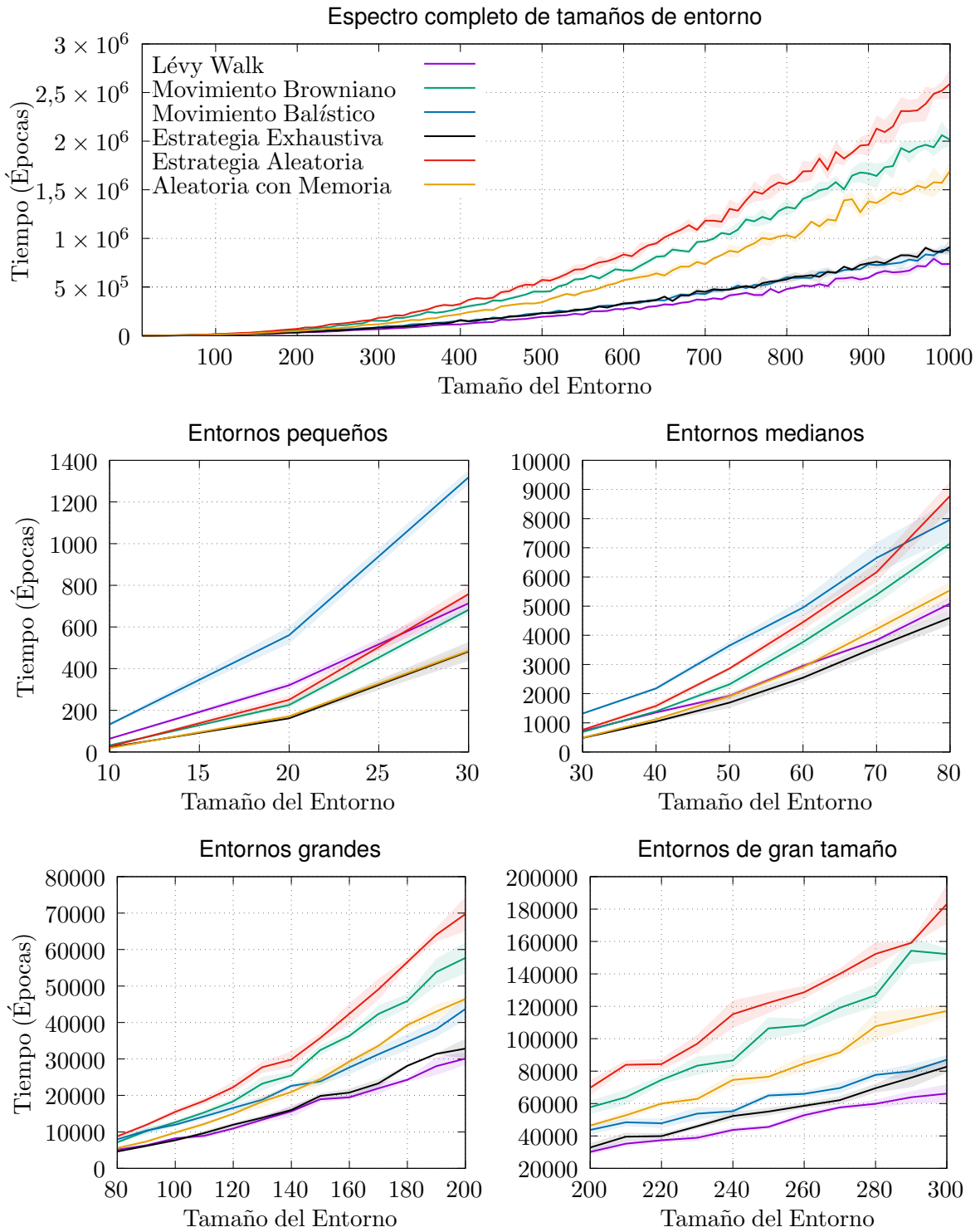


Figura 5.4: Tiempo medio de búsqueda de 250 objetivos estáticos y perezcos de todas las estrategias analizadas, promedio de cuatro iteraciones. Se muestra el tiempo medio en encontrar un solo objetivo. Los resultados muestran dos agrupaciones principales, con la estrategia aleatoria, el movimiento Browniano y la estrategia aleatoria con memoria con tiempos más altos y la estrategia exhaustiva, el movimiento balístico y los Lévy walks mostrando el mejor rendimiento. Los Lévy walks se presentan como la estrategia de búsqueda más eficiente.

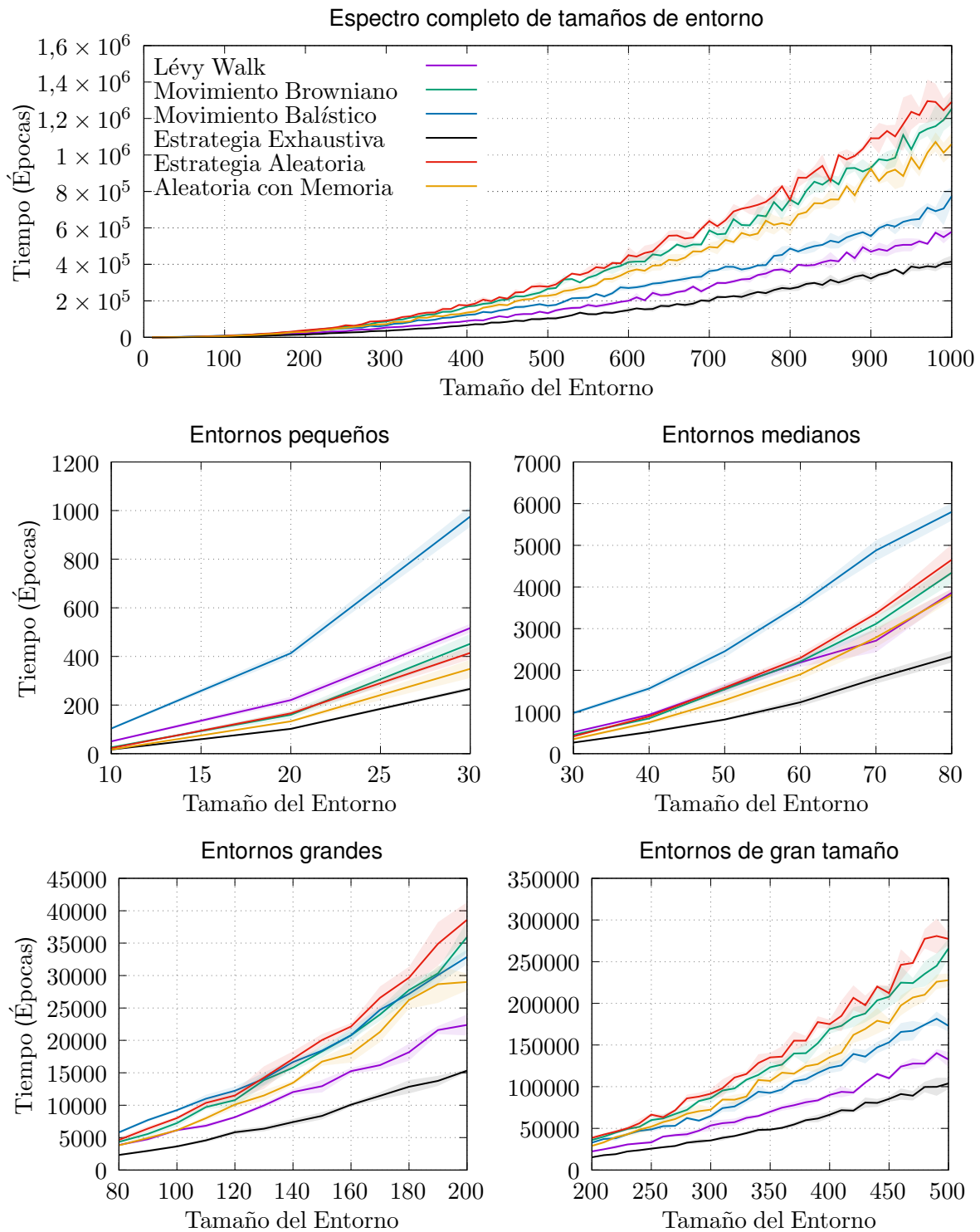


Figura 5.5: Tiempo medio de búsqueda de todas las estrategias analizadas de 250 objetivos aleatorios no percederos. Se muestra el promedio de cuatro iteraciones y el tiempo medio en encontrar un solo objetivo. El cambio hacia objetivos aleatorios produce simulaciones de menor duración, en las que domina la estrategia exhaustiva a través de todo el espectro de tamaños de entorno.

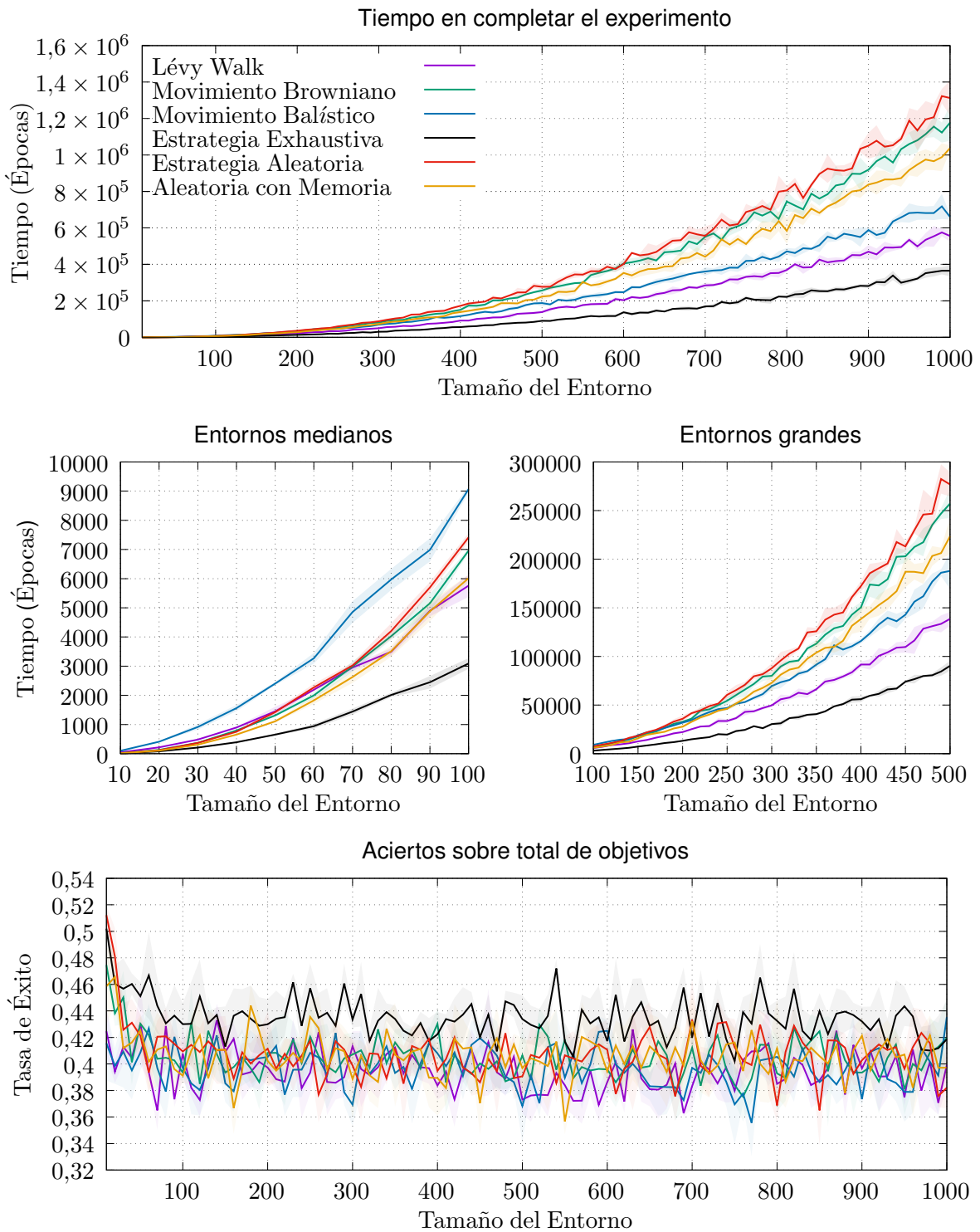


Figura 5.6: Tiempo medio de búsqueda y tasa de éxito de 250 episodios de búsqueda de objetivos aleatorios percederos. Se promedian los resultados de cuatro iteraciones y se muestra el tiempo medio de búsqueda de un solo objetivo. La estrategia exhaustiva presenta los mejores resultados para ambas métricas a lo largo de todo el espectro de tamaños de entorno.

5.5.1. Cambio de la línea base de comparación

En los experimentos realizados sobre **objetivos perecederos**, tanto el descrito en el **Apartado 5.3** como el descrito en el **Apartado 5.5**, el **tiempo de vida** de los **objetivos** era diferente para cada estrategia de búsqueda y cada tamaño de **entorno**. En este experimento cambiamos la línea base de comparación, escogiendo **tiempos de vida** dependientes únicamente del tamaño del **entorno**, utilizando el mismo para todas las estrategias de búsqueda. Así, repetimos el experimento descrito en el **Apartado 5.5** bajo restricciones en el **tiempo de vida** de los **objetivos** mucho más estrictas. Para llevarlo a cabo, tomaremos medidas de la tasa de éxito y el tiempo necesario para completar 250 **episodios de búsqueda** con **objetivos aleatorios perecederos**.

De esta manera, nos aseguramos de que no es el **tiempo de vida** calculado para la estrategia exhaustiva el que le está otorgando la ventaja de eficiencia que ésta muestra en los resultados. Escogemos, así, como **tiempo de vida** de los **objetivos**, un 25 % del tiempo medio de búsqueda de un **objetivo** para la estrategia exhaustiva en el experimento del **Apartado 5.4**, por ser la estrategia que mostró el menor tiempo de búsqueda durante dicho experimento.

Sigue existiendo un único **objetivo** al mismo tiempo en los **entornos**, los cuales serán de nuevo cuadrangulares, no toroidales y de tamaño variable.

La **Figura 5.7** muestra los resultados obtenidos tanto para el tiempo medio de búsqueda como para la tasa de éxito. Como esperábamos, las estrictas restricciones en el **tiempo de vida** han bajado las tasas de éxito hasta menos de la mitad de las observadas en el **Apartado 5.5**. La estrategia exhaustiva ha mostrado una bajada aproximada de un 10 %, pero aún así sus resultados se sitúan muy por encima del resto.

Los tiempos de búsqueda se mantienen parecidos respecto al experimento del **Apartado 5.5** para el **movimiento balístico**, los **Lévy walks** y la estrategia exhaustiva, pero se han reducido para la estrategia aleatoria con y sin memoria y el **movimiento Browniano**. Siendo así, la estrategia exhaustiva continúa presentando los mejores resultados de entre todas las estrategias de búsqueda a través de todos los tamaños de entorno.

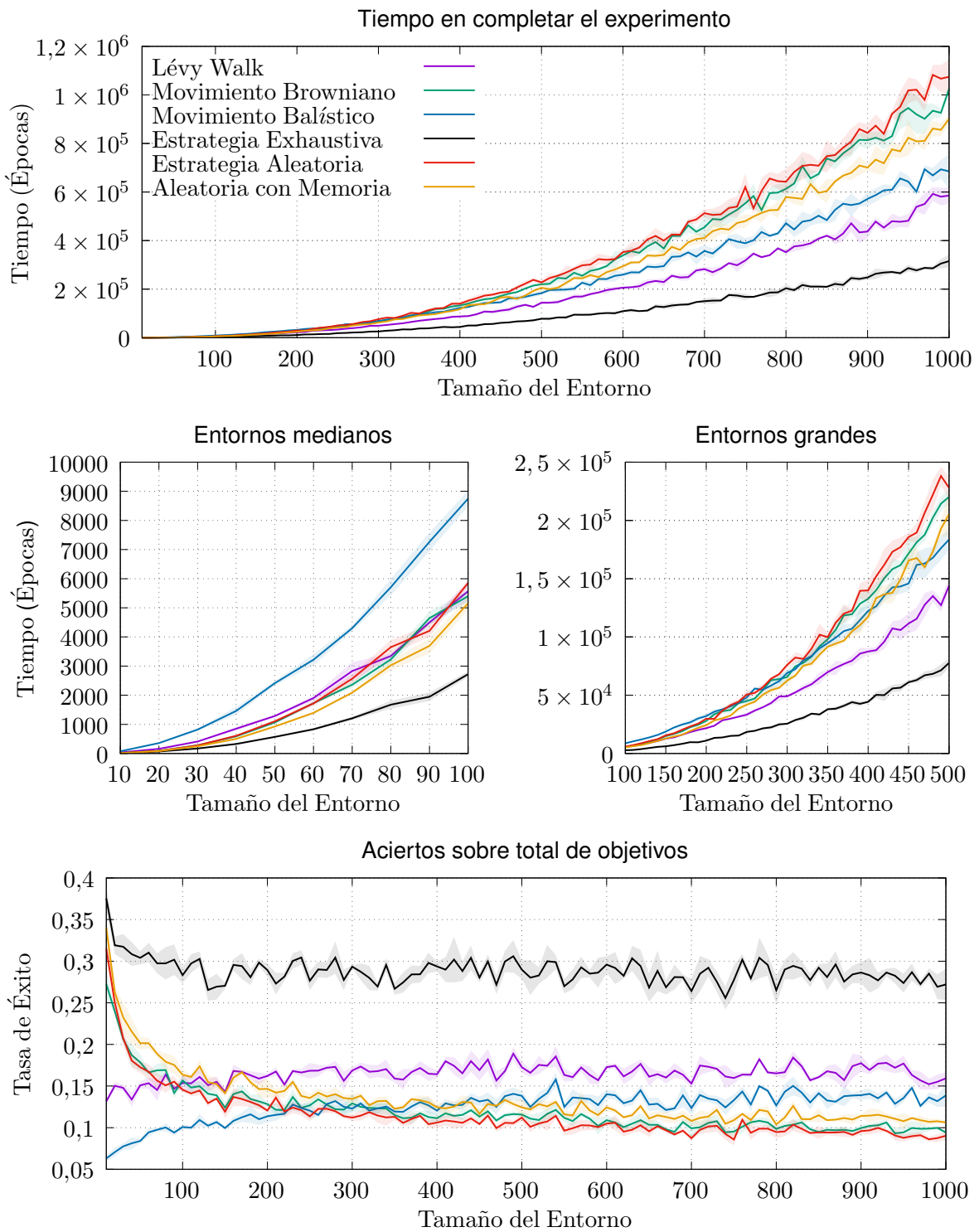


Figura 5.7: Tiempo medio de búsqueda y tasa de éxito de 250 episodios de búsqueda de objetivos aleatorios percederos tras el cambio en la línea base de comparación. Se muestra el tiempo medio de búsqueda de un solo objetivo. Todos los resultados son promediados de cuatro iteraciones del experimento. La estrategia exhaustiva muestra el mejor rendimiento para ambas métricas de entre todas las estrategias de búsqueda analizadas.

CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

En vista de los resultados obtenidos en el **Capítulo 5**, no podemos señalar una estrategia de búsqueda como la más eficiente ante cualquier situación, en contra de lo afirmado por Viswanathan *et al.* [1], que señalaba a las estrategias de tipo Lévy como las más eficientes para cualquier búsqueda aleatoria. La estrategia de búsqueda más eficiente depende tanto del **entorno** de búsqueda como de las características de sus **objetivos**. Así, diferentes estrategias de búsqueda muestran mayor eficiencia para aquellos **entornos** de menor tamaño, donde los **objetivos** aparecen próximos al **buscador**, que para aquellos de gran tamaño, donde los **objetivos** pueden aparecer a una gran distancia de éste.

Sobre estos primeros **entornos** de menor tamaño, como era de esperar, las estrategias de búsqueda que se centran en el ámbito local son las que dominan, independientemente del tipo de **objetivo** que se esté buscando. Éstas son el **movimiento Browniano**, la estrategia aleatoria y la estrategia aleatoria con memoria. La estrategia exhaustiva también muestra rendimientos muy buenos bajo estos **entornos**, ya que son suficientemente pequeños para que un barrido secuencial de su superficie se complete rápidamente.

Los resultados más interesantes son los obtenidos cuando el tamaño de los **entornos** simulados comienza a aumentar. Para estos **entornos** de mayor tamaño, la estrategia exhaustiva continúa mostrando uno de los mejores rendimientos. En los experimentos realizados, la estrategia exhaustiva es la más eficiente excepto cuando se buscan **objetivos estáticos** y **percederos**. Tanto los **Lévy walks** como el **movimiento balístico** muestran también unos resultados muy favorables, pero sólo para el caso mencionado, con **objetivos estáticos** y **percederos**, aparecen los **Lévy walks** como la estrategia de búsqueda de mayor rendimiento. No se ha realizado ningún experimento para el cual el **movimiento balístico** muestre mayor eficiencia en las búsquedas que los **Lévy walks**. Adicionalmente, el **movimiento balístico** rinde muy por debajo de los **Lévy walks** en **entornos** de tamaño reducido.

Otro caso interesante a destacar es el del **movimiento Browniano**. Ésta se trata de una de las estrategias de búsqueda más discutidas en la literatura relativa a la **psicobiología**, además de ser una de las más utilizadas en cualquier tipo de trabajo que requiera simular el movimiento de un animal por

su entorno. A pesar de esto, nuestras simulaciones no muestran ninguna evidencia a favor de este tipo de movimiento en el contexto de las búsquedas ciegas o aleatorias. Sus resultados han sido incluso mejorados por la implementación de una cantidad muy reducida de memoria sobre la estrategia de búsqueda más simple, la estrategia aleatoria, deambular por el entorno sin retroceder sobre nuestros pasos más inmediatos. Esto podría ser un indicador de que el movimiento Browniano no está tan extendido en el mundo animal como suponemos.

Según los resultados recogidos en los experimentos realizados, no parece existir un motivo de peso atado a la eficiencia en las búsquedas para realizar un esfuerzo evolutivo hacia la adopción de los Lévy walks. Es cierto, sin embargo, que sus resultados la presentan como la estrategia de búsqueda más versátil de las estudiadas. Ninguno de los experimentos que se han llevado a cabo ha supuesto una gran penalización para las búsquedas con Lévy walks, mientras esta estrategia mostraba uno de los mejores rendimientos. La estrategia exhaustiva ha mostrado los mejores resultados en general, pero el experimento con objetivos estáticos perecederos ha supuesto un obstáculo para su eficiencia. Esto supone que, en los casos en que los individuos realizan búsquedas a través de entornos de características variables, los Lévy walks probablemente sean la estrategia que, en conjunto, presenta las mayores ventajas.

A la vista de los resultados de la estrategia exhaustiva, sería necesario realizar una batería de experimentos más amplia, en los que se involucre el gasto y el aporte energético. En entornos naturales, resultaría muy costoso realizar una búsqueda secuencial como la descrita por la estrategia exhaustiva que utilizamos en nuestros experimentos, a pesar de que ésta haya mostrado el mejor rendimiento para la mayoría de las pruebas realizadas como parte de este trabajo.

6.2. Trabajo futuro

Debido a que se ha realizado un estudio teórico computacional, el primer aspecto en que se podría invertir una mayor cantidad de trabajo es en realizar más simulaciones. Primeramente, es posible realizar nuevas iteraciones de los experimentos propuestos, así como aumentar el rango de tamaños de entorno bajo estudio. Adicionalmente, se puede realizar un estudio más a fondo de las estrategias de búsqueda y su eficiencia a través de nuevos experimentos.

Con este fin, existen características implementadas en el simulador que no se han utilizado en los experimentos explorados en este trabajo debido al tiempo de simulación requerido para llevar a cabo un estudio exhaustivo. Siendo así, es posible realizar simulaciones en las que el buscador posee una cantidad de energía limitada para realizar las búsquedas, la cual puede ser recuperada por objetivos que le proporcionan un cierto aporte energético. Estas simulaciones nos permitirían encontrar la cantidad de tiempo que un buscador podría sobrevivir bajo ciertas condiciones hasta agotar toda su energía. Así, podríamos explorar el equilibrio entre la inversión en una búsqueda y el retorno obtenido.

Igualmente, los **buscadores** de nuestras simulaciones tienen una capacidad de detección de **objetivos** variable, parámetro que también podría ponerse bajo estudio. Ésta es nuestra forma de simular la capacidad de detección sensorial, a través, por ejemplo, de la vista o el olfato, de los individuos. Nuestro simulador también es capaz de simular **objetivos** que reaparecen con cierta demora tras ser consumidos. Esta funcionalidad permitiría simular algunos recursos como árboles frutales, los cuales pueden ser revisitados y consumidos pasado cierto tiempo. De esta manera, los **buscadores** podrían combinar una búsqueda ciega, como las realizadas en nuestros experimentos, con otras estrategias de búsqueda basadas en memoria a largo plazo, como ocurriría en la naturaleza para gran parte de los animales superiores. Finalmente, el simulador también permite simular la coexistencia de varios **objetivos**, tanto móviles como estáticos, en el **entorno** al mismo tiempo. Esto contrasta con los **entornos** en los que existía un único **objetivo** de nuestros experimentos.

Es más, como se explica en el **Capítulo 4**, el simulador fue diseñado con la idea de ser ampliado con facilidad. De esta manera, se podrían añadir nuevas estrategias de búsqueda al estudio, así como nuevos modos de simulación, nuevas condiciones de parada. También sería interesante la implementación de nuevas estrategias de posicionamiento de **entidades** en el **entorno**, así como la adición de nuevos tipos de **entidades** a nuestras simulaciones, como por ejemplo obstáculos que den lugar a **entornos** irregulares.

BIBLIOGRAFÍA

- [1] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. da Luz, E. P. Raposo, and H. E. Stanley, "Optimizing the success of random searches," *Nature*, vol. 401, pp. 911–914, Oct 1999.
- [2] M. E. Wosniack, M. C. Santos, E. P. Raposo, G. M. Viswanathan, and M. G. E. da Luz, "The evolutionary origins of lévy walk foraging," *PLOS Computational Biology*, vol. 13, pp. 1–31, Oct 2017.
- [3] G. Viswanathan, E. Raposo, and M. da Luz, "Lévy flights and superdiffusion in the context of biological encounters and random searches," *Physics of Life Reviews*, vol. 5, no. 3, pp. 133 – 150, 2008.
- [4] A. James, M. J. Plank, and A. M. Edwards, "Assessing lévy walks as models of animal foraging," *Journal of The Royal Society Interface*, vol. 8, no. 62, pp. 1233–1247, 2011.
- [5] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, "Lévy flight search patterns of wandering albatrosses," *Nature*, vol. 381, pp. 413–415, May 1996.
- [6] M. de Jager, F. J. Weissing, P. M. J. Herman, B. A. Nolet, and J. van de Koppel, "Lévy walks evolve through interaction between movement and environmental complexity," *Science*, vol. 332, no. 6037, pp. 1551–1553, 2011.
- [7] G. Ramos-Fernández, J. L. Mateos, O. Miramontes, G. Cocho, H. Larralde, and B. Ayala-Orozco, "Lévy walk patterns in the foraging movements of spider monkeys (*ateles geoffroyi*)," *Behavioral Ecology and Sociobiology*, vol. 55, pp. 223–230, Jan 2004.
- [8] R. Atkinson, C. Rhodes, D. Macdonald, and R. Anderson, "Scale-free dynamics in the movement patterns of jackals," *Oikos*, vol. 98, pp. 134 – 140, Jun 2002.
- [9] D. W. Sims, E. J. Southall, N. E. Humphries, G. C. Hays, C. J. A. Bradshaw, J. W. Pitchford, A. James, M. Z. Ahmed, A. S. Brierley, M. A. Hindell, D. Morritt, M. K. Musyl, D. Righton, E. L. C. Shepard, V. J. Wearmouth, R. P. Wilson, M. J. Witt, and J. D. Metcalfe, "Scaling laws of marine predator search behaviour," *Nature*, vol. 451, pp. 1098–1102, Feb 2008.
- [10] N. Humphries, N. Queiroz, J. Dyer, N. Pade, M. Musyl, K. Schaefer, D. Fuller, J. Brunnschweiler, T. Doyle, J. Houghton, G. Hays, C. Jones, L. Noble, V. Wearmouth, E. Southall, and D. Sims, "Environmental context explains lévy and brownian movement patterns of marine predators," *Nature*, vol. 465, pp. 1066–9, 06 2010.
- [11] E. Korobkova, T. Emonet, J. M. G. Vilar, T. S. Shimizu, and P. Cluzel, "From molecular noise to behavioural variability in a single bacterium," *Nature*, vol. 428, pp. 574–578, Apr 2004.
- [12] Y. Tu and G. Grinstein, "How white noise generates power-law switching in bacterial flagellar motors," *Physical review letters*, vol. 94, p. 208101, Jun 2005.
- [13] D. A. Raichlen, B. M. Wood, A. D. Gordon, A. Z. P. Mabulla, F. W. Marlowe, and H. Pontzer, "Evidence of lévy walk foraging patterns in human hunter–gatherers," *Proceedings of the National*

- Academy of Sciences*, vol. 111, no. 2, pp. 728–733, 2014.
- [14] M. C. González, C. A. Hidalgo, and A.-L. Barabási, “Understanding individual human mobility patterns,” *Nature*, vol. 453, pp. 779–782, Jun 2008.
- [15] A. Edwards, R. Phillips, N. Watkins, M. Freeman, E. Murphy, V. Afanasyev, S. Buldyrev, M. Luz, E. Raposo, H. Stanley, and G. Viswanathan, “Revisiting lévy flight search patterns of wandering albatrosses, bumblebees and deer,” *Nature*, vol. 449, pp. 1044–8, 11 2007.
- [16] A. Mashanova, T. H. Oliver, and V. A. A. Jansen, “Evidence for intermittency and a truncated power law from highly resolved aphid movement data,” *Journal of the Royal Society, Interface*, vol. 7, pp. 199–208, Jan 2010. 19474077[pmid].
- [17] V. A. A. Jansen, A. Mashanova, and S. Petrovskii, “Comment on “lévy walks evolve through interaction between movement and environmental complexity”,” *Science*, vol. 335, no. 6071, pp. 918–918, 2012.
- [18] D. Boyer, G. Ramos-Fernández, O. Miramontes, J. L. Mateos, G. Cocho, H. Larralde, H. Ramos, and F. Rojas, “Scale-free foraging by primates emerges from their interaction with a complex environment,” *Proceedings. Biological sciences*, vol. 273, pp. 1743–1750, Jul 2006. 16790406[pmid].
- [19] F. Bartumeus, “Lévy processes in animal movement: an evolutionary hypothesis,” *Fractals*, vol. 15, no. 02, pp. 151–162, 2007.
- [20] A. Reynolds, “Liberating lévy walk research from the shackles of optimal foraging,” *Physics of Life Reviews*, vol. 14, pp. 59 – 83, 2015.
- [21] P. Billard, A. K. Schnell, N. S. Clayton, and C. Jozet-Alves, “Cuttlefish show flexible and future-dependent foraging cognition,” *Biology Letters*, vol. 16, no. 2, p. 20190743, 2020.
- [22] Y. Prat and Y. Yovel, “Decision making in foraging bats,” *Current Opinion in Neurobiology*, vol. 60, pp. 169 – 175, 2020. *Neurobiology of Behavior*.
- [23] A. K. Shaw, “Causes and consequences of individual variation in animal movement,” *Movement Ecology*, vol. 8, p. 12, Feb 2020.
- [24] U. Wilensky, “Netlogo wolf sheep predation model.” <http://ccl.northwestern.edu/netlogo/models/WolfSheepPredation>, 1997.
- [25] T. S. E. Foundation, “Rabbits and wolves.” <http://www.shodor.org/interactivate/activities/RabbitsAndWolves/>, consultado en enero de 2021.
- [26] C. of Biological Sciences at University of Minnesota, “Populus.” <https://cbs.umn.edu/populus>, consultado en enero de 2021.
- [27] A. J. Nicholson and V. A. Bailey, “The balance of animal populations.—part i.,” *Proceedings of the Zoological Society of London*, vol. 105, no. 3, pp. 551–598, 1935.
- [28] R. M. May, “Host-parasitoid systems in patchy environments: A phenomenological model,” *Journal of Animal Ecology*, vol. 47, no. 3, pp. 833–844, 1978.
- [29] M. P. Hassell, “The dynamics of arthropod predator-prey systems,” *Monogr Popul Biol*, no. 13, pp. 1–237, 1978.
- [30] “pdoc - auto-generate api documentation for python projects.” <https://pdoc3.github.io/>

`pdoc/`, Consultado en enero de 2021.

- [31] Google, "Google python style guide. comments and docstrings." <https://google.github.io/styleguide/pyguide.html#38-comments-and-docstrings>, Consultado en marzo de 2021.
- [32] C. G. Lasater, *Design patterns*, ch. 3-4. 1st edition. ed., 2007.

DEFINICIONES

aleatorio Dicho de una **entidad** que se mueve aleatoriamente por su **entorno**.

buscador Protagonista de una búsqueda realizada sobre un **entorno** para encontrar **objetivos**. También “cazador”.

cazador Ver “buscador”.

cola pesada Distribución de probabilidad cuya varianza es no finita. También “*heavy-tailed*”.

entidad Durante las simulaciones, **buscador** u **objetivo** que existe en el **entorno** simulado.

entorno Conjunto de la situación que rodea a un buscador, y sus características, incluyendo las características de sus objetivos.

episodio de búsqueda Periodo de tiempo desde que un **buscador** comienza una búsqueda hasta que encuentra un **objetivo**.

estático Dicho de una **entidad** que se mantiene inmóvil.

Lévy flight Paseo aleatorio de **cola pesada** cuya longitud de paso viene dada por:

$$P(l) \approx l^{-\mu}, \quad 1 < \mu < 3.$$

Lévy walk **Lévy flight** cuya trayectoria se recorre de forma continua a lo largo de su entorno en lugar de mediante “saltos” entre un paso y otro.

movimiento balístico Paseo aleatorio de **cola pesada** cuya longitud de paso viene dada por:

$$P(l) \approx l^{-\mu}, \quad \mu = 1$$

Se trata de una búsqueda que se centra menos en el ámbito local que un **Lévy flight** o **Lévy walk** y por tanto realiza pasos más largos.

movimiento Browniano Paseo aleatorio de **cola pesada** cuya longitud de paso viene dada por: $P(l) \approx l^{-\mu}$, $\mu \geq 3$. Se trata de una búsqueda más local, con pasos más cortos, que un **Lévy flight** o **Lévy walk**.

objetivo Fin último de las búsquedas llevadas a cabo en un **entorno**. También “presa”.

paseo aleatorio Proceso aleatorio donde la posición de su protagonista depende únicamente de su posición anterior y una variable aleatoria que decide la dirección que tomará su próximo paso y la longitud de éste.

percedero Dicho de una **entidad** cuyo **tiempo de vida** es limitado.

presa Ver “objetivo”.

psicobiología Rama de la ciencia que enfoca sus estudios en la relación de los procesos biológicos con el comportamiento de los animales, entre los que se incluye al ser humano.

tiempo de vida Cantidad de tiempo, medida en épocas, en que una **entidad** dentro de las simulaciones dejará de existir por no poseer energía suficiente para poder seguir viviendo.

ACRÓNIMOS

LFFH *Lévy Flight Foraging Hypothesis.*

APÉNDICES

FRAGMENTO DE CÓDIGO EJEMPLO

Código A.1: Fragmento de la clase *Environment* que ejemplifica el estilo del código desarrollado.

```

1  #!/usr/bin/env python3
2  #-*-coding:utf-8-*-
3
4  """Environment.py: Environment in which different entities exist."""
5
6  __author__ = "Carlos Gijón"
7
8  from .Cell import Cell
9  from src.entities.Target import Target
10 from src.entities.Seeker import Seeker
11
12 class Environment():
13     """Generic environment with entities existing in it.
14
15     Args:
16     dimension_(int tuple): Dimension of the environment.
17     toroidal_(boolean): Whether the environment behaves as a toroid.
18
19     Attributes:
20     size_(int tuple): Dimension / size of the environment.
21     environment_(Cell matrix): Matrix of Cells. Uses "size" dimensions.
22     toroidal_(boolean): Whether the environment behaves as a toroid.
23     targets_(Target list): List of Targets in the Environment.
24     seeker_(Seeker): Simulation Seeker.
25     """
26
27     def __init__(self, dimension, toroidal):
28
29         (x, y) = dimension
30
31         assert x > 0, "Invalid number of rows_{0} <= 0".format(x)
32         assert y > 0, "Invalid number of columns_{0} <= 0".format(y)
33
34         self.size = dimension
35         # Initialize the Environment's Cell matrix.
36         self.environment = [[Cell((i, j)) for j in range(self.size[1])]
37                             for i in range(self.size[0])]

```


UAM

UNIVERSIDAD AUTONOMA
DE MADRID