



# UNIVERSIDAD DE GRANADA

GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FIN DE GRADO

**Análisis del consumo energético en algoritmos secuenciales y paralelos para la resolución de la envolvente convexa**

Presentado por:  
Dª. Rocío Alberca Bebia

Tutor:  
Dª. María Isabel García Arenas

2025 / 2026

ETS de Ingenierías Informática y de Telecomunicación

# Índice

<b>1</b>	<b>Introducción</b>	<b>1</b>
1.1	Contextualización de la problemática energética en computación . . . . .	1
1.2	Geometría computacional y el problema de la envolvente convexa . . . . .	2
1.2.1	Definición de geometría computacional . . . . .	2
1.2.2	Envolvente convexa y algoritmos de resolución . . . . .	2

# Capítulo 1

## Introducción

El avance en inteligencia artificial ha traído consigo un aumento significativo del consumo energético asociado a sus métodos computacionales. En este trabajo se analiza, en concreto, el impacto energético de los algoritmos evolutivos implementados en lenguajes de bajo nivel. Este capítulo introduce el contexto del problema, la relevancia de la eficiencia energética en inteligencia computacional y las motivaciones que justifican este estudio.

### 1.1 Contextualización de la problemática energética en computación

El avance continuo de la computación ha supuesto una mejora significativa en numerosos ámbitos de la sociedad, posibilitando el desarrollo de aplicaciones cada vez más complejas y con mayores exigencias de cálculo. Sin embargo, este progreso ha supuesto a su vez un incremento notable del consumo energético de los sistemas computacionales, especialmente en entornos de alto rendimiento, arquitecturas paralelas y aceleradores como las GPU, lo que conlleva un impacto medioambiental creciente. Estudios recientes han comprobado que un mayor rendimiento computacional no implica un menor consumo energético, ya que técnicas como la paralelización o el uso intensivo de hardware pueden reducir el tiempo de ejecución a costa de un aumento en la potencia consumida [1]. En consecuencia, la energía ha pasado a considerarse una métrica a evaluar tan relevante como el rendimiento o la eficiencia temporal. En este trabajo, concretamente se va analizar esta problemática en el ámbito de la geometría computacional, donde se tratará el problema clásico del cálculo de la envolvente convexa, ampliamente utilizado en aplicaciones prácticas, y que puede resolverse mediante diferentes algoritmos. El objetivo es comparar las características computacionales y energéticas de cada algoritmo entre ellos, de sus versiones secuenciales y paralelas.

En respuesta a esta creciente preocupación por el consumo energético, surge el concepto de **Green Computing**, que engloba aquellas prácticas de computación que integran consideraciones medioambientales en su desarrollo, con el objetivo de alcanzar un equi-

librio entre el rendimiento computacional y el coste energético. Desde este enfoque, no se persigue únicamente el avance tecnológico, sino que se prioriza la sostenibilidad a lo largo de todo el proceso de desarrollo del software y los sistemas computacionales [2]. Para perseguir esta meta, no se limita a estudiar diseño de infraestructuras o al hardware, sino que también se abarca el análisis y la evaluación de algoritmos desde el punto de vista energético.

Para avanzar hacia esta computación sostenible, resulta esencial identificar las métricas a evaluar. La eficiencia energética se ve influida por múltiples factores: desde la elección del lenguaje de programación hasta el compilador utilizado, el nivel de optimización aplicado y la arquitectura hardware en la que se ejecuta el código. De hecho, un estudio reciente ha evidenciado diferencias significativas en el consumo energético de un mismo algoritmo dependiendo del lenguaje, del conjunto de instrucciones empleado y del entorno de ejecución. Estos resultados justifican la necesidad de crear metodologías para medir y comparar dichas diferencias. [3]

## 1.2 Geometría computacional y el problema de la envolvente convexa

### 1.2.1 Definición de geometría computacional

La geometría computacional es una rama de la informática que aborda problemas definidos en términos de objetos geométricos (como puntos, líneas y polígonos) mediante algoritmos eficientes y estructuras de datos apropiadas. Se consolidó como disciplina el 1975, siendo introducida por M.I. Shamos en su tesis doctoral. Su objetivo es encontrar soluciones óptimas desde el punto de vista de la complejidad temporal y espacial, integrando problemas matemáticos con técnicas algorítmicas propias de la ciencia de la computación. Esta área estudia problemas como la convexidad, proximidad entre objetos geométricos, reconocimiento de patrones, planificación de movimientos, diseño de circuitos o síntesis de imágenes. Muchos de estos problemas constituyen conceptos básicos empleados en aplicaciones que requieren procesamiento geométrico intensivo como gráficos por ordenador, robótica o sistemas de información geográfica (SIG) [4, 5].

### 1.2.2 Envolvente convexa y algoritmos de resolución

Dentro de la geometría computacional, el problema de la envolvente convexa (convex hull) es uno de los más fundamentales y ampliamente estudiados. Dado un conjunto finito de puntos en el plano o en un espacio de dimensión superior, la envolvente convexa se define como el menor conjunto convexo que contiene a todos los puntos, o, de forma equivalente, como la intersección de todos los conjuntos convexos que los contienen [6].

Desde un punto de vista matemático, si  $S \subset \mathbb{R}^d$  es un conjunto de puntos, su envolvente convexa se define como el conjunto de todas las combinaciones convexas de puntos de  $S$ , es decir:

$$\text{conv}(S) = \left\{ \sum_{i=1}^n \lambda_i p_i \mid p_i \in S, \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1 \right\}.$$

Desde el punto de vista algorítmico, no hay una única forma de resolverlo; existen múltiples algoritmos con diferentes estrategias y características computacionales. Entre los más relevantes se encuentran:

Quickhull, basado en un enfoque de divide y vencerás, eficiente en la práctica para muchos conjuntos de datos [7].

Algoritmos de tipo divide y vencerás, como el de Preparata–Hong, con complejidad óptima en el caso general [8].

El algoritmo de Chan, que combina enfoques previos para alcanzar una complejidad óptima  $O(n \log h)$ , donde  $h$  es el número de puntos de la envolvente [9]

# Bibliografía

- [1] L. Lannelongue, J. Grealey y M. Inouye. “Green Algorithms: Quantifying the Carbon Footprint of Computation”. En: (2021). URL: <https://onlinelibrary.wiley.com/doi/10.1002/advs.202100707>.
- [2] You Zhou et al. “On the Opportunities of Green Computing: A Survey”. En: (2023). URL: <https://arxiv.org/pdf/2311.00447.pdf>.
- [3] Clemens Lutz, Georg Hager y Gerhard Wellein. “Energy Efficiency of Programming Languages: A Comparative Study”. En: *2021 International Conference on Energy-Aware High Performance Computing (EnA-HPC)*. IEEE, 2021. URL: <https://doi.org/10.1109/EnergyAwareHPC.2021.00010>.
- [4] I. Dragan. “Computational Geometry: A Complete Guide”. En: (2026). URL: <https://www.cs.kent.edu/~dragan/CG/CG-Book.pdf>.
- [5] D. Lee y F. Preparata. “Computational Geometry: A Survey”. En: (1984). URL: [https://www.researchgate.net/publication/3048370\\_Comptational\\_Geometry-A\\_Survey](https://www.researchgate.net/publication/3048370_Comptational_Geometry-A_Survey).
- [6] F. P. Preparata y M. I. Shamos. “Computational Geometry: An Introduction”. En: (1985). URL: <https://www.cs.jhu.edu/~misha/ReadingSeminar/Papers/PreparataShamos85.pdf>.
- [7] C. Bradford Barber, David P. Dobkin y Hannu Huhdanpaa. “The QuickHull Algorithm for Convex Hulls”. En: (1998). URL: [https://www.researchgate.net/publication/2641780\\_The\\_QuickHull\\_Algorithm\\_for\\_Convex\\_Hulls](https://www.researchgate.net/publication/2641780_The_QuickHull_Algorithm_for_Convex_Hulls).
- [8] Franco P. Preparata y S. J. Hong. “Convex Hulls of Finite Sets of Points in Two and Three Dimensions”. En: (1977). URL: <https://www.cs.jhu.edu/~misha/Spring16/Preparata77.pdf>.

- [9] T. M. Chan. “Optimal output-sensitive convex hull algorithms in two and three dimensions”. En: (1996). URL: <https://doi.org/10.1007/BF02712873>.