

# **Estudio y Análisis de Metaheurísticas Modernos para la Selección de Características**

**Titulación:**

Grado en Ingeniería Informática.

**Autor:**

Miguel García López

**Director:**

Daniel Molina Cabrera

---

<https://github.com/Migue8gl/>

TFG-Wrapper-Based-Metaheuristics-Feature-Selection



**UNIVERSIDAD  
DE GRANADA**

---

**ETSIIT**

Escuela Técnica Superior  
de Ingenierías Informática  
y de Telecomunicación



# Índice

## 1 Introducción

- Motivación
- Contexto
- Objetivos
- Metaheurísticas

## 2 Planificación

- Presupuesto y planificación

## 3 Revisión de la literatura

- Búsquedas Scopus

- Algoritmos

## 4 Diseño Experimental

- Conjuntos de datos

## 5 Resultados y Análisis

- Continuos

- Binarios

## 6 Conclusiones

- Respuestas

- ① **La selección de características** es un procesamiento esencial en multitud de problemas relacionados con el aprendizaje automático.
  - ② Existen muchos algoritmos y métodos para abordar este problema. En este proyecto, nos centraremos en las metaheurísticas.
  - ③ Se han propuesto muchos algoritmos metaheurísticos que se aplican a este problema. Tanto versiones con codificación real como binaria.
  - ④ Pese al gran número de algoritmos y versiones, no existen apenas comparaciones entre ellas.

# Motivación

- 1 Este proyecto pretende hacer comparaciones rigurosas entre algoritmos.
  - 2 Comparaciones a nivel de distintas métricas, tales como el *accuracy* o la reducción de características.
  - 3 Uso de test estadísticos robustos.
  - 4 Con el fin de dar la información más precisa sobre los algoritmos metaheurísticos en este problema.

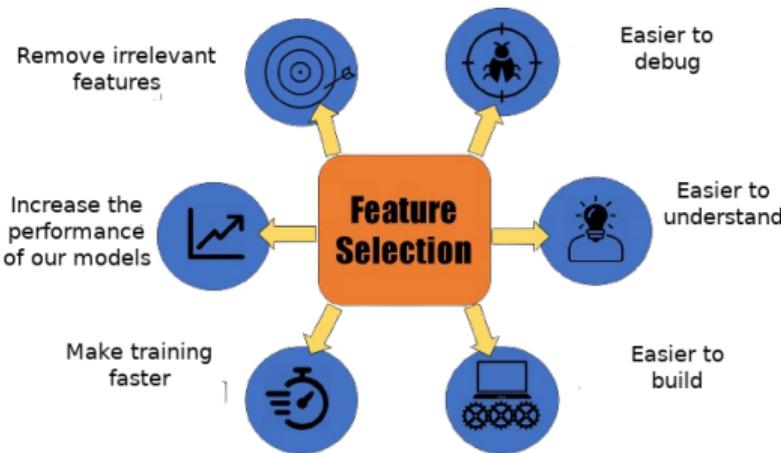
# Introducción a la Selección de Características

1 La selección de características es un proceso crucial en el aprendizaje automático.

- Implica elegir un subconjunto de características relevantes.
- Es un problema **NP duro**.

2 Importancia de la reducción de dimensionalidad:

- Mejora la generalización y precisión del modelo.
- Reduce el ruido en los datos.



Beneficios de la selección de características

# Objetivos

- 1 Evaluar el desempeño de las metaheurísticas.
- 2 Investigar versiones continuas y binarias.
- 3 Fortalezas y debilidades de las metaheurísticas.
- 4 Recomendaciones prácticas según el contexto.
- 5 Evaluación de resultados finales y realizar comparaciones.
- 6 Resolver una serie de preguntas de investigación. ¿Binarios vs continuos?

1	2	3	4	5	6
0	1	1	0	1	1

(a) Binary Encoding

1	2	3	4	5	6
86.3	115.0	10.7	121.9	1.6	64.2

(b) Real Number Encoding

1	2	3	4	5	6
A	F	D	E	C	B

(c) Symbol Encoding

Codificaciones binarias y continuas<sup>1</sup>

<sup>1</sup>Shing-Tai Pan, Chih-Hung Wu, Chen-Sen Ouyang y Ying-Wei Lee. «Emotion Recognition from Speech Signals by Using Evolutionary Algorithm and Empirical Mode Decomposition». En: jul. de 2018, págs. 140-147. DOI: 10.14236/ewic/EVA2018.29

# ¿Qué son las metaheurísticas?

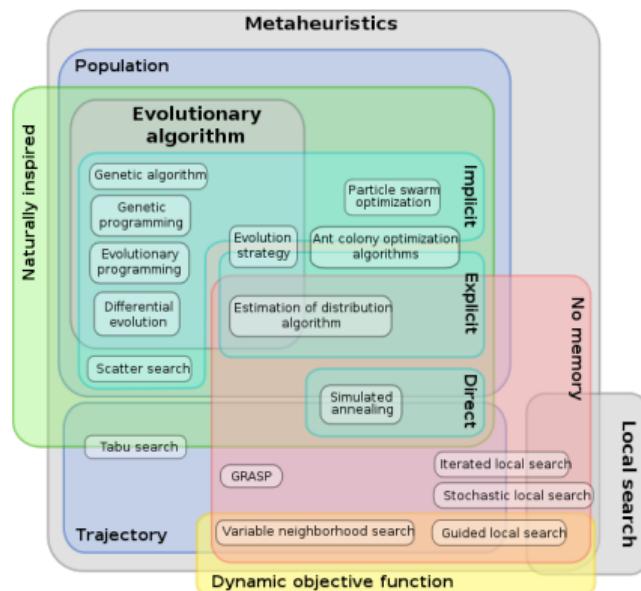
- ➊ Las metaheurísticas son algoritmos de optimización.
- ➋ Suelen estar bioinspiradas.
- ➌ Son algoritmos generales que no dependen de un dominio en un problema específico.
- ➍ Consiguen soluciones muy buenas en tiempos razonables.

Algoritmo PSO en busca del **mínimo** de una función<sup>2</sup>

---

<sup>2</sup>Wikipedia contributors. *Particle swarm optimization – Wikipedia, The Free Encyclopedia*. [Online; accessed 26-June-2024]. 2024. URL: [https://en.wikipedia.org/wiki/Particle\\_swarm\\_optimization](https://en.wikipedia.org/wiki/Particle_swarm_optimization)

## Tipos de mateheurísticas



## Tipos de metaheurísticas

- 1 Las metaheurísticas seleccionadas en este proyecto son todas del tipo **poblacional** y evolutivas.

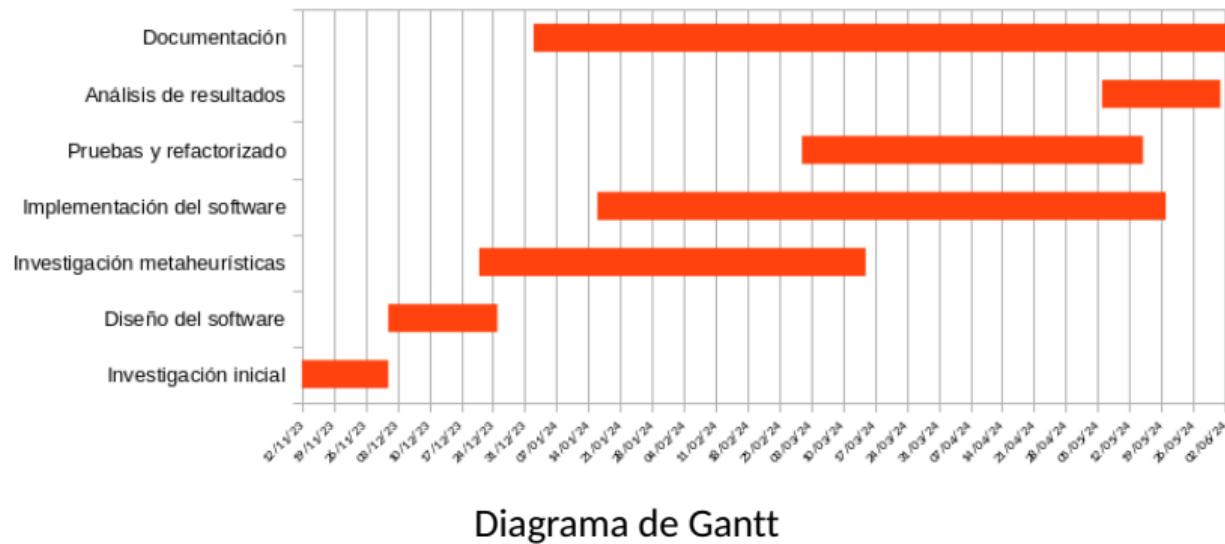


# Tareas

Tarea	Duración Prevista (h)	Duración Final (h)
Investigación inicial	20	20
Diseño del software	10	10
Investigación metaheurísticas	60	60
Implementación del software	70	90
Pruebas y refactorizado	15	25
Ánalysis de resultados	40	20
Documentación	90	120

Tabla de duración de cada tarea

## Diagrama de Gantt



# Presupuesto

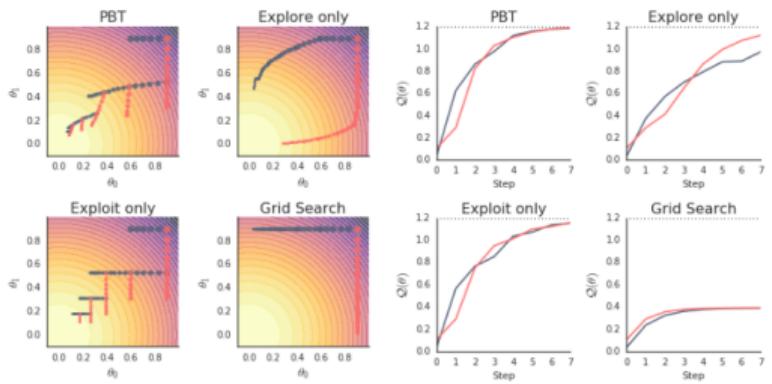
- **Sueldo:** 25€/hora, 340 horas, total 8,500€.
  - **Portátil:** HP Pavilion, 1000€, usado 8 meses, costo 133,3€.
  - **Servidor:** Uso del servidor **Hércules**, costo real cero. Estimación en Google Cloud.

# Presupuesto

Item	Costo (€)
Salario	8500
Ordenador portátil	133.3
Servidor CPU - GC Compute Engine	75.84
<b>Total</b>	<b>8709.14</b>

Costo estimado del proyecto

## Ventajas actuales de las Metaheurísticas

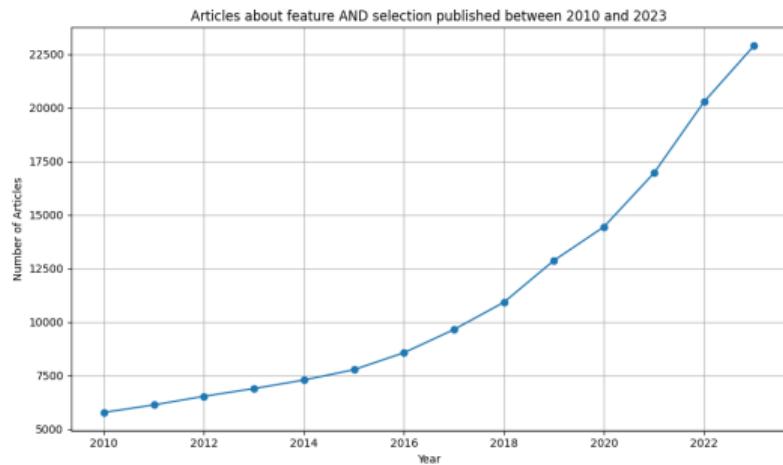


## Uso de algoritmo poblacional PBT en búsqueda de hiperparámetros<sup>3</sup>

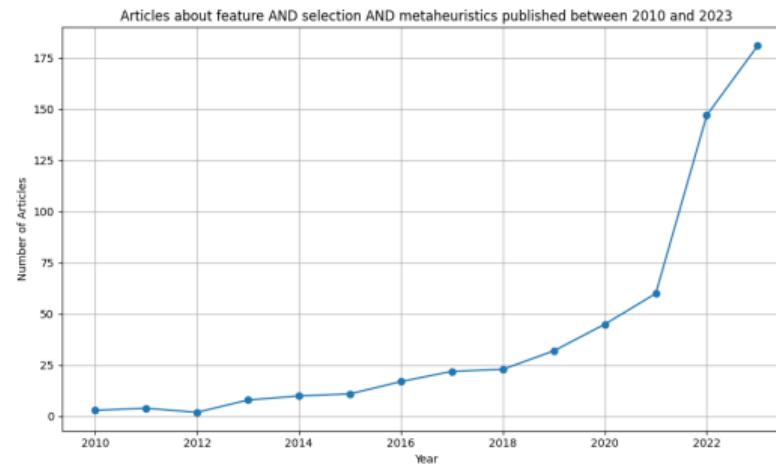
- **Implementación práctica:** Utilizadas en aplicaciones reales, como la planificación de rutas, diseño de redes y gestión de recursos.
  - **Integración con AI:** Combinadas con técnicas de inteligencia artificial para mejorar el rendimiento.
  - **Adaptabilidad:** Ajustables a problemas específicos mediante parametrización de los cuales no es posible obtener funciones de pérdida diferenciables.

<sup>3</sup>M. Mafarja, I. Aljarah, A. A. Heidari, A. I. Hammouri, H. Faris, A. M. Al-Zoubi y S. Mirjalili. «Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems». En: *Knowledge-Based Systems* 145 (2018). Cited by: 351; All Open Access, Green Open Access, págs. 25-45. DOI: 10.1016/j.knosys.2017.12.037. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85040020546&doi=10.1016%2fj.knosys.2017.12.037&partnerID=40&md5=48c70ba8b45fb49d6218a39cfdc5c11d>

## Tendencia Scopus



Tendencia en artículos publicados de **feature selection** en Scopus. Se incrementa exponencialmente con el tiempo.



Tendencia en artículos publicados sobre **feature selection** usando **metaheurísticas** en Scopus. La tendencia es igualmente exponencial, aunque el número total no es muy alto.

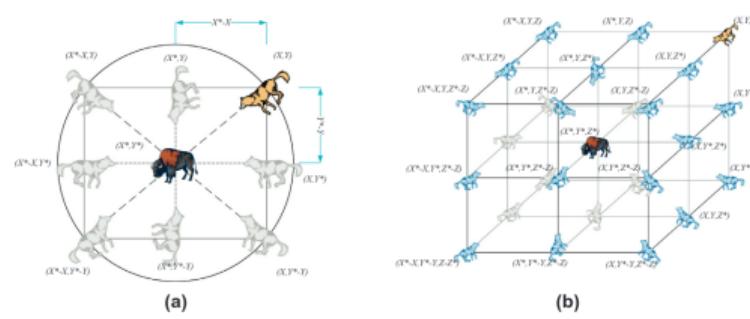


# Algoritmos seleccionados

- ① Se escogen para el proyecto una serie de algoritmos basándose en la investigación de aquellos más novedosos, con mejor rendimiento y más citados. Estos son los que se denominarán **modernos**.
- ② Además de los algoritmos más novedosos, se incluyen una serie de algoritmos clásicos, cuyo robustez a lo largo de los años tras multitud de aplicaciones en problemas es notable. Esta categoría, es la de los algoritmos **clásicos**.

# Grey Wolf Optimizer

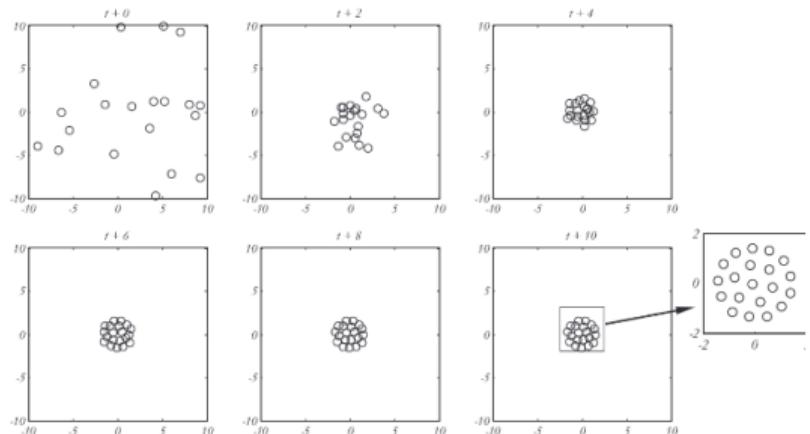
- 1 Inspirado en el comportamiento social y la técnica de caza de los lobos grises.
  - 2 Utiliza una **jerarquía social** para guiar la búsqueda de soluciones óptimas, donde los lobos alfa, beta y delta lideran el proceso de exploración y explotación.



## Caza de los lobos grises<sup>4</sup>

<sup>4</sup>S. Mirjalili, S. M. Mirjalili y A. Lewis. «Grey Wolf Optimizer». en-US. En: *Advances in Engineering Software* 69 (mar. de 2014), págs. 46-61. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2013.12.007. URL: <https://www.sciencedirect.com/science/article/pii/S0965997813001853> (visitado 18-11-2023)

# Grasshopper Optimization Algorithm



Convergencia de los saltamontes<sup>5</sup>

- 1 Simula el movimiento y la interacción de los saltamontes en sus distintas etapas de vida para encontrar soluciones óptimas.

<sup>5</sup>S. Saremi, S. Mirjalili y A. Lewis. «Grasshopper Optimisation Algorithm: Theory and application». en-US. En: *Advances in Engineering Software* 105 (mar. de 2017), págs. 30-47. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2017.01.004. URL: <https://www.sciencedirect.com/science/article/pii/S0965997816305646> (visitado 04-11-2023)

# Firefly Algorithm

- 1 Inspirado en el comportamiento de **parpadeo y atracción** de las luciérnagas.
- 2 Utiliza la intensidad de la luz (o brillo) como guía para la atracción entre luciérnagas, donde las luciérnagas menos brillantes se mueven hacia las más brillantes.



Imagen de una libélula con su característico brillo

# Cuckoo Search



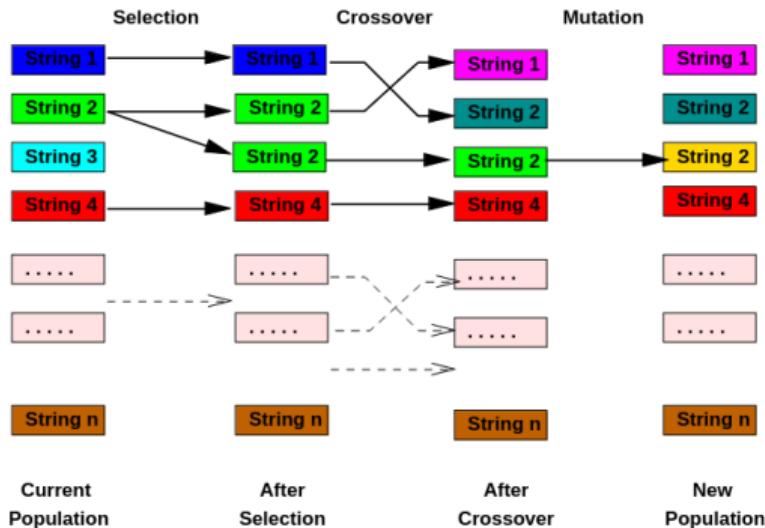
Cuatro nidos de huevos de pájaro. En cada uno de ellos un huevo visiblemente más grande del pájaro Cuco<sup>6</sup>

- ➊ Inspirado en el comportamiento de anidación de los cucos y el **parasitismo** de puesta.
- ➋ Caracterizado por usar métodos de búsqueda aleatoria y el método **Levy flight** para la exploración del espacio de soluciones.

<sup>6</sup>Chiswick Chap. Cuckoo Eggs Mimicking Reed Warbler Eggs. CC BY-SA 3.0. 2024. URL: <https://commons.wikimedia.org/w/index.php?curid=18861052>

# Genetic Algorithm

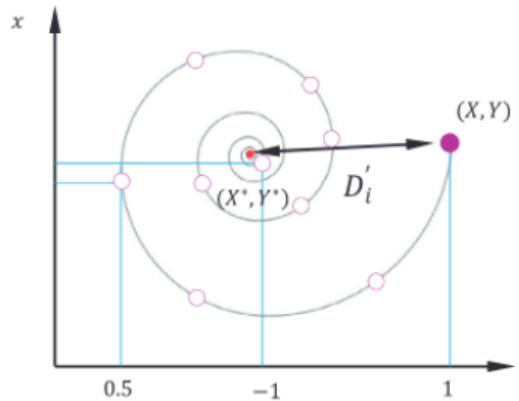
- 1 Algoritmo basado en la recombinación de cromosomas, que toma inspiración de la **evolución** biológica y genética.
- 2 Hace uso de operadores tales como la **mutación** o el **cruce**.



Principio básico del algoritmo GA <sup>7</sup>

<sup>7</sup>T. V. Mathew. «Genetic algorithm». En: *Report submitted at IIT Bombay 53* (2012)

# Whale Optimization Algorithm



Espiral para simular el mecanismo de ataque de la red de burbujas de las ballenas jorobadas<sup>8</sup>

- ① Inspirado en el comportamiento de las ballenas jorobadas.
- ② Usa principalmente dos variantes de operadores de caza:
  - Espiral de búsqueda.
  - Técnica de burbujeo de red.

<sup>8</sup>S. Mirjalili y A. Lewis. «The Whale Optimization Algorithm». en-US. En: *Advances in Engineering Software* 95 (mayo de 2016), págs. 51-67. ISSN: 0965-9978. DOI: 10.1016/j.advengsoft.2016.01.008. URL: <https://www.sciencedirect.com/science/article/pii/S0965997816300163> (visitado 14-10-2023)

# Artificial Bee Colony Optimization

- 1 Simula la **búsqueda de alimentos** de las abejas empleadas, las abejas observadoras y las abejas exploradoras para encontrar soluciones óptimas.

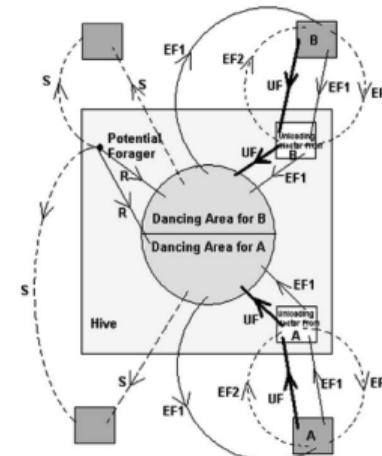
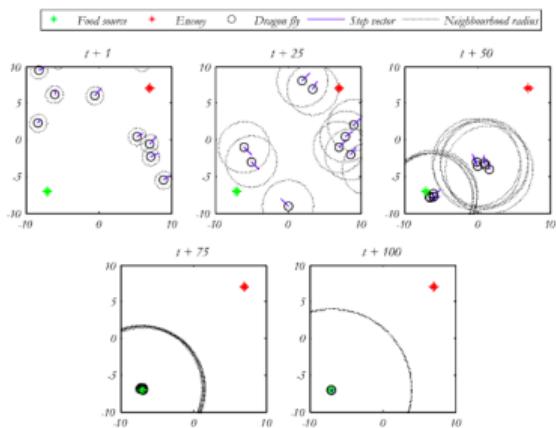


Diagrama de funcionamiento del ABCO <sup>9</sup>

<sup>9</sup>D. Karaboga y B. Akay. «A comparative study of Artificial Bee Colony algorithm». En: *Applied Mathematics and Computation* 214.1 (2009). Cited by: 2940, págs. 108-132. DOI: 10.1016/j.amc.2009.03.090. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-67349273050&doi=10.1016%2fj.amc.2009.03.090&partnerID=40&md5=505464030a4a96a1998b20803cd113ce>

# Dragonfly Algorithm



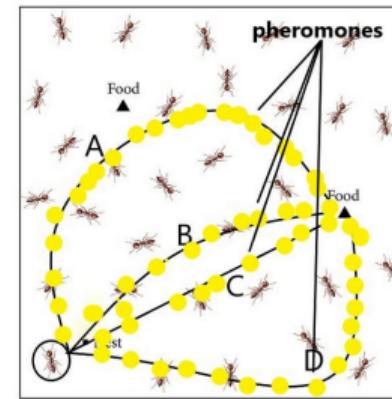
Operadores del algoritmo DA<sup>10</sup>

- 1 Basado en el comportamiento de enjambre y formación de las libélulas, usando operadores que controlan características como la **cohesión** de grupo o **distanciamiento** del enemigo, entre otros.
- 2 Simula las interacciones sociales y el movimiento de las libélulas para equilibrar la exploración y explotación del espacio de soluciones.

<sup>10</sup>Yassine Meraihi, Amar Ramdane-Cherif, Dalila Acheli y Mohammed Mahseur. «Dragonfly algorithm: a comprehensive review and applications». En: *Neural Computing and Applications* 32.21 (nov. de 2020), págs. 16625-16646. ISSN: 1433-3058. DOI: 10.1007/s00521-020-04866-y. URL: <https://doi.org/10.1007/s00521-020-04866-y>

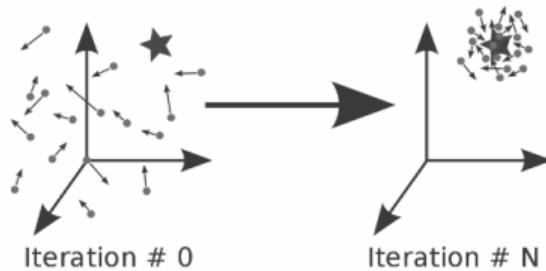
# Ant Colony Optimization

- 1 Simula las colonias de hormigas. Para ello usa el rastro de **feromonas** para guiar la búsqueda de soluciones óptimas, donde las hormigas depositan y siguen feromonas en los caminos más prometedores.



Caminos de un grafo marcados por la feromona,  
operador esencial de ACO

# Particle Swarm Optimization

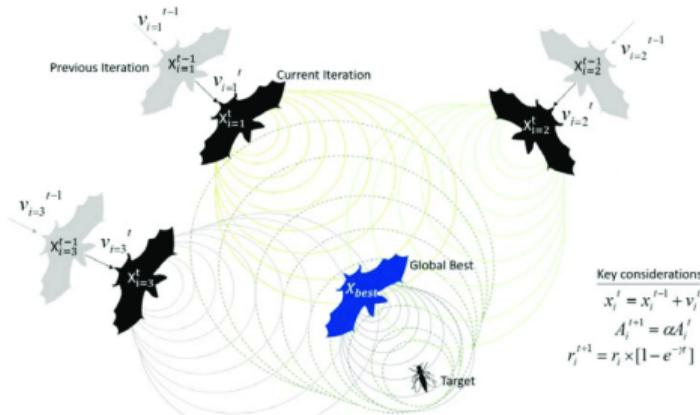


Partículas en el espacio (con una velocidad y dirección) convergiendo en la iteración  $N$

- ➊ Inspirado en el comportamiento social de los enjambres de aves y peces.
- ➋ Simula la búsqueda colectiva de soluciones, donde cada partícula ajusta su posición basada en su **experiencia** personal y la de sus **vecinos**.

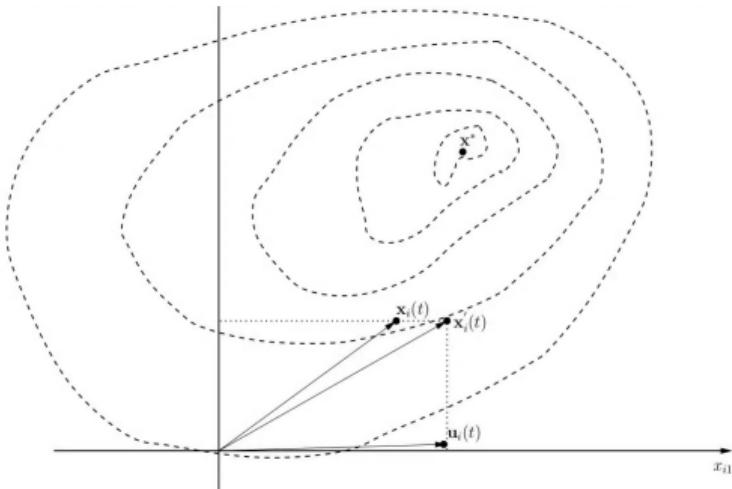
# Bat Algorithm

- 1 Basado en el comportamiento de **ecolocalización** de los murciélagos.
- 2 Utiliza la técnica de emisión de pulsos y el ajuste de frecuencia para explorar y explotar el espacio de soluciones.



Funcionamiento del algoritmo BA

# Differential Evolution



- 1 Utiliza la combinación y mutación de vectores solución para buscar la mejor solución, enfocándose en la **diferencia** entre las soluciones actuales para generar nuevas.

Operador de crossover o cruce de DE <sup>11</sup>

---

<sup>11</sup>\cite{10.5555/1557464}

# Conjuntos de datos

1 Se escogen conjuntos de datos por:

- Variedad de áreas
- Diversidad de problemas
- Número de características
- Relevancia práctica

Dataset	Inst.	Car.	Cl.	Área
sonar	207	60	2	Biología
spambase-460	459	54	2	Informática
spectf-heart	348	44	2	Medicina
waveform5000	5000	40	3	Física
ionosphere	350	34	2	Meteorología
dermatology	366	34	6	Medicina
wdbc	568	29	2	Medicina
parkinsons	200	22	2	Medicina
zoo	101	18	7	Biología
wine	182	13	3	Alimentación
breast-cancer	286	9	2	Medicina
diabetes	768	8	2	Medicina
yeast	1483	8	10	Biología
ecoli	336	7	8	Biología
iris	149	4	3	Biología

Información de conjuntos de datos por número de características

# Diseño Experimental

1 La función *fitness* se construye con:

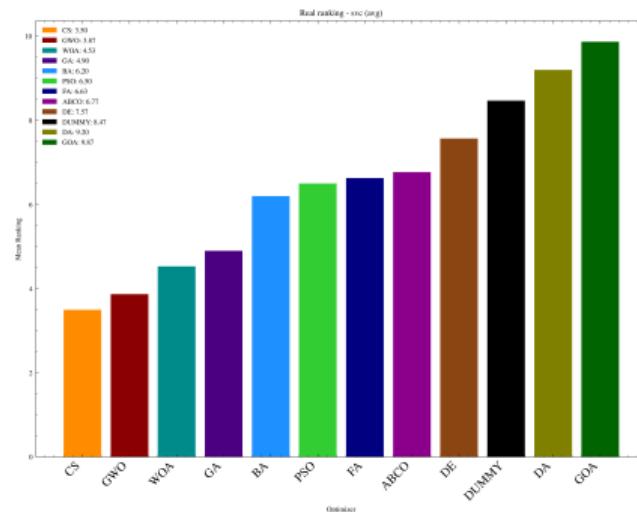
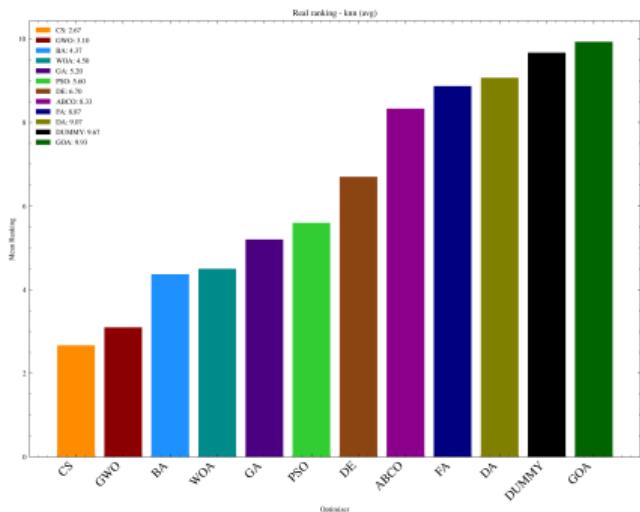
- Accuracy al 90 %.
- Reducción al 10 %.

2 Se define como:

$$\textit{fitness} = \textit{acc} \cdot \alpha + \textit{red} \cdot (1 - \alpha) \quad (1)$$

Donde  $\alpha$  es la ponderación dada a la precisión o *accuracy*.

# Ranking en continuos para fitness



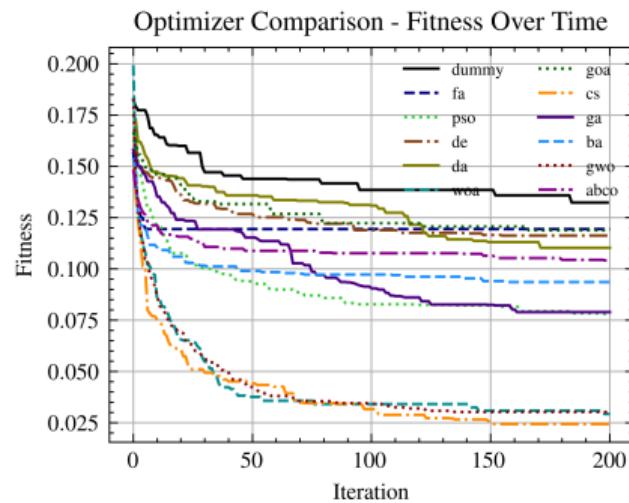
Ranking de los algoritmos en versión continua para kNN

Ranking de los algoritmos en versión continua para SVC

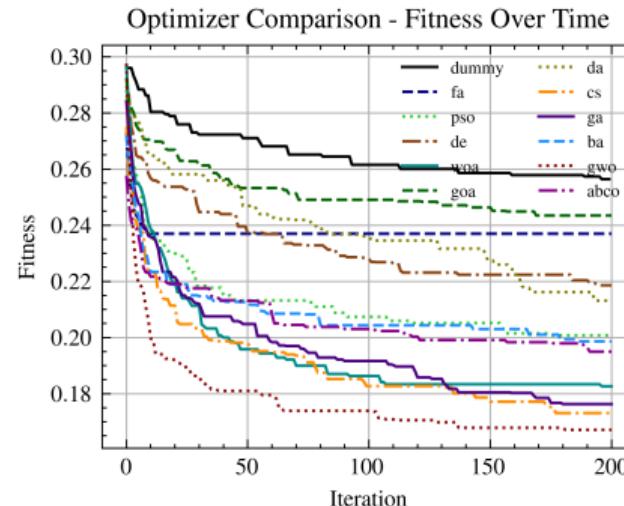
# Resultados en continuos

- ➊ Los mejores algoritmos en *fitness* son **CS** y **GWO**. Los peores algoritmos son **GOA** y **DA**.
- ➋ Los mejores reduciendo características vuelven a ser **CS** y **GWO**. Además con mucha diferencia.
- ➌ El algoritmo más rápido es **FA**, mientras que el más lento es **ABCO**.

## **Convergencia en continuo**



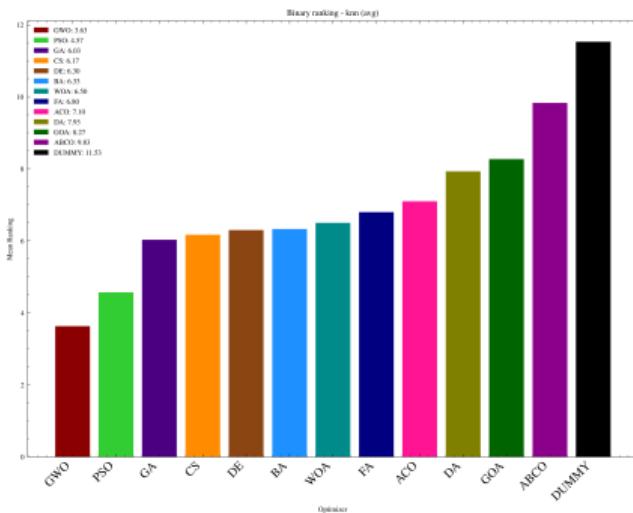
## Convergencia de todas las metaheurísticas en ionosphere - knn - real



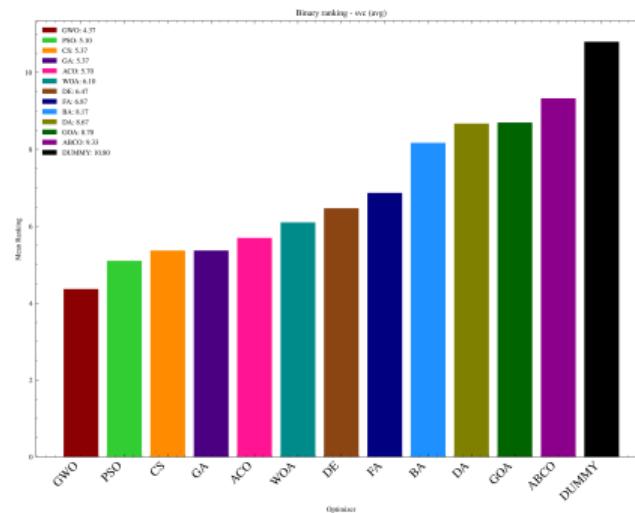
## Convergencia de todas las metaheurísticas en diabetes - knn - real



# Ranking en binario para fitness



Ranking de los algoritmos en versión binaria para kNN

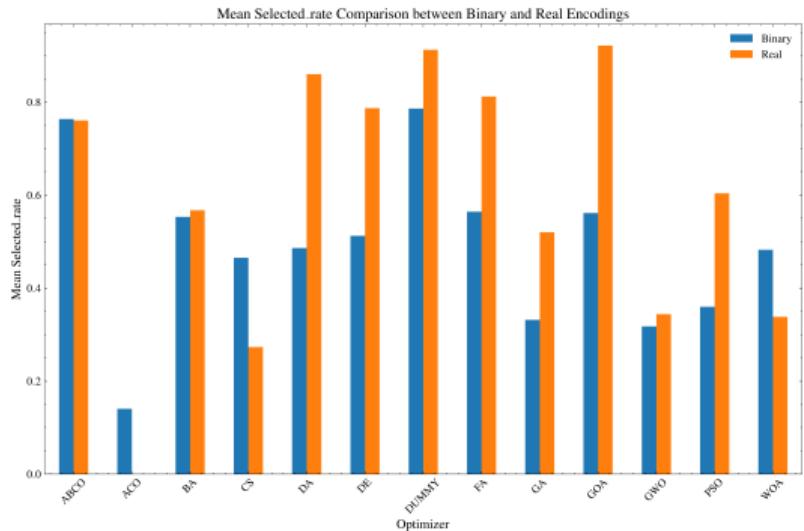


Ranking de los algoritmos en versión binaria para SVC

# Resultados en binarios

- ① Los mejores algoritmos en *fitness* son **bGWO** y **bPSO**. Los peores algoritmos son **bGOA** y **bABCO**.
- ② El mejor reduciendo características es **ACO**.
- ③ El algoritmo más rápido vuelve a ser **bFA**. Ocurre igual con el más lento, que es **ABCO**.

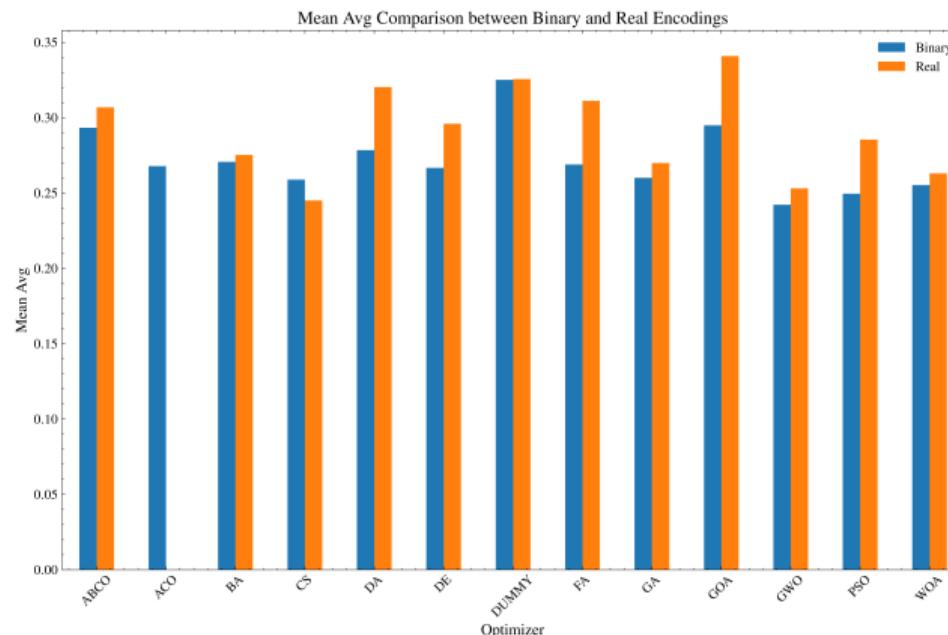
# Preguntas investigadas



- ¿Merece pues la pena el uso de algoritmos específicos para la selección de características o las versiones originales son totalmente capaces de reducir?

Comparación binaria vs continuo en selección de características

## Preguntas investigadas



## Comparación de *fitness* en binario vs continuo



## Preguntas investigadas

- ¿Cómo son los recientes en comparación con los más clásicos?
  - ¿Cuáles de los recientes parecen más prometedores?
  - ¿Son los algoritmos buenos en su versión original igualmente eficaces en su versión binaria?
  - ¿Cuáles son las opciones más interesantes dentro de ciertos contextos?

# Agradecimientos

Gracias por su atención.

¿Dudas, preguntas o comentarios?

