

# Algoritmo de búsqueda gravitacional

Chávez Castañeda Pablo, Corona Anchondo José Antonio, Sánchez Flores  
María Fernanda, and Franco Rodríguez María Guadalupe

Escuela Superior de cómputo, Instituto Politécnico Nacional, Unidad Zacatenco

**Abstract.** El algoritmo de búsqueda gravitacional es un método de optimización metaheurístico, perteneciente al grupo de los algoritmos bioinspirados, cuyo objetivo es minimizar una función objetivo en un espacio de búsqueda complejo. Este proceso se realiza mediante la configuración de varios parámetros, que permiten generar agentes capaces de atraer a otros agentes, basándose en la ley de la gravitación universal.

Los agentes con mejor valor de fitness (mayor calidad como solución) poseen mayor masa y ejercen una mayor atracción sobre los demás, guiando así la búsqueda hacia soluciones óptimas.

Entre los parámetros principales utilizados para su configuración se encuentran: el número de agentes o partículas, la constante gravitacional, el número de iteraciones y el valor de K-best, que define cuántos agentes influyen sobre los demás en cada paso. Estos elementos permiten al algoritmo explorar eficientemente el espacio de búsqueda y concentrarse en las regiones más prometedoras. Una vez que se alcanza una solución óptima o un mínimo local, el algoritmo tiende a estabilizarse alrededor de esa región, mejorando la precisión de la optimización.

En este trabajo, el algoritmo fue aplicado al problema de optimización del diseño de una viga, logrando encontrar soluciones que minimizan su peso cumpliendo con las restricciones.

**Keywords:** Algoritmos Bioinspirados · Algoritmo de búsqueda gravitacional · Función objetivo · Optimización metaheurística · Ley de gravitación universal.

## 1 Introducción

El algoritmo de búsqueda gravitacional (Gravitational Search Algorithm, GSA) es un método de optimización metaheurística propuesto por Rashedi, Nezamabadi-pour y Saryazdi en el año 2009, basado en la ley de la gravitación universal, propuesta por el científico y matemático Isaac Newton.

Esta ley describe la interacción de las masas dentro de un espacio de búsqueda, donde cada masa es considerada un agente buscador que se encarga de generar desplazamiento entre los demás agentes, en función de la gravedad y las leyes del movimiento; de esta manera se logra impulsar a las masas hacia soluciones potencialmente óptimas.

El uso de este algoritmo está dirigido a espacios complejos, en donde la calidad

de los agentes es evaluada por la función objetivo y se reajusta su rendimiento. Cuanto mejor es la solución del algoritmo, mayor es la masa y, por lo tanto, aumenta el nivel de atracción sobre otros agentes.

Este algoritmo introduce una interacción física directa entre las soluciones, lo que provoca un equilibrio entre la exploración global y la explotación local. Ha sido implementado en diversos tipos de problemas de optimización. En este caso, se decidió aplicarlo al problema del diseño de una viga, con el objetivo de encontrar una solución óptima que minimice su peso, respetando las restricciones físicas y estructurales.

## 2 Trabajos relacionados

Algunos de los trabajos relacionados utilizando el algoritmo de búsqueda gravitacional son:

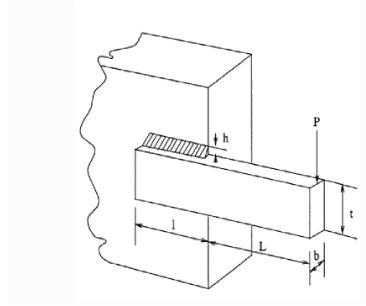
Algoritmo de búsqueda gravitacional: Se propuso el algoritmo de búsqueda gravitacional, basado en la ley de gravitación universal de Newton. En su estudio original, aplicaron el algoritmo a diversos problemas matemáticos de optimización, demostrando su capacidad para encontrar soluciones eficientes, proponiendo que cada agente representa una posible solución regida por la fuerza gravitacional, entre mejor fitness, mayor probabilidad de alcanzar un óptimo local dentro del espacio de búsqueda. Esta propuesta es la base del GSA.

GSA para optimización de estructuras de contención: Para este problema se implementó el GSA al diseño óptimo de estructuras de retención (muros de contención), enfocándose en minimizar el costo total mientras se cumplen todas las restricciones geotécnicas y estructurales como lo son las dimensiones, estabilidad, y propiedades del suelo. Demostrando que el GSA es óptimo para problemas complejos.

PSOGSA: Se propuso un algoritmo híbrido que combina el PSO y el GSA para mejorar la exploración y explotación en problemas de optimización. Aplicado a funciones matemáticas estándar, el método mostró mejor convergencia y evitó óptimos locales, ya que, se utilizó la capacidad exploratoria del GSA y la velocidad de convergencia del PSO, mejorando el rendimiento significativamente ante problemas complejos.[1], [2].

### 3 Propuesta de solución

El problema a resolver es la optimización del diseño de una viga soldada, como se aborda en [3], [4], [5], [6], el cual está sujeto a múltiples restricciones estructurales y físicas. Se propone utilizar el **Algoritmo de Búsqueda Gravitacional**, propuesto por [7], como técnica metaheurística para encontrar la mejor solución factible que minimice la función objetivo.



**Fig. 1.** Representación del problema. Fuente: [3]

#### 3.1 Función Objetivo

Minimizar el costo de fabricación de la viga, definido por la siguiente expresión:

$$f(x_1, x_2, x_3, x_4) = 1.10471 x_1^2 x_2 + 0.04811 x_3 x_4 (14 + x_2)$$

donde las variables de decisión están definidas como:

- $x_1$ : ancho del cordón de soldadura
- $x_2$ : longitud del cordón de soldadura
- $x_3$ : altura del refuerzo
- $x_4$ : grosor del refuerzo

#### 3.2 Restricciones

El problema considera siete restricciones no lineales relacionadas con esfuerzos cortantes, tensiones normales, deflexión máxima, entre otras. Todas las restricciones están implementadas mediante funciones de penalización que incrementan el valor de la función objetivo si se violan.

$$\begin{aligned}
g_1(x) &= \tau(x) - 13,600 \leq 0 \\
g_2(x) &= \sigma(x) - 30,000 \leq 0 \\
g_3(x) &= x_1 - x_4 \leq 0 \\
g_4(x) &= 0.10471 x_1^2 + 0.04811 x_3 x_4 (14 + x_2) - 5 \leq 0 \\
g_5(x) &= 0.125 - x_1 \leq 0 \\
g_6(x) &= \delta(x) - 0.25 \leq 0 \\
g_7(x) &= 6,000 - P_c(x) \leq 0
\end{aligned}$$

donde:

- $\tau(x)$ : esfuerzo cortante combinado,
- $\sigma(x)$ : tensión normal,
- $\delta(x)$ : deflexión máxima,
- $P_c(x)$ : carga crítica de pandeo.

### 3.3 Técnica de Optimización

La solución propuesta consiste en aplicar el algoritmo **Gravitational Search Algorithm (GSA)** con las siguientes características:

- **Representación de partículas:** cada agente representa una posible solución del problema, expresada como un vector de variables continuas  $(x_1, x_2, x_3, x_4)$ .
- **Función objetivo con penalización:** la evaluación del fitness de cada partícula incluye tanto el valor de la función a minimizar como una penalización proporcional al grado de violación de las restricciones del problema.
- **Dinámica gravitacional:** las partículas actualizan sus posiciones y velocidades en cada iteración en función de fuerzas gravitacionales simuladas, determinadas por la aptitud relativa de cada solución.
- **Adaptación del parámetro gravitacional:** la constante gravitacional  $G$  se reduce exponencialmente con el tiempo para promover la explotación en etapas avanzadas de la búsqueda.
- **Mecanismo de *early stopping*:** si no se detecta mejora tras un número determinado de iteraciones, se reinicia la población conservando la mejor solución hallada hasta el momento, lo que ayuda a escapar de óptimos locales.
- **Evaluación de configuraciones:** se exploran múltiples combinaciones de hiperparámetros mediante un diseño experimental, y se analizan los resultados usando un análisis de varianza (ANOVA) para determinar el impacto individual de cada parámetro.
- **Visualización y monitoreo:** se incluyen gráficas de convergencia para visualizar el comportamiento del algoritmo a lo largo de las iteraciones.

### 3.4 Pseudocódigo

A continuación, se presenta el pseudocódigo del algoritmo **Gravitational Search Algorithm (GSA)** implementado para resolver el problema de optimización propuesto. Este esquema resume las principales etapas del algoritmo: inicialización de la población, evaluación de aptitud con penalización, actualización dinámica basada en interacciones gravitacionales, y mecanismos adicionales como *early stopping* para mejorar la eficiencia y robustez de la búsqueda.

---

#### Algorithm 1 Gravitational Search Algorithm (GSA)

---

```

1: procedure GSA( $n\_particulas, n\_iteraciones, G_0, \alpha, dimensiones, rango$ )
2:   Inicializar  $x\_posiciones, v\_velocidades \leftarrow$  INICIALIZARPOBLACION
3:    $mejor\_valor \leftarrow \infty, mejor\_solucion \leftarrow$  None
4:   Inicializar historial de fitness
5:    $paciencia \leftarrow 0.3 \cdot n\_iteraciones, cuenta\_sin\_mejora \leftarrow 0$ 
6:   for  $t = 1$  to  $n\_iteraciones$  do
7:      $fitness \leftarrow$  EVALUARPOBLACION( $x\_posiciones$ )
8:     Actualizar historial de mejor, peor y promedio
9:      $mejora \leftarrow$  False
10:    for cada  $x_i$  en  $x\_posiciones$  do
11:      if PENALIZACION( $x_i$ ) == 0 and  $fitness_i < mejor\_valor$  then
12:         $mejor\_valor \leftarrow fitness_i$ 
13:         $mejor\_solucion \leftarrow x_i$ 
14:         $mejora \leftarrow$  True
15:      end if
16:    end for
17:    if  $mejora$  then
18:       $cuenta\_sin\_mejora \leftarrow 0$ 
19:    else
20:       $cuenta\_sin\_mejora \leftarrow cuenta\_sin\_mejora + 1$ 
21:    end if
22:    if  $cuenta\_sin\_mejora \geq paciencia$  then
23:      Reiniciar población y conservar mejor solución
24:      continue
25:    end if
26:     $masas \leftarrow$  CALCULARMASAS( $fitness$ )
27:     $G \leftarrow G_0 \cdot e^{-\alpha \cdot t / n\_iteraciones}$ 
28:     $F \leftarrow$  CALCULARFUERZA( $G, x\_posiciones, masas$ )
29:    Actualizar  $x\_posiciones, v\_velocidades \leftarrow$  ACTUALIZARPOSICIONVELOCIDADES( $v\_velocidades, x\_posiciones, F, masas, rango$ )
30:  end for
31:  GRAFICARCONVERGENCIA( $historial$ )
32:  return  $mejor\_solucion, mejor\_valor$ 
33: end procedure

```

---

## 4 Pruebas y resultados

Para evaluar el desempeño del algoritmo **Gravitational Search Algorithm (GSA)** en el problema de optimización propuesto, se llevaron a cabo múltiples experimentos variando los principales hiperparámetros del algoritmo. En particular, se consideraron las siguientes variables:

- Tamaño de la población inicial
- Número de iteraciones
- Constante gravitacional inicial ( $G_0$ )
- Tasa de decaimiento gravitacional ( $\alpha$ )

Cada configuración fue evaluada con **10 ejecuciones independientes** utilizando diferentes semillas aleatorias para asegurar robustez estadística. La siguiente tabla resume los mejores resultados para cada configuración, los valores promedio, mínimo (mejor), máximo (peor) y desviación estándar de la función objetivo obtenida:

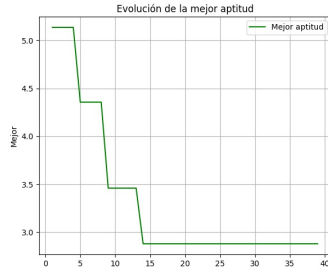
Configuración	Promedio de la función objetivo	Mejor valor de la función objetivo	Peor valor de la función objetivo	Desviación estándar de la función objetivo
1	3.206702	2.644799	4.18592	0.614372
Población inicial: 50				
Iteraciones: 40				
Constante gravitacional: 50				
Alpha: 5	3.57926	2.720657	5.245631	1.010874
2				
Población inicial: 50				
Iteraciones: 40				
Constante gravitacional: 50	3.41499	2.698653	4.003103	0.521629
Alpha: 10				
3				
Población inicial: 50				
Iteraciones: 40	3.75489	3.089025	4.504137	0.655203
Constante gravitacional: 100				
Alpha: 5				
Alpha: 10	3.405088	2.794363	4.628002	0.711121
4				
Población inicial: 50				
Iteraciones: 50				
Constante gravitacional: 50	3.405088	2.794363	4.628002	0.711121
Alpha: 5				

**Fig. 2.** Tabla de resultados con 5 configuraciones distintas (10 ejecuciones por cada configuración).

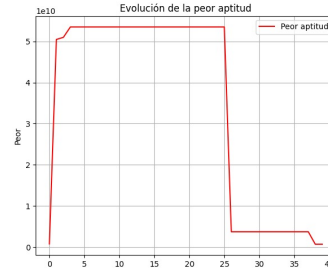
A continuación, se presentan las gráficas de convergencia de la mejor ejecución (menor valor de la función objetivo) para tres de las configuraciones seleccionadas. En cada caso se visualiza la evolución de la mejor, peor y promedio aptitud a lo largo de las iteraciones.

**Configuración 1:**

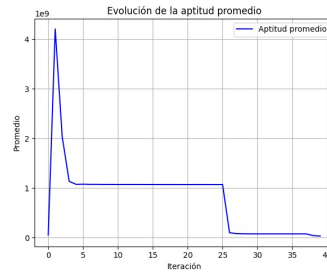
**Población: 50 Iteraciones: 40  $G_0$ : 50 Alpha: 5**



**Fig. 3.** Evolución de la mejor aptitud.



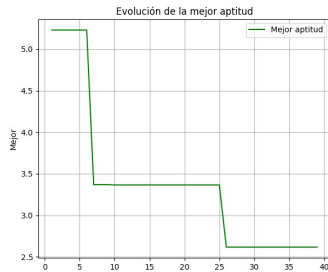
**Fig. 4.** Evolución de la peor aptitud.



**Fig. 5.** Evolución de la aptitud promedio.

**Configuración 2:**

**Población: 50 Iteraciones: 40  $G_0$ : 50 Alpha: 10**



**Fig. 6.** Evolución de la mejor aptitud.



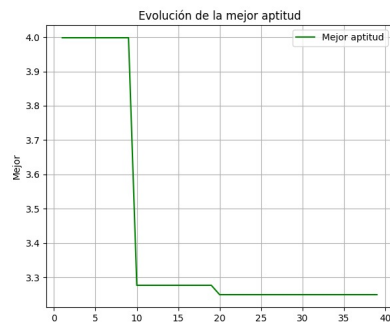
**Fig. 7.** Evolución de la peor aptitud.



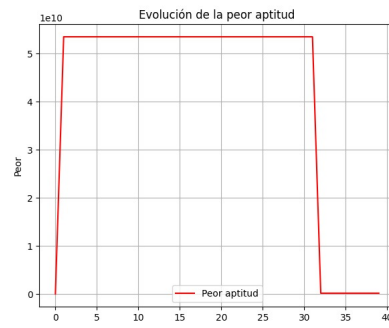
**Fig. 8.** Evolución de la aptitud promedio.

### Configuración 3

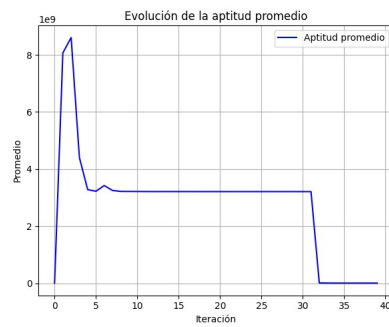
Población: 50 Iteraciones: 40  $G_0$ : 100 Alpha: 5



**Fig. 9.** Evolución de la mejor aptitud.



**Fig. 10.** Evolución de la peor aptitud.

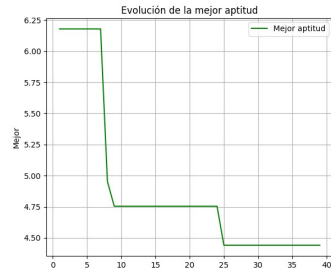


**Fig. 11.** Evolución de la aptitud promedio.

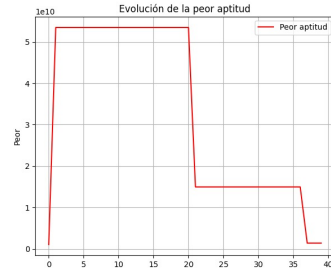


#### Configuración 4

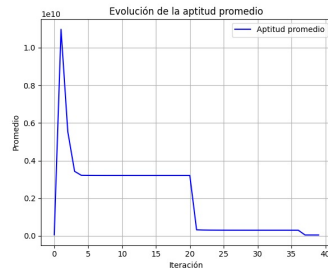
Población: 50 Iteraciones: 40  $G_0$ : 100 Alpha: 10



**Fig. 12.** Evolución de la mejor aptitud.



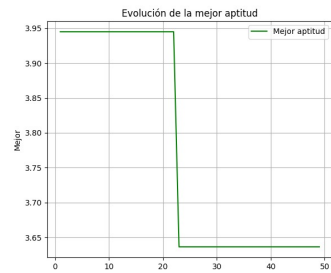
**Fig. 13.** Evolución de la peor aptitud.



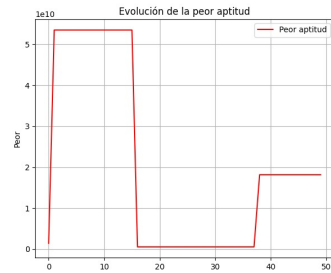
**Fig. 14.** Evolución de la aptitud promedio.

#### Configuración 5

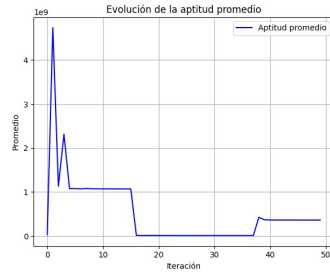
Población: 50 Iteraciones: 50  $G_0$ : 50 Alpha: 5



**Fig. 15.** Evolución de la mejor aptitud.



**Fig. 16.** Evolución de la peor aptitud.



**Fig. 17.** Evolución de la aptitud promedio.

Al observar los resultados promedio y mínimos obtenidos en las distintas configuraciones, se aprecia que la configuración 1 ( $G_0 = 50$ ,  $\alpha = 5$ ) produce los mejores resultados tanto en promedio como en la mejor ejecución individual, con un valor objetivo mínimo de aproximadamente **2.64** y una desviación estándar relativamente baja.

Por otro lado, configuraciones con valores altos de  $\alpha$  o  $G_0$  tienden a mostrar mayor variabilidad en los resultados (ver configuraciones 2 y 4), lo cual sugiere una menor estabilidad en la convergencia.

La visualización de la convergencia refuerza esta observación: las configuraciones con mejor desempeño tienden a estabilizarse rápidamente en torno a valores bajos del objetivo, mientras que otras exhiben oscilaciones o estancamiento.

En conjunto, estos experimentos permitieron identificar una combinación de hiperparámetros eficiente y estable para el algoritmo GSA aplicado a este problema, lo que respalda el uso de análisis estadístico (como ANOVA) para fundamentar decisiones sobre el ajuste fino del algoritmo.

## 5 Conclusiones

Finalmente, podemos concluir que el algoritmo de búsqueda gravitacional, implementado como solución al problema de la viga, fue una técnica de optimización eficaz para reducir el peso, a través de la simulación del comportamiento gravitacional entre agentes. A pesar de los retos que se presentaron al momento de su implementación, logró identificar configuraciones óptimas que cumplen el objetivo de minimizar el peso, respetando las restricciones del problema.

Tomando en cuenta que los parámetros son parte fundamental para lograr una solución eficiente, se generó un proceso de convergencia estable hacia un mínimo local, mostrando un equilibrio entre la explotación y la exploración de las soluciones con mejor fitness.

## 6 Trabajo a futuro

Nuestra implementación enfrentó varios retos que pudimos resolver a tiempo, sin embargo, para mejorar el comportamiento del algoritmo respecto a la función objetivo en un futuro se podrían implementar distintas penalizaciones y algoritmos. En el área de penalización lo ideal sería que nuestra constante se adapte según ciertos criterios, como el estancamiento, de esta manera sólo obtenemos una penalización brutal cuando sea necesario, donde podríamos incluso modificar aquellas soluciones cercanas al espectro factible, para que entren al rango y podamos obtener más soluciones para explotar. En cuanto a la adaptación de parámetros, podríamos aplicarlo con nuestra constante gravitacional y Alpha, esto con la técnica de recocido simulado, que nos permite tener un equilibrio entre exploración y explotación, teniendo resultados certeros, con los que podemos refinar las combinaciones exitosas para el algoritmo. Finalmente, sería interesante crear un híbrido con PSO, dado que la lógica es similar, pero de manera local para refinar los mejores individuos, para después aplicar GSA y profundizar la búsqueda, dada la naturaleza del algoritmo.

## 7 Referencias

1. S. Mirjalili and S. Z. M. Hashim, “A new hybrid psogsa algorithm for function optimization,” in *Proceedings of the 2010 International Conference on Computer and Information Application (ICCIA)*, Tianjin, China, 2010, pp. 374–377, accessed: 01-Jul-2025. [Online]. Available: <https://doi.org/10.1109/ICCIA.2010.6141614>
2. M. Khajezadeh and M. Eslami, “Gravitational search algorithm for optimization of retaining structures,” *Indian Journal of Science and Technology*, vol. 5, no. 1, pp. 1–7, 2012, accessed: 01-Jul-2025. [Online]. Available: <https://indjst.org/articles/gravitational-search-algorithm-for-optimization-of-retaining-structures>
3. E. Corp. and MIT. (2025) Example 3: Welded-beam design. Accessed: 2025-07-01. [Online]. Available: <https://neorl.readthedocs.io/en/latest/examples/ex3.html>
4. T. MathWorks. (2025) Design optimization of a welded beam. Accessed: 01-Jul-2025. [Online]. Available: <https://la.mathworks.com/help/gads/multiobjective-optimization-welded-beam.html>
5. A. A. H. Alkurdi, “Optimization of welded beam design problem using water evaporation optimization algorithm,” *Academic Journal of Nawroz University*, vol. 12, no. 3, pp. 499–507, 2023. [Online]. Available: <https://doi.org/10.25007/ajnu.v12n3a1753>
6. C. N. D. David and S. E. A. Stephen, “Cost minimization of welded beam design problem using nontraditional optimization through matlab and validation through ansys simulation,” *International Journal of Mechanical Engineering and Technology*, vol. 9, no. 8, pp. 180–192, 2018, article ID: IJMET\_09\_08\_020. [Online]. Available: [https://iaeme.com/MasterAdmin/Journal\\_uploads/IJMET/VOLUME9\\_ISSUE8/IJMET0908020.pdf](https://iaeme.com/MasterAdmin/Journal_uploads/IJMET/VOLUME9_ISSUE8/IJMET0908020.pdf)
7. E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, “Gsa: A gravitational search algorithm,” *Information Sciences*, vol. 179, pp. 2232–2248, 2009. [Online]. Available: <https://doi.org/10.1016/j.ins.2009.03.004>