

UNIVERSIDAD POLITÉCNICA DE MADRID

DEPARTAMENTO DE INTELIGENCIA ARTIFICIAL



UNIVERSIDAD
POLITÉCNICA
DE MADRID



Búsqueda Inteligente Basada en Metaheurísticas

Recocido Simulado aplicado al problema de las N-Reinas

Fragua Baeza, Ángel

Jiménez López de Castro, Joaquín

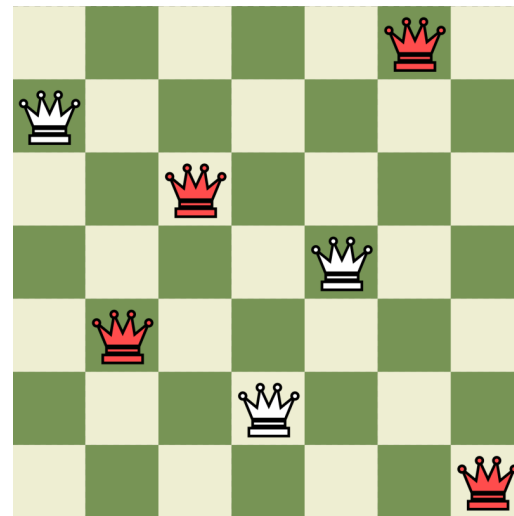
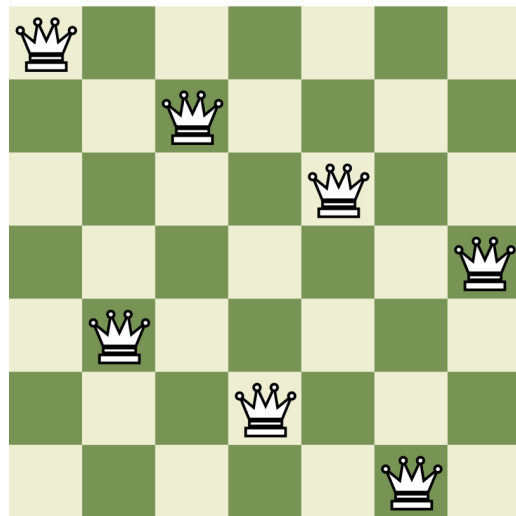
Solís García, Javier

Índice

- Problema de las N-Reinas
- Recocido Simulado
- Implementación
- Resultados
- Conclusiones
- Trabajo Futuro

Problema de las N-Reinas

- El problema para $N=8$ fue inicialmente propuesto por Bezzel (1848).
- Consiste en colocar N reinas en un tablero de ajedrez $N \times N$, cuyas reinas deben estar colocadas de tal forma que ninguna se pueda capturar entre sí.
- Una reina puede capturar a otra si están en la misma fila, columna o diagonal.



Recocido Simulado Kirkpatrick et al. (1983)

- Variante del **método de máximo descenso** que intenta evitar los óptimos locales.
- Al igual que en máximo descenso, se genera una solución aleatoria del entorno de la solución actual y **si esta es mejor** pasa a ser la actual.
- Se escogen **soluciones peores** que la actual con una probabilidad que decrece con respecto al número de iteraciones transcurridas (y por tanto la temperatura) y el empeoramiento que supone la nueva solución.

Decisiones			
Genéricas		Específicas	
T_0 :	Temperatura Inicial	x_0 :	Solución Inicial
T_k :	Función Temperatura		Generación de entornos
$p(k)$:	Aceptación		Evaluación de Δf_{ij}
	Criterio de Parada		

Implementación. Framework

- Hemos partido de la librería Python **simanneal** (<https://github.com/perrygeo/simanneal>)
- Se ha adaptado el código para:
 - Una codificación específica del problema
 - Representación gráfica de la evolución de los parámetros más relevantes
 - Representación gráfica de la solución o tablero
 - Programa interactivo mediante ipywidgets

Implementación. Decisiones Genéricas

Decisiones			
Genéricas		Específicas	
T_0 :	Temperatura Inicial	x_0 :	Solución Inicial
T_k :	Función Temperatura		Generación de entornos
$p(k)$:	Aceptación		Evaluación de Δf_{ij}
	Criterio de Parada		

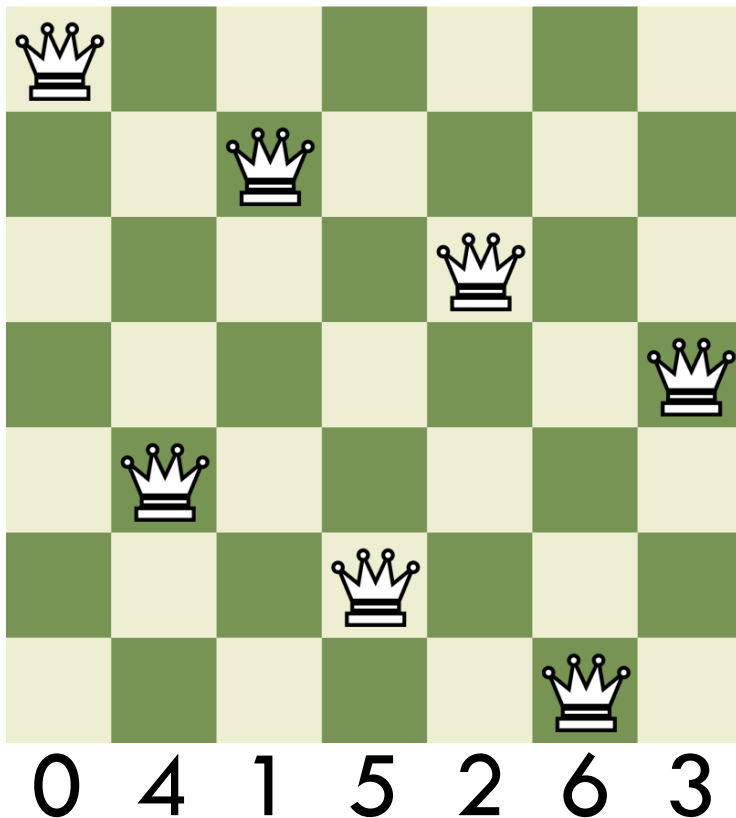
El framework define los siguientes hiperparámetros:

- **Tmax**: Temperatura máxima
- **Tmin**: Temperatura mínima
- **steps**: n° de iteraciones total

- $T_0 = \mathbf{Tmax}$
- $T_k = \alpha^k T_0$ (Recocido Geométrico)
 - k : n° de iteraciones ($0 \leq k \leq \mathbf{steps}$)
 - $\alpha^{\mathbf{steps}} \cdot \mathbf{Tmax} = \mathbf{Tmin}$ (Decrecimiento exponencial de la Temperatura)
- $p(k) = \min\{1, \exp(-\frac{f(y) - f(x_k)}{T_k})\}$
- Criterio de parada:
 - $k = \mathbf{steps}$
 - Obtener solución óptima (especificado por el usuario)

Implementación. Decisiones Específicas

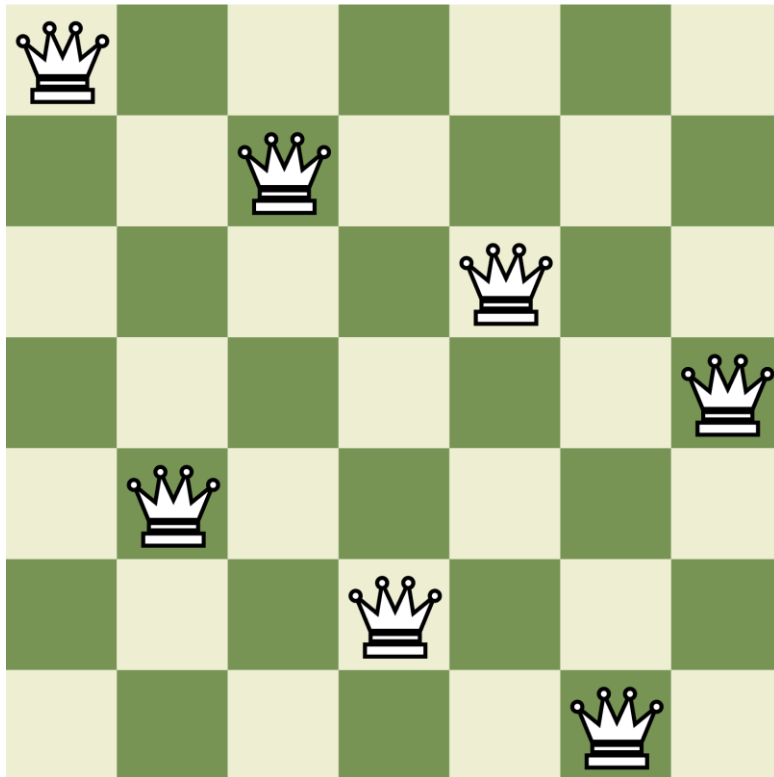
Decisiones			
Genéricas		Específicas	
T_0 :	Temperatura Inicial	x_0 :	Solución Inicial
T_k :	Función Temperatura		Generación de entornos
$p(k)$:	Aceptación		Evaluación de Δf_{ij}
	Criterio de Parada		



- Array unidimensional de tamaño N cuya posición **i -ésima** indica la fila de la reina en la columna i .
- Ejemplo: $[0, 4, 1, 5, 2, 6, 3]$
- Otras representaciones serían posibles como una matriz binaria $N \times N$, pero introduciría la posibilidad de generar soluciones no válidas.

Implementación. Decisiones Específicas

Decisiones			
Genéricas		Específicas	
T_0 :	Temperatura Inicial	x_0 :	Solución Inicial
T_k :	Función Temperatura		Generación de entornos
$p(k)$:	Aceptación		Evaluación de Δf_{ij}
	Criterio de Parada		



A. Se intercambian los valores de dos posiciones aleatorias del array.

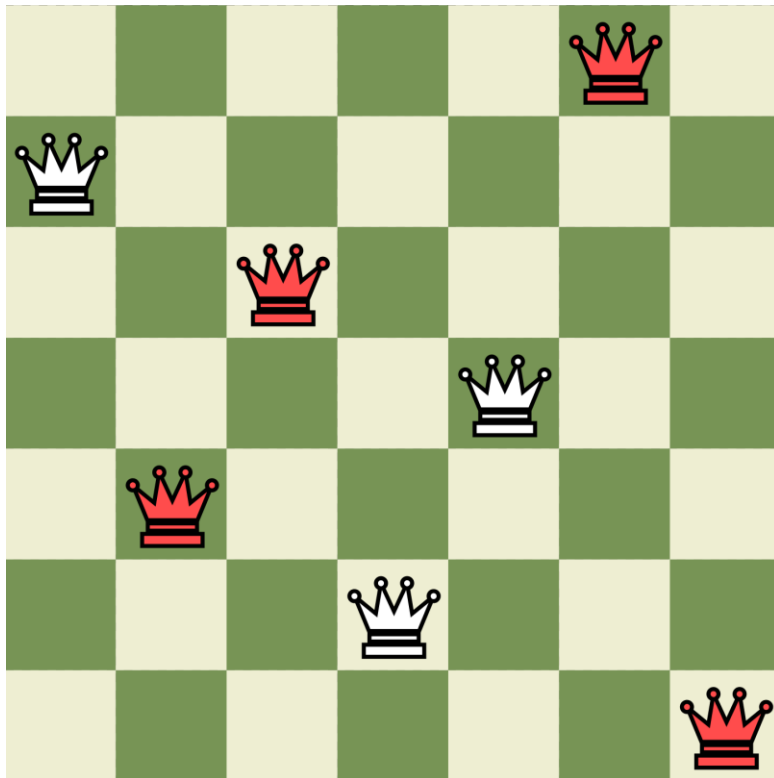
- Requiere que el estado inicial sea un reordenamiento de la secuencia $[0,1,\dots,N]$.
- Evita colisiones en filas y columnas. (Solo hay colisiones en diagonales)

B. Se escoge una reina aleatoriamente y se mueve a una fila aleatoria dentro de su misma columna.

C. Se escoge una reina aleatoria y se mueve una fila hacia arriba o hacia abajo.

Implementación. Decisiones Específicas

Decisiones			
Genéricas		Específicas	
T_0 :	Temperatura Inicial	x_0 :	Solución Inicial
T_k :	Función Temperatura		Generación de entornos
$p(k)$:	Aceptación		Evaluación de Δf_{ij}
	Criterio de Parada		

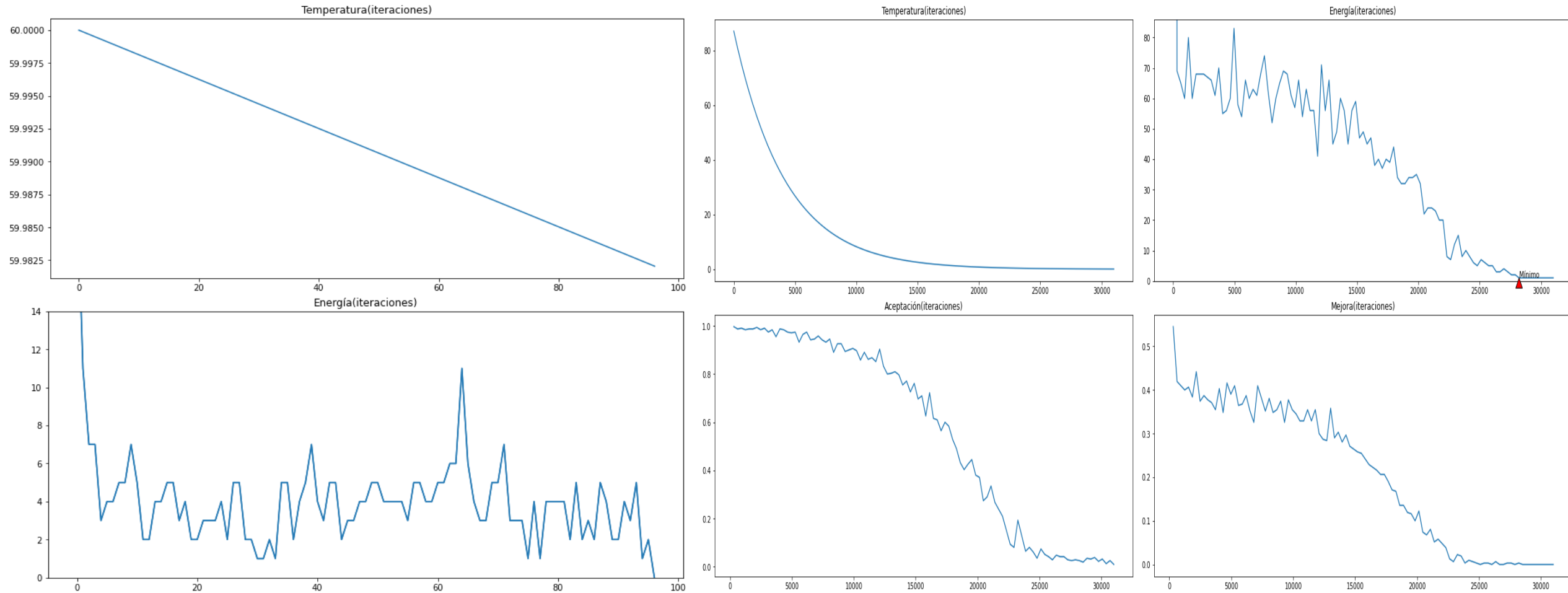


- Cada solución es evaluada en relación al número de colisiones únicas que hay entre sus reinas. Ejemplo: 2 colisiones.
- El objetivo es obtener cero colisiones. (Una de las condiciones de parada)

Implementación. Estimación mejores parámetros

- El framework cuenta con una función de estimación llamada **auto**.
- Ejecuta el método de recocido simulado múltiples veces:
 1. Tmax con una aceptación de soluciones cercana al 98%
 2. Tmin con una mejora de soluciones del 0%
- Estima el número de iteraciones máximas para que el tiempo de ejecución del algoritmo sea menor al establecido. Las pruebas las hemos realizado con tiempo máximo de **2 minutos**.

Resultados. Evolución del algoritmo



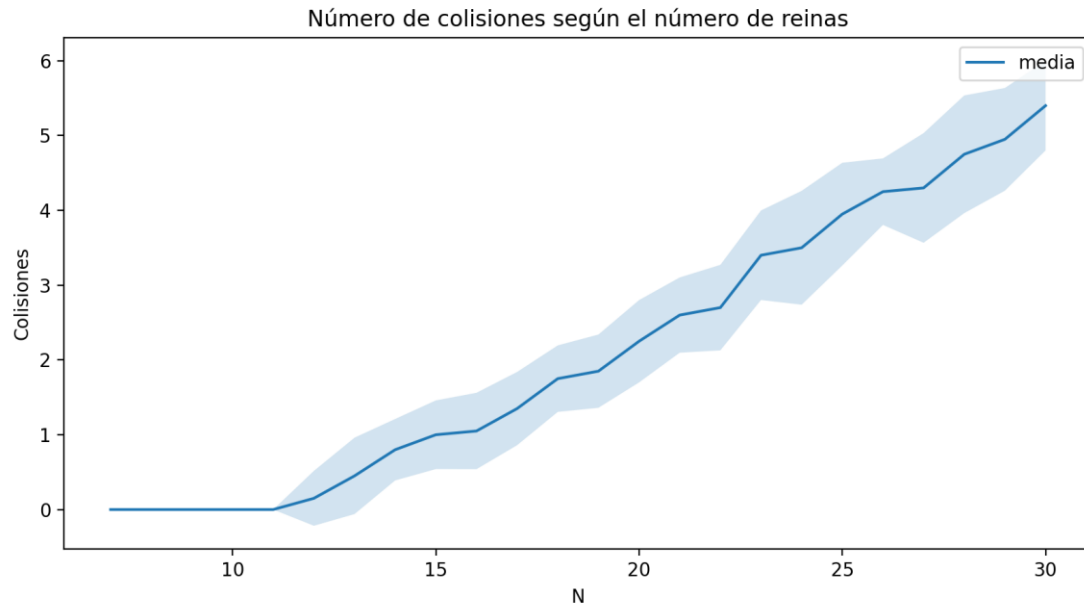
Hiperparámetros estimados por **auto** (N=7):

- Tmax = 60
- Tmin = 0.3
- steps = 1700000

Hiperparámetros estimados por **auto** (N=100):

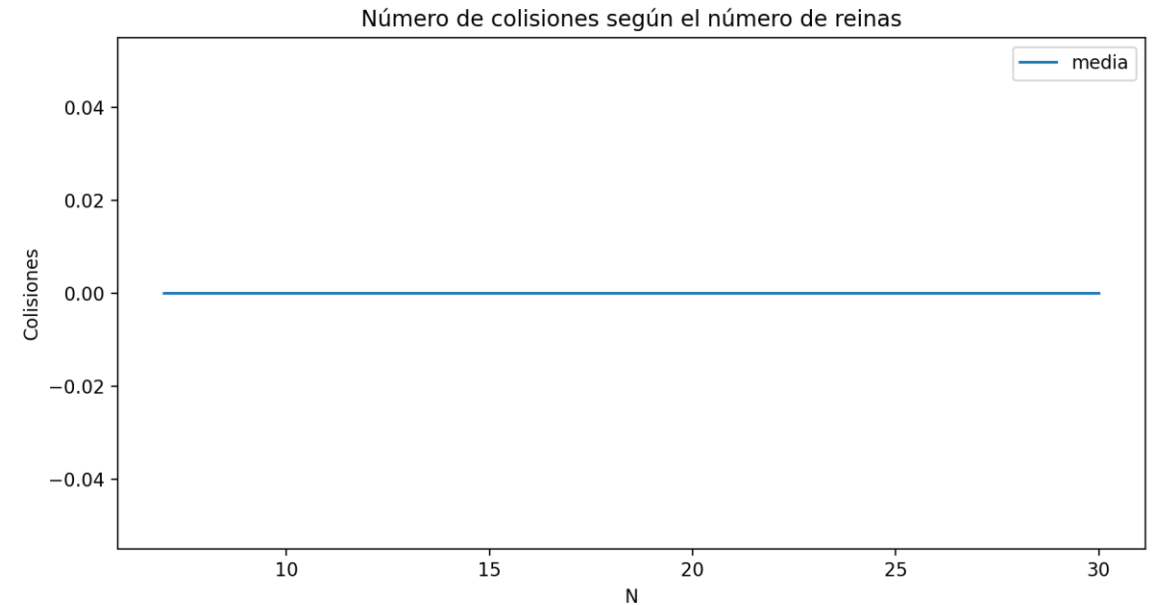
- Tmax = 50
- Tmin = 0.058
- steps = 31000

Resultados. Comparación de colisiones



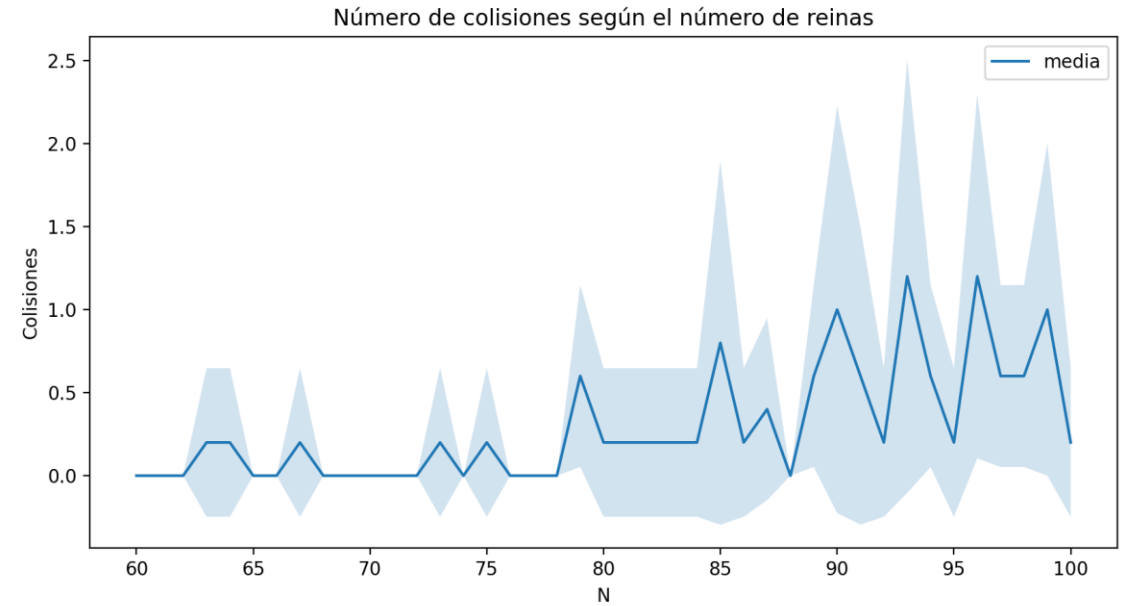
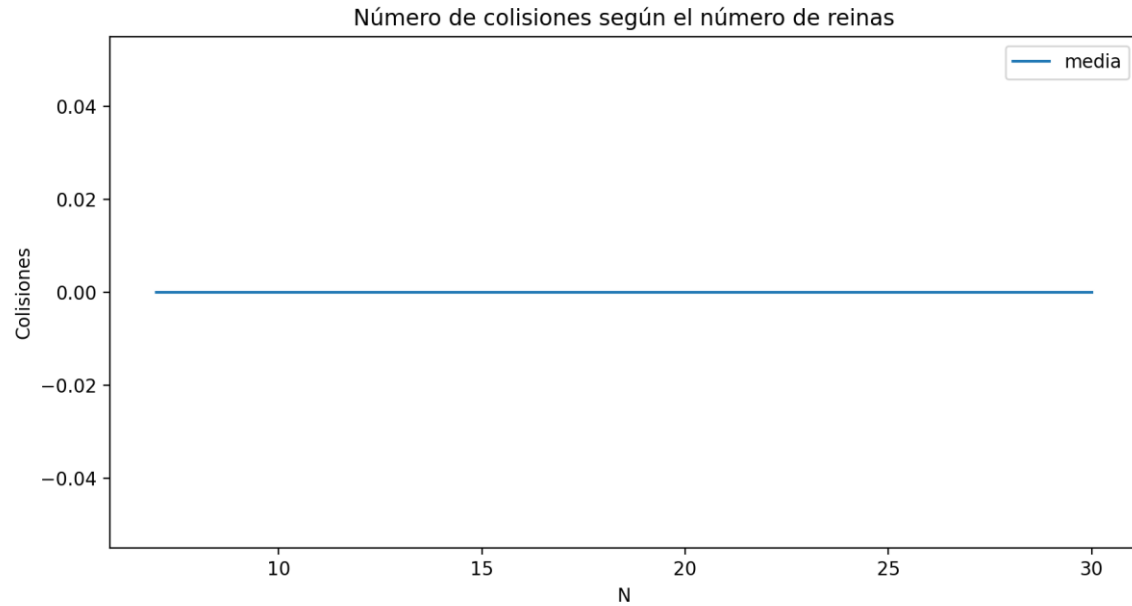
Hiperparámetros por defecto:

- Tmax = 25000
- Tmin = 2.5
- steps = 50000



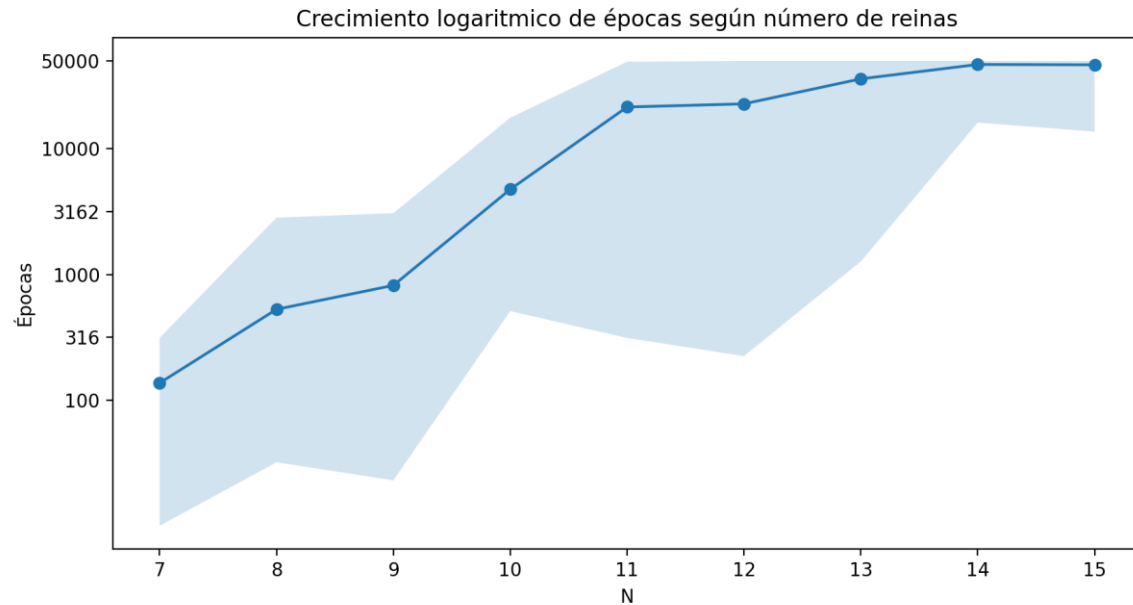
Hiperparámetros establecidos por el estimador *auto*

Resultados. Evolución de colisiones



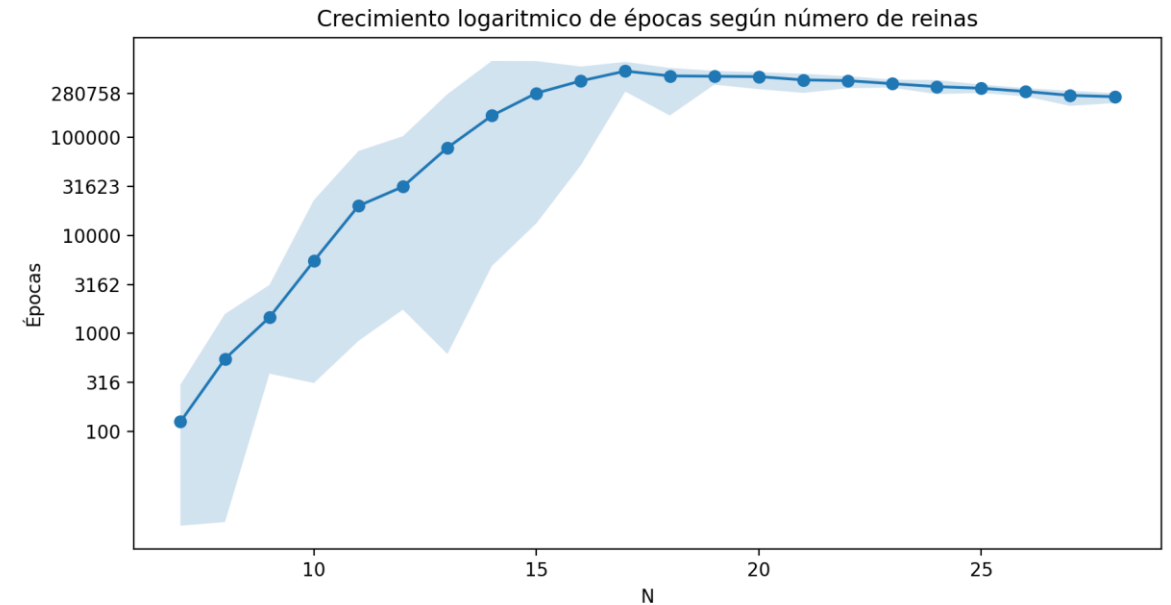
Hiperparámetros establecidos por el estimador *auto*

Resultados. Comparación de épocas



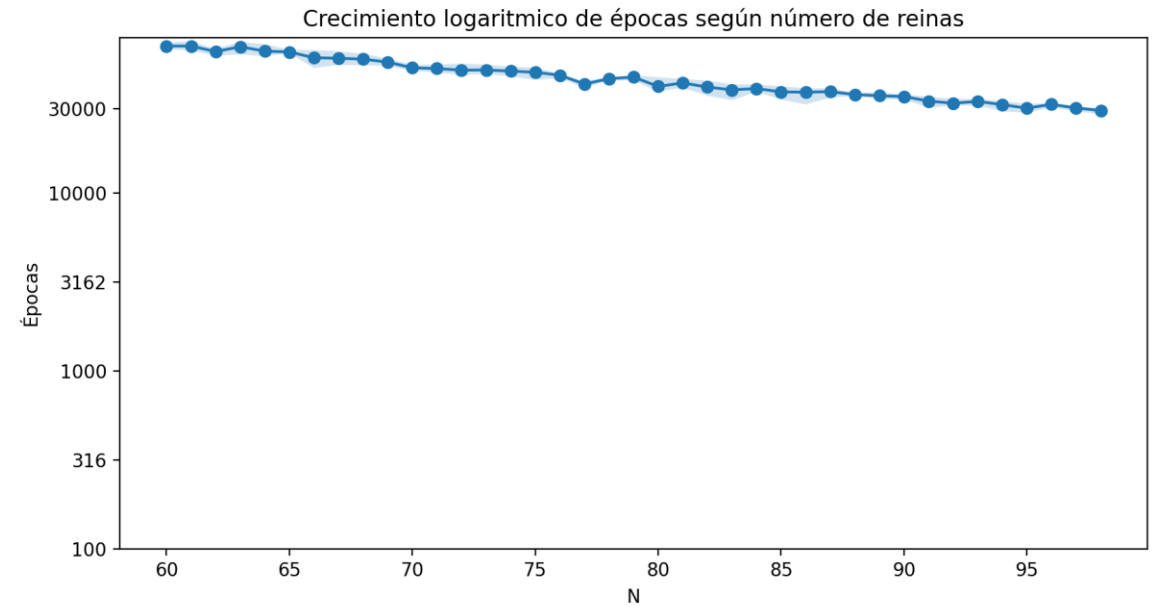
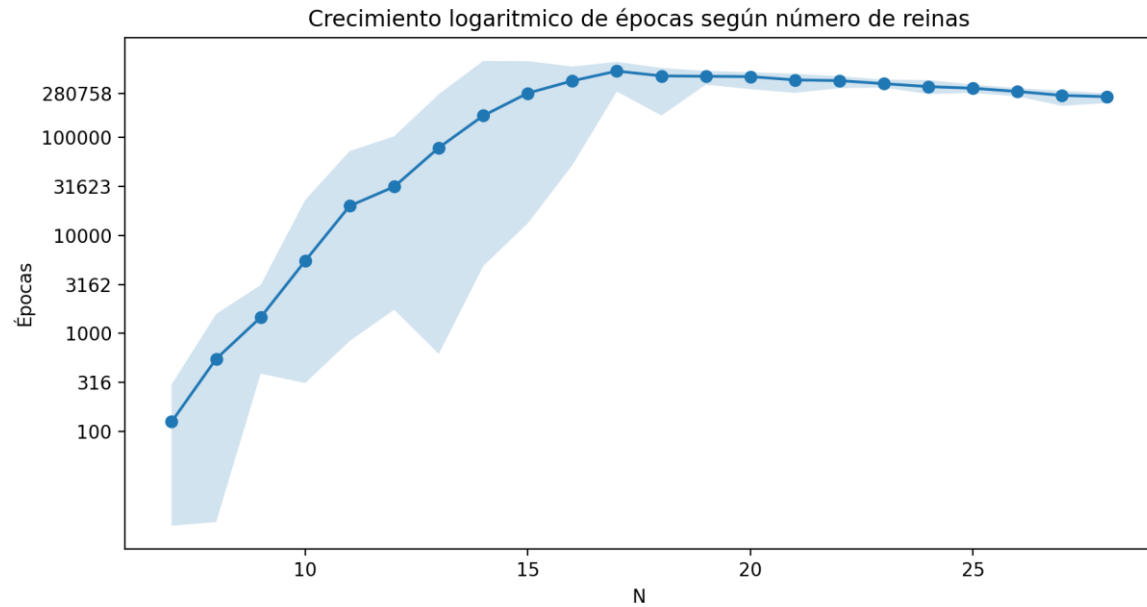
Hiperparámetros por defecto:

- Tmax = 25000
- Tmin = 2.5
- steps = 50000



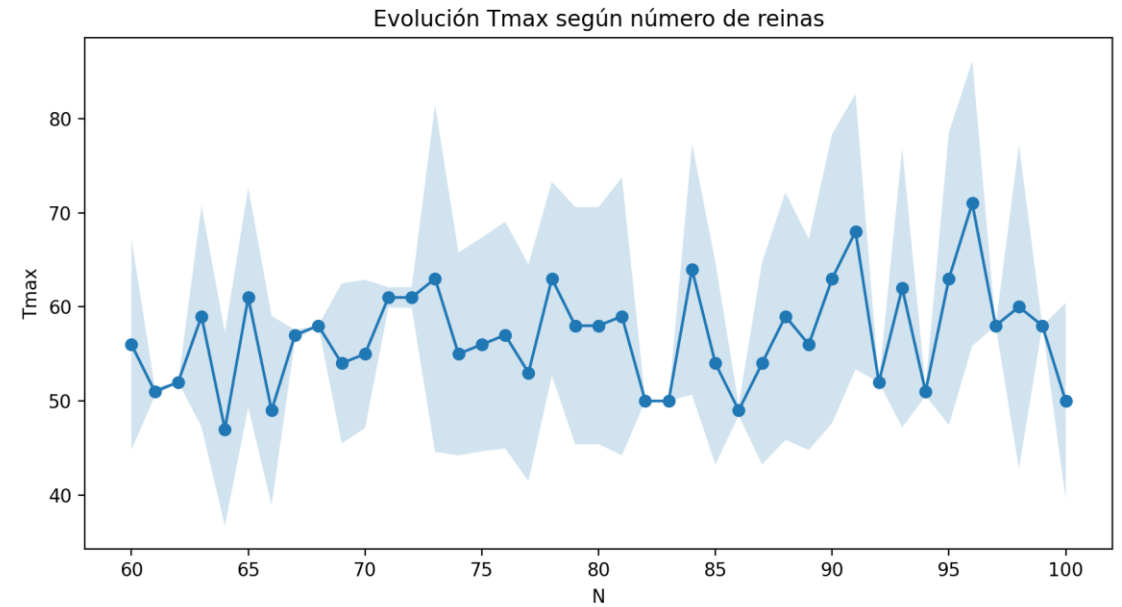
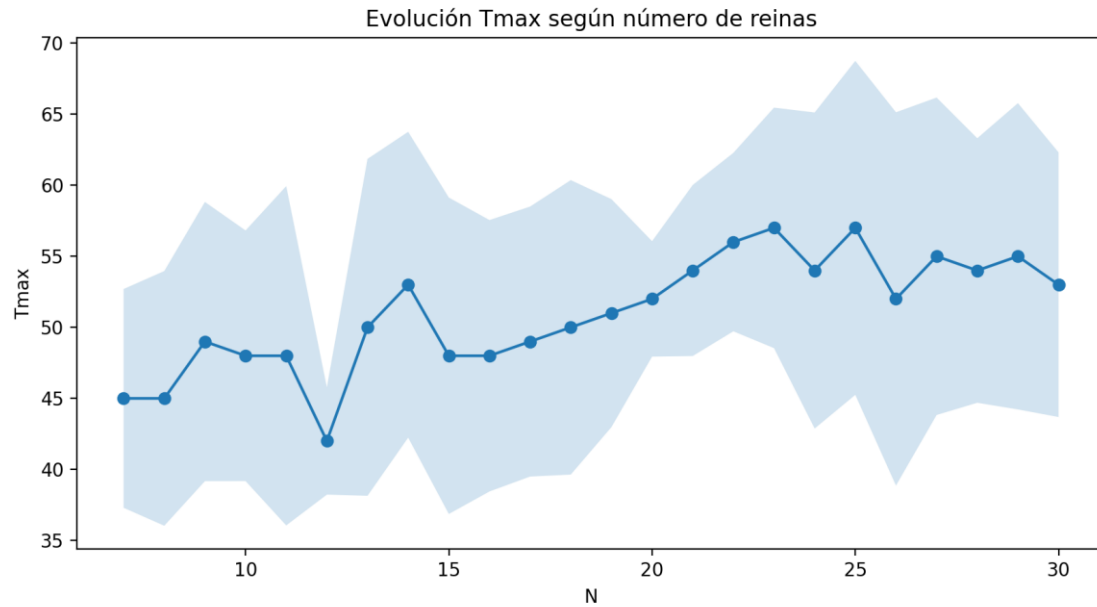
Hiperparámetros establecidos por el estimador *auto*

Resultados. Evolución de épocas



Hiperparámetros establecidos por el estimador *auto*

Resultados. Evolución de Tmax

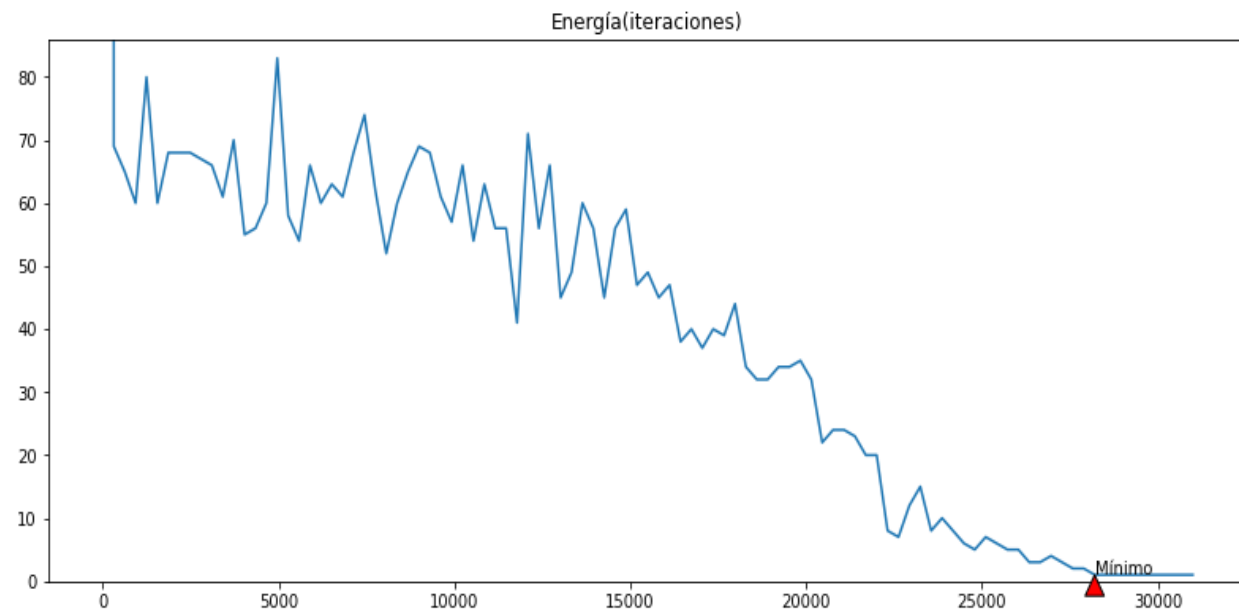


Hiperparámetros establecidos por el estimador *auto*

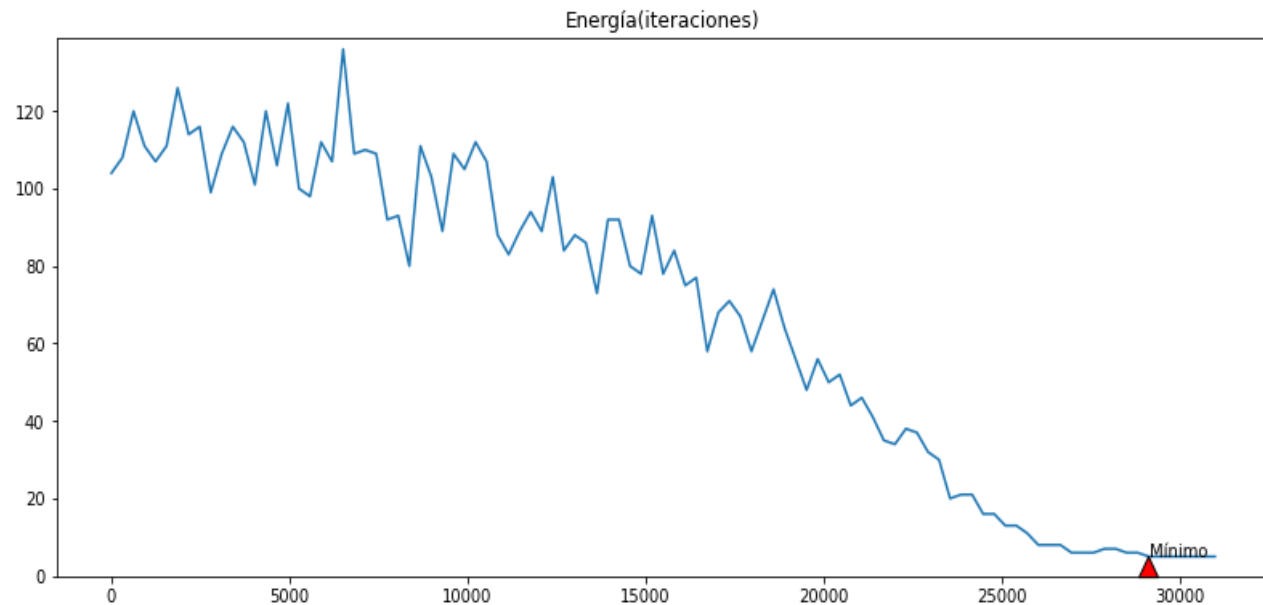
Resultados. Comparación de codificaciones

Hiperparámetros establecidos por el estimador *auto*
N = 100

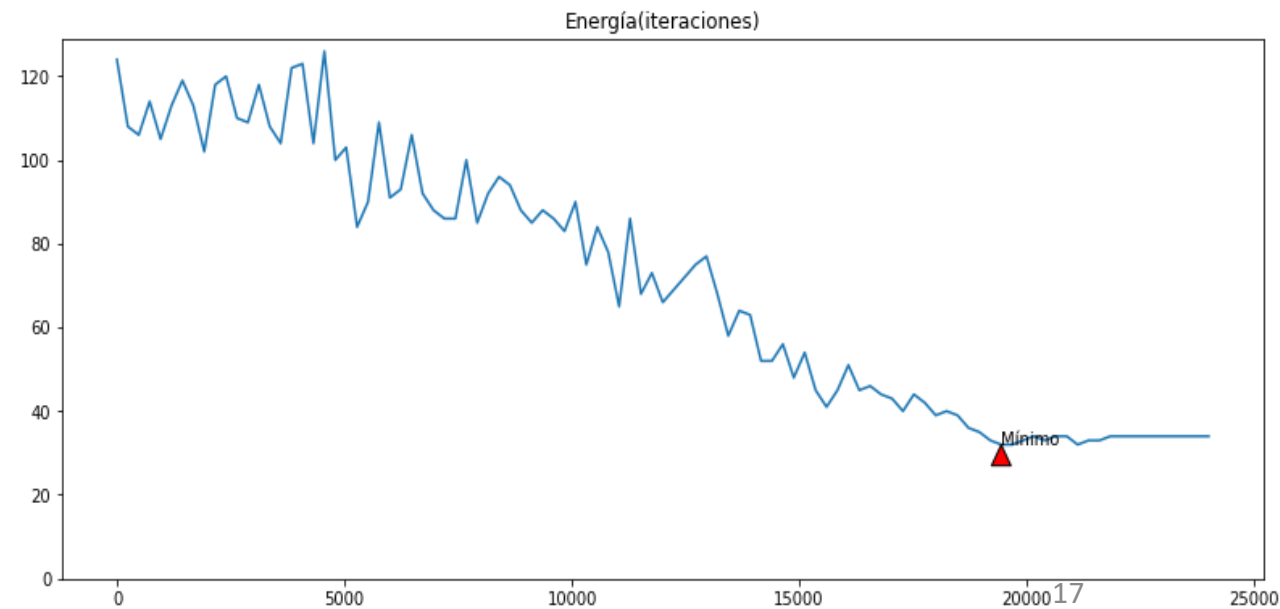
Codificación A



Codificación B



Codificación C

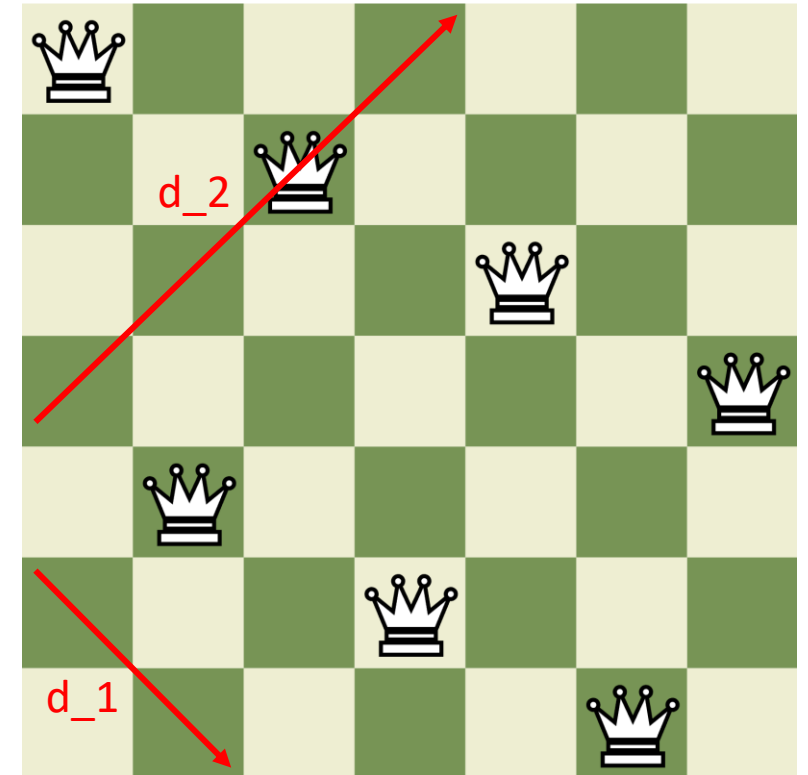


Conclusiones

- La codificación del problema afecta significativamente al rendimiento.
- La elección de la función de evaluación afecta a la elección de la temperatura.
- La metaheurística es muy sensible a la elección de los hiperparámetros.
- El problema de las 7-Reinas es fácilmente resoluble con cualquier configuración, mientras que para un número de reinas mayor, es imprescindible la realización de un ajuste inicial.

Trabajo Futuro

- Coste actual de evaluación es $O(N^2)$
 - Para cada reina evaluar sus colisiones siguientes
- Se podría conseguir coste $O(N)$
 - Tablero $N \times N$ tiene $2(2N - 1)$ diagonales
 - Calcular las colisiones por diagonal
 - Detalles específicos de codificación:
 - $d_1[2N-1]=\{-1,..,-1\}$, $d_2[2N-1]=\{-1,..,-1\}$
 - for r in solución:
 - $d_1[r.i - r.i] += 1$
 - $d_2[r.i - r.j] += 1$
 - colisiones = valores de d_1 y d_2 mayores que 0



Referencias

- M. Bezzel, Proposal of 8-queens problem, Berliner Schachzeitung 3 (1848), 363. (Submitted under the author name "Schachfreund".)
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. science, 220(4598), 671-680.

MUCHAS GRACIAS POR SU ATENCIÓN