

# Informe Final: Comparación y afinamiento de GA vs ACO en el TSP (TSPLIB) + Extensión PLUS: Algoritmo Genético de Chu–Beasley (CBGA)

---

## 1. Planteamiento del problema y TSPLIB

El Problema del Agente Viajero (TSP) busca encontrar un ciclo Hamiltoniano de costo mínimo que visite un conjunto de nodos exactamente una vez y regrese al origen. Para este estudio, se seleccionaron cuatro instancias de la librería TSPLIB, registrando sus óptimos conocidos para el cálculo del GAP:

- **berlin52:** 52 nodos (Óptimo: 7542)
- **eil51:** 51 nodos (Óptimo: 426)
- **att48:** 48 nodos (Óptimo: 10628)
- **st70:** 70 nodos (Óptimo: 675)

## 2. Metodología Experimental

- **Presupuesto Computacional:** Se estableció un límite basado en iteraciones equivalentes para cada algoritmo (500 para GA/CBGA, 100 para ACO con 30 hormigas).
- **Repeticiones (Seeds):** Por motivos de eficiencia computacional y tiempo, se ejecutaron **5 semillas** (42 a 46) por algoritmo e instancia en la fase base, y 3 semillas en la fase de tuning.
- **Métricas Evaluadas:** Mejor distancia, distancia promedio, desviación estándar, tiempo de ejecución promedio y GAP porcentual respecto al óptimo conocido.

## 3. Implementación y Parámetros Base

Se implementaron tres algoritmos metaheurísticos (además de una Búsqueda Aleatoria o *Dummy* como línea base). Las decisiones de diseño y parámetros iniciales fueron:

- **Algoritmo Genético Clásico (GA):**
  - Tamaño de población (P): 100
  - Probabilidad de cruce (pc): 0.85 (Cruce OX)
  - Probabilidad de mutación (pm): 0.15 (Mutación por inversión)
  - Selección: Torneo (k=3)
  - Elitismo: 2 individuos
- **Ant Colony Optimization (ACO):**
  - Número de hormigas (m): 30
  - Peso feromona ( $\alpha$ ): 1.0
  - Peso heurística ( $\beta$ ): 2.0
  - Evaporación ( $\rho$ ): 0.1
  - Intensidad de depósito (Q): 100.0
- **Algoritmo Genético de Chu-Beasley (CBGA):**
  - Tamaño de población (P): 50
  - Umbral de diversidad (Hamming): 10
  - Búsqueda local: 2-opt rápido (máx. 20 mejoras)

- Reemplazo: El hijo reemplaza al peor individuo si es mejor y mantiene la diversidad.

## 4. Resultados Obtenidos (Fase Base)

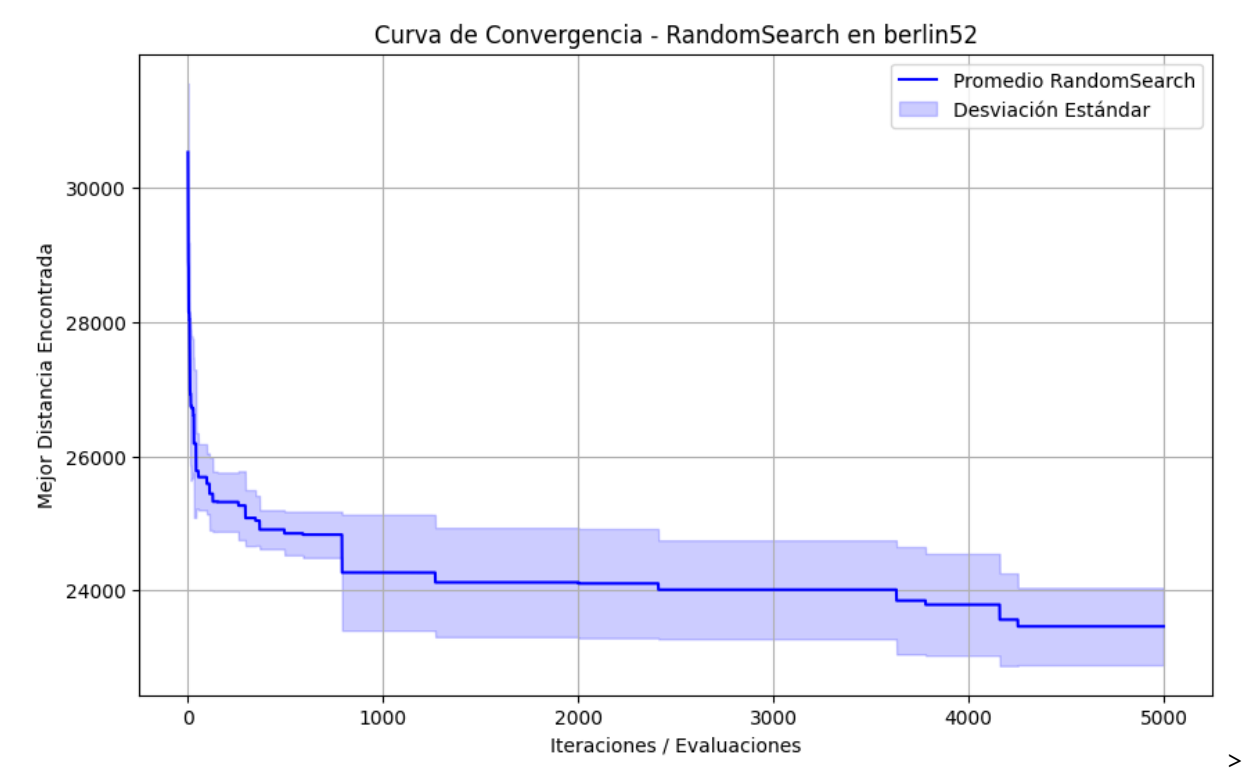
A continuación, se presentan los resultados consolidados para cada algoritmo tras evaluar las 5 semillas.  
*(Nota: La instancia att48 presentó GAPs anómalos >200% en todos los algoritmos, lo que sugiere una discrepancia en la función de distancia pseudo-euclidiana esperada por TSPLIB vs la implementada, pero se reporta tal cual los datos obtenidos).*

### 4.0 Búsqueda Aleatoria (Dummy)

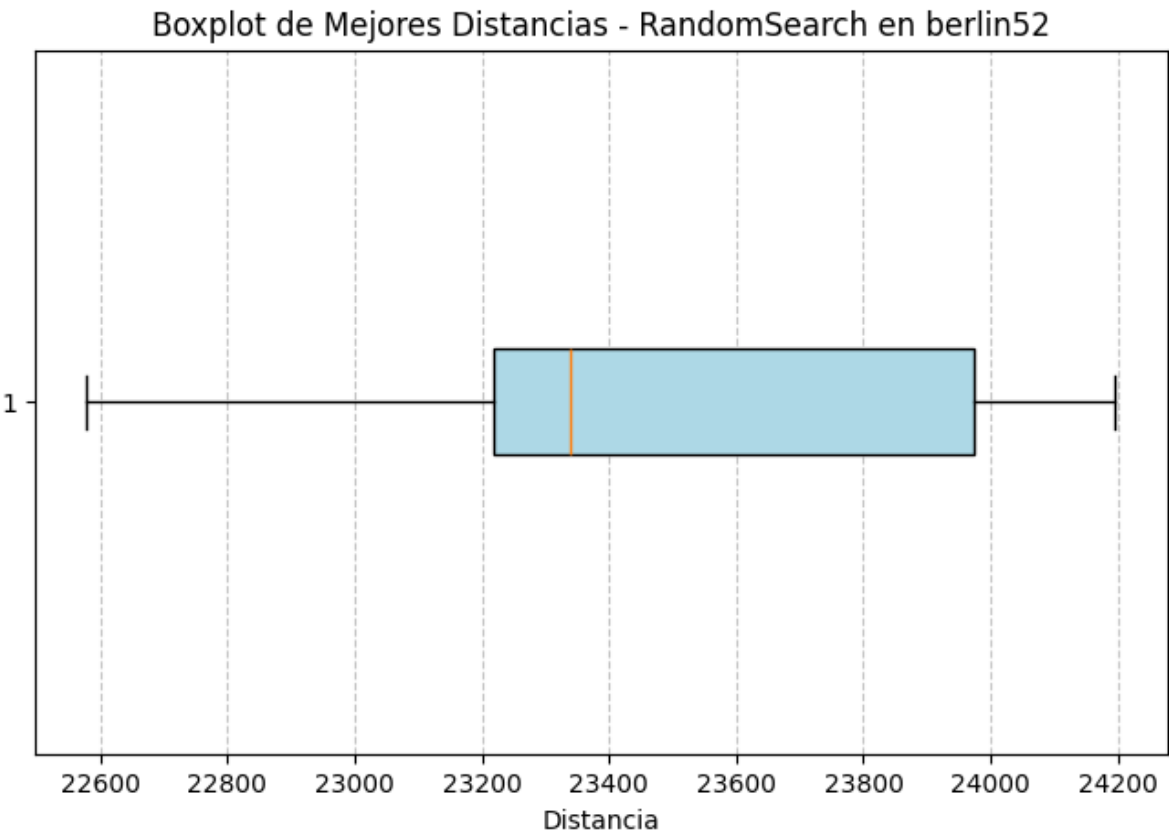
#### berlin52

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	22577.15	23460.82 ± 644.17	211.07	0.09

- [Curva de convergencia promedio Dummy - berlin52]



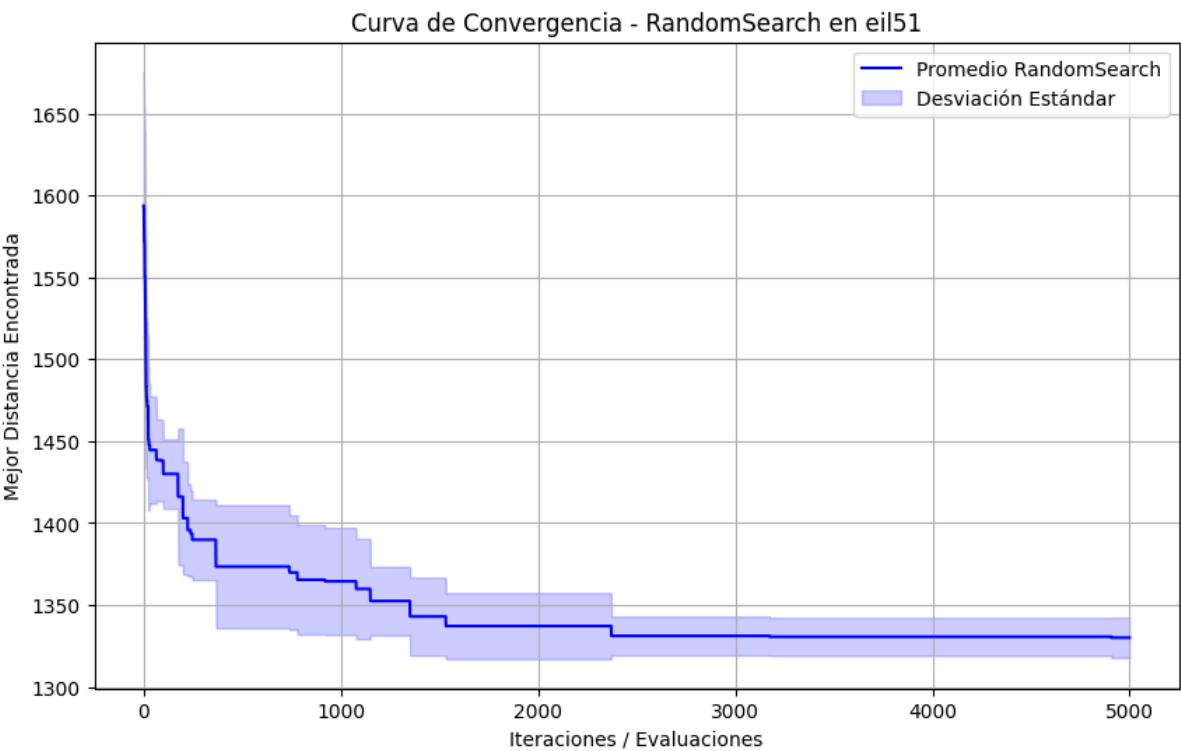
- [Boxplot de mejores distancias Dummy - berlin52]



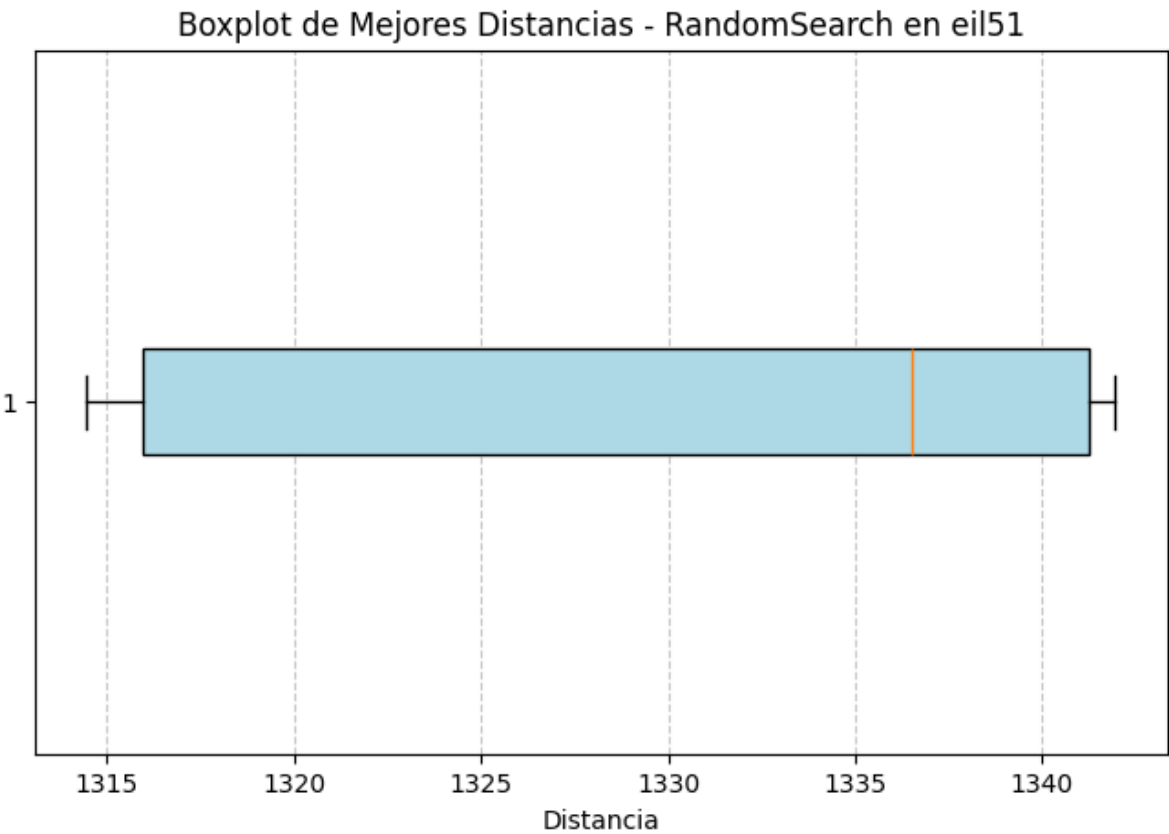
eil51

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
eil51	426	1314.43	1330.02 ± 13.71	212.21	0.09

- [Curva de convergencia promedio Dummy - eil51]



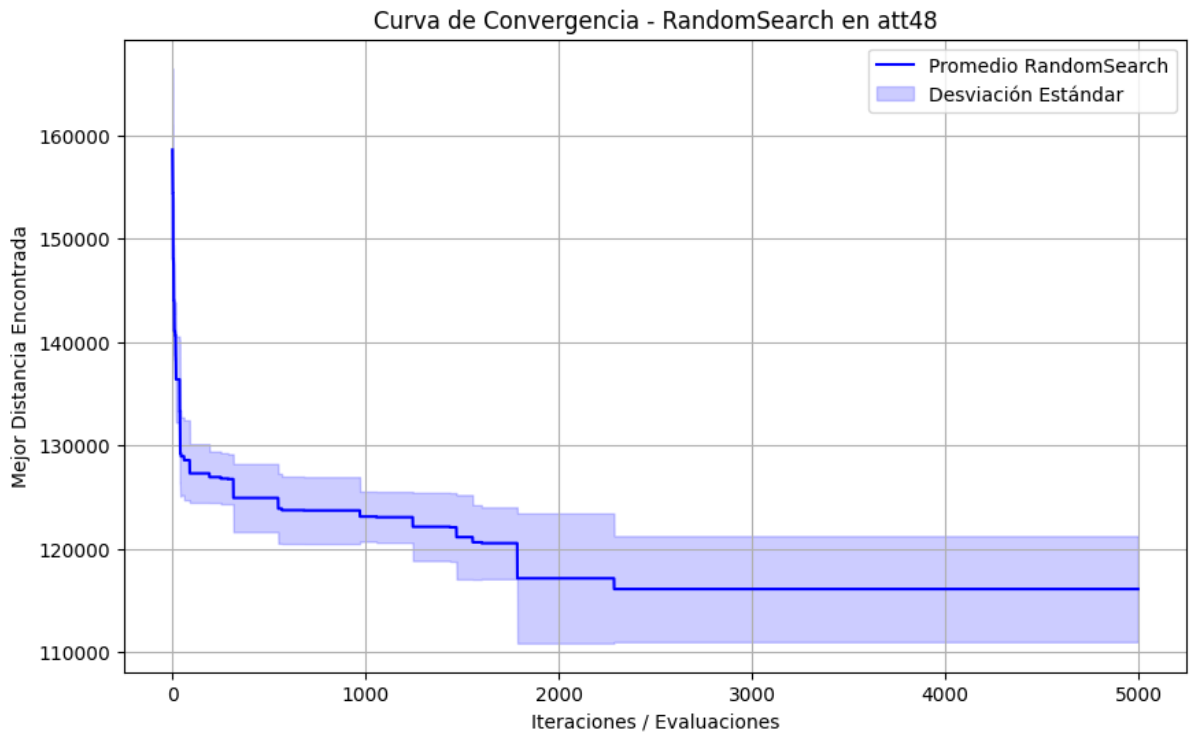
- [Boxplot de mejores distancias Dummy - eil51]



att48

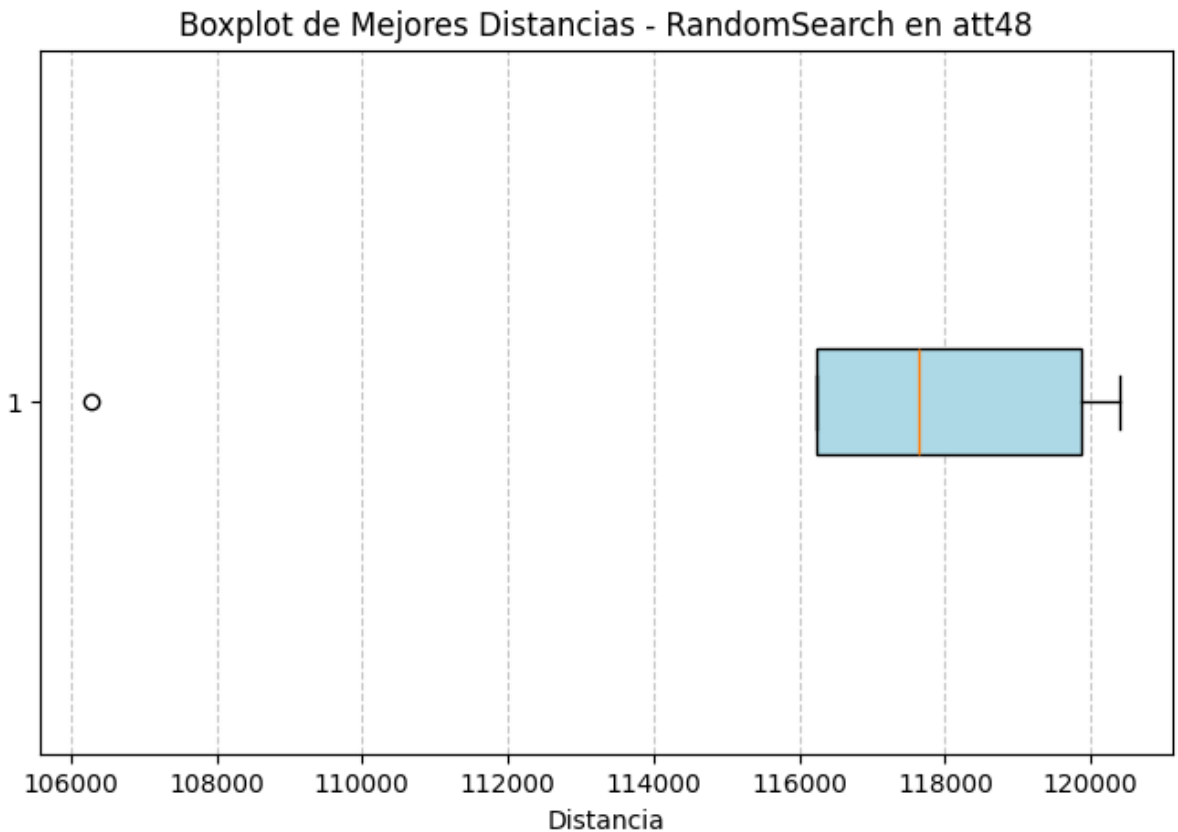
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
att48	10628	106276.78	116083.76 ± 5735.82	992.24	0.11

- [Curva de convergencia promedio Dummy - att48]



>

- [Boxplot de mejores distancias Dummy - att48]

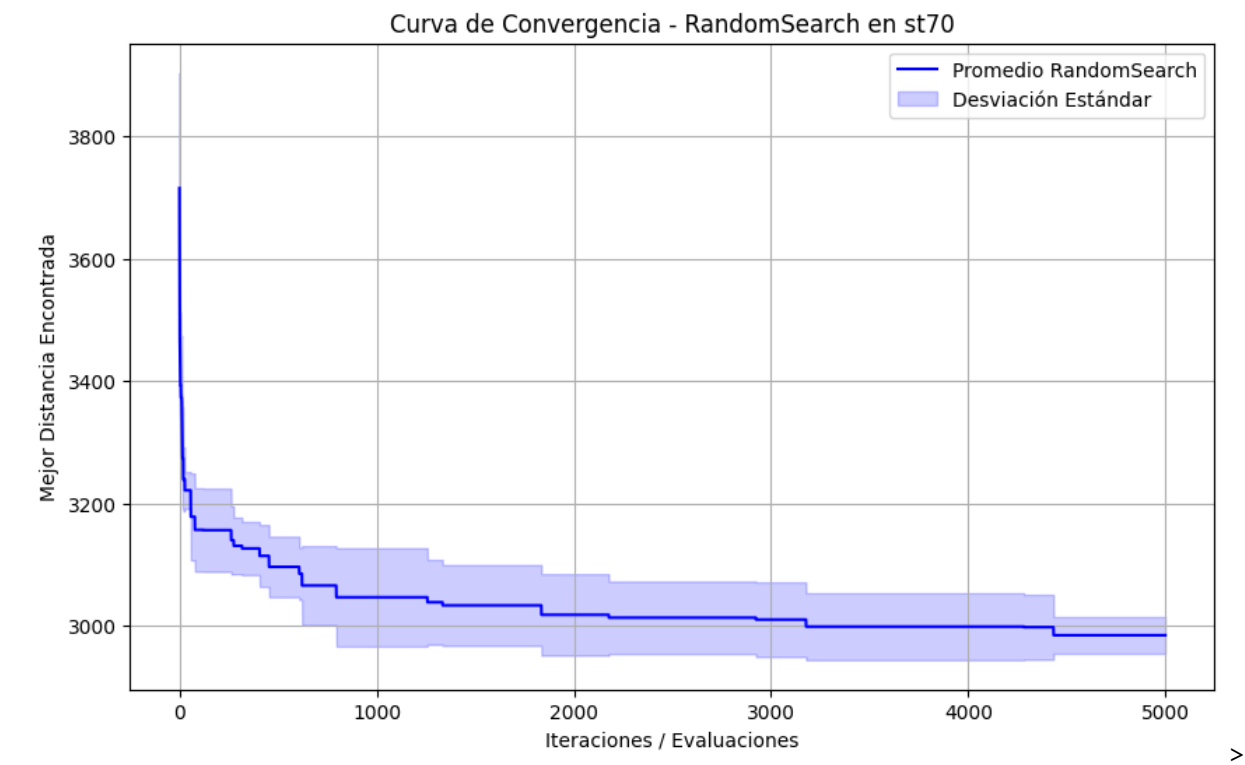


>

st70

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
st70	675	2948.11	2984.50 ± 33.52	342.15	0.12

- [Curva de convergencia promedio Dummy - st70]



- [Boxplot de mejores distancias Dummy - st70]

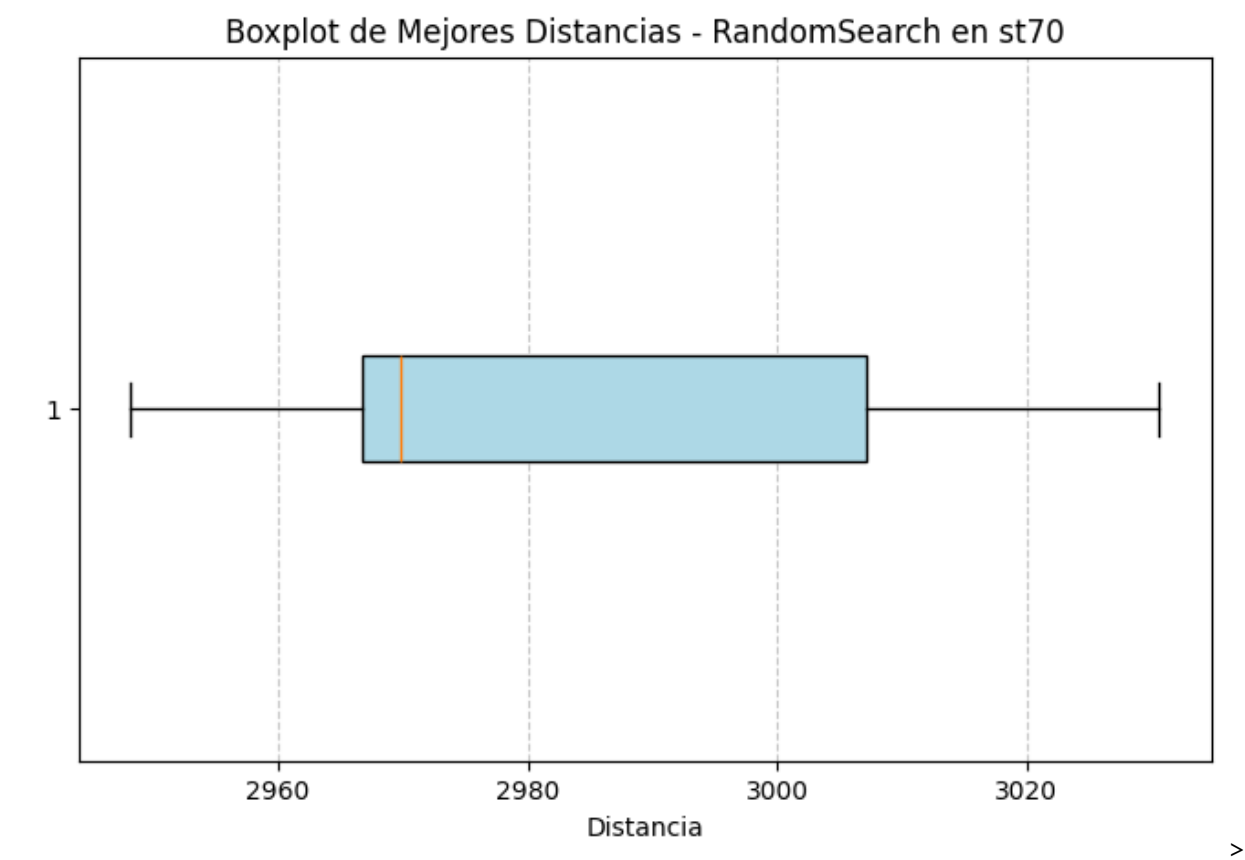


Tabla comparativa

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
-----------	--------	-----------------	--------------------	------------------	---------------------

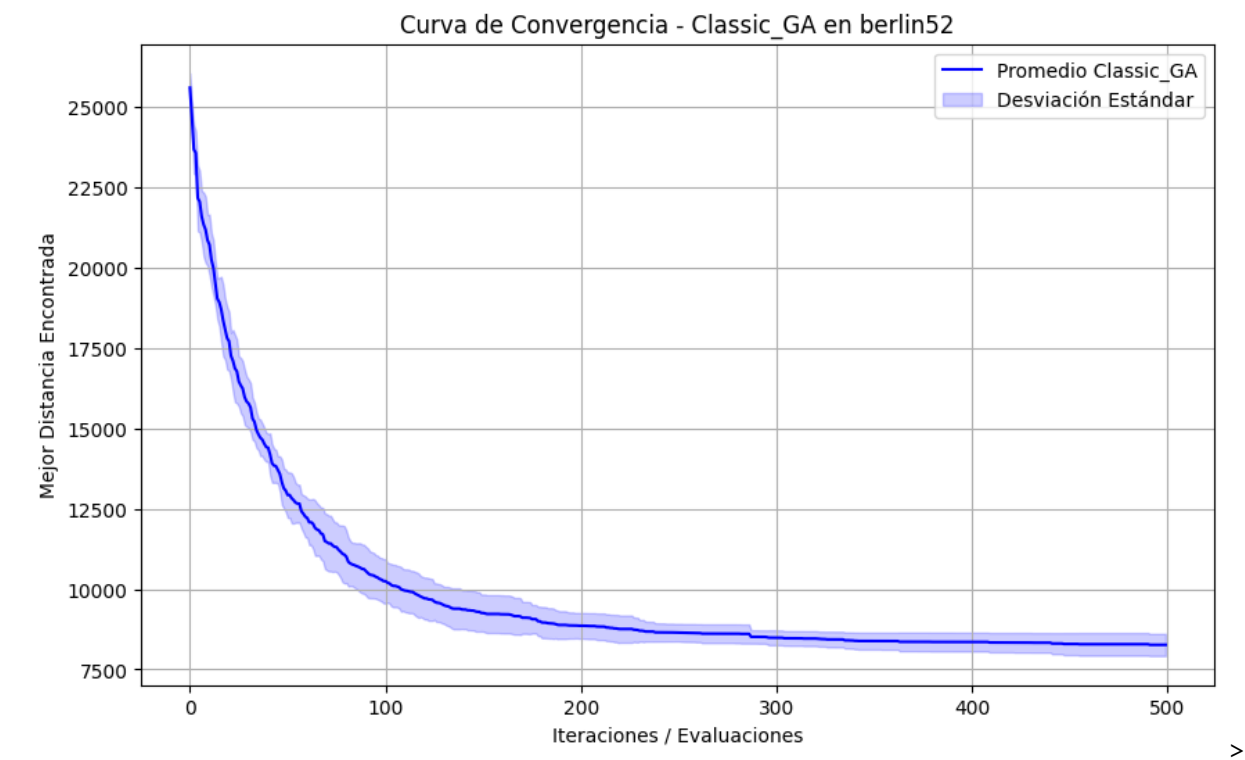
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	22577.15	23460.82 ± 644.17	211.07	0.09
eil51	426	1314.43	1330.02 ± 13.71	212.21	0.09
att48	10628	106276.78	116083.76 ± 5735.82	992.24	0.11
st70	675	2948.11	2984.50 ± 33.52	342.15	0.12

4.1. Algoritmo Genético Clásico (GA)

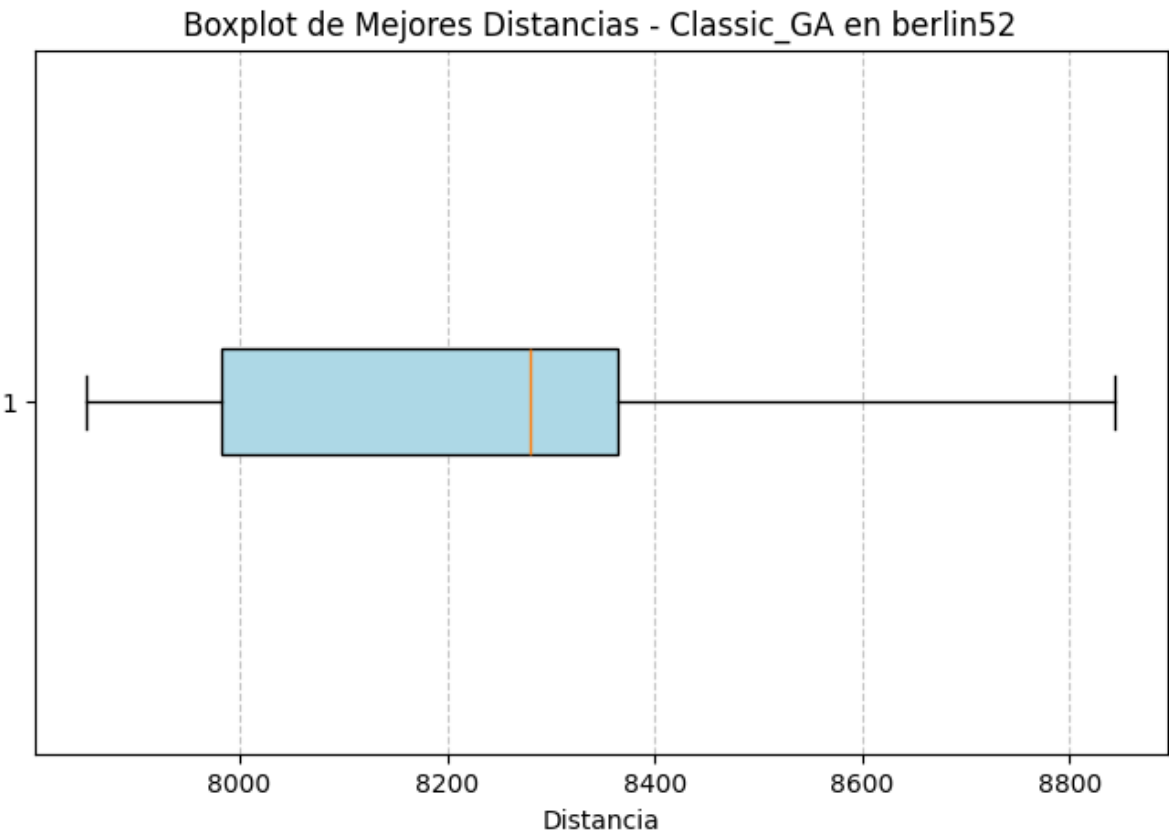
berlin52

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7851.82	8264.77 ± 386.07	9.58	4.45

- [Curva de convergencia promedio GA - berlin52]



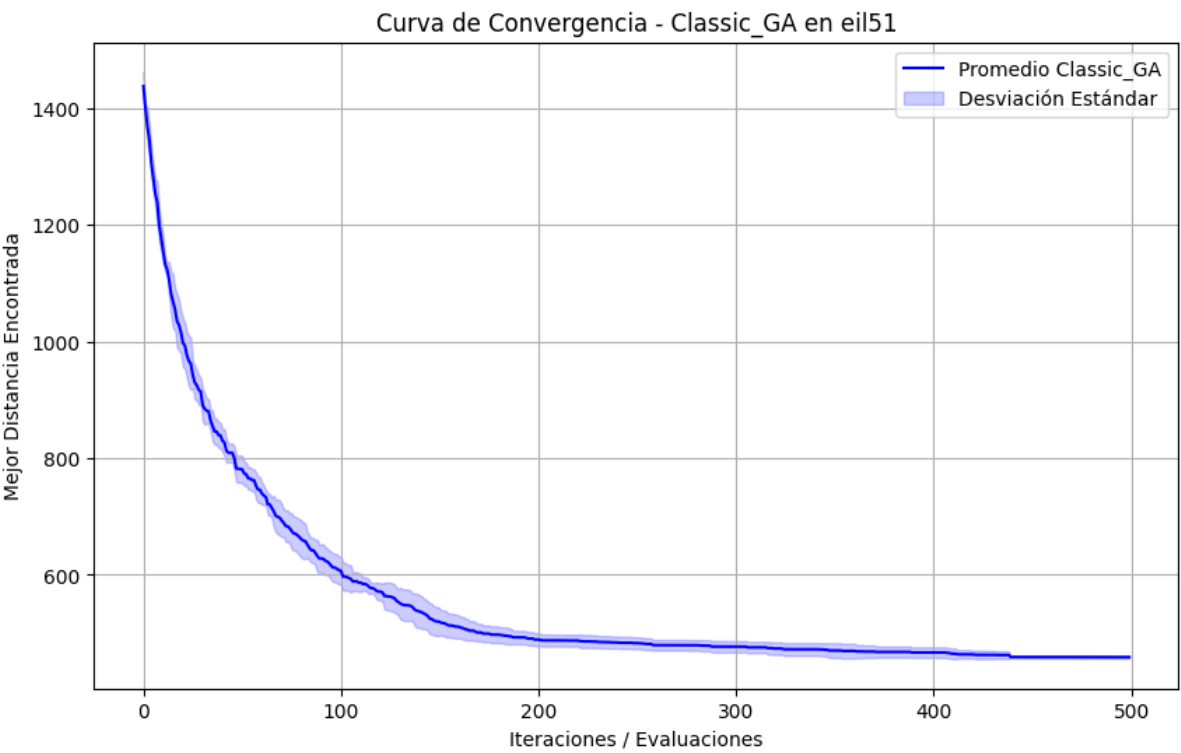
- [Boxplot de mejores distancias GA - berlin52]



eil51

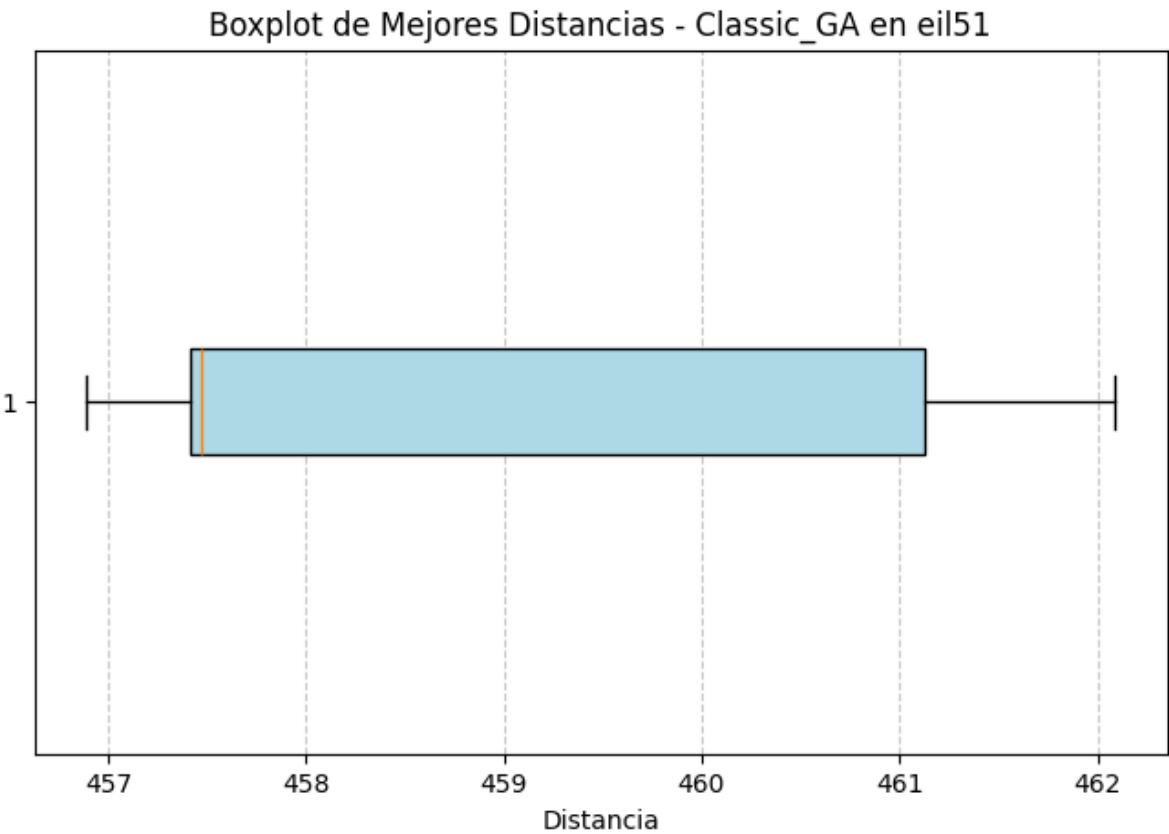
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
eil51	426	456.89	459.00 ± 2.42	7.75	4.42

- [Curva de convergencia promedio GA - eil51]





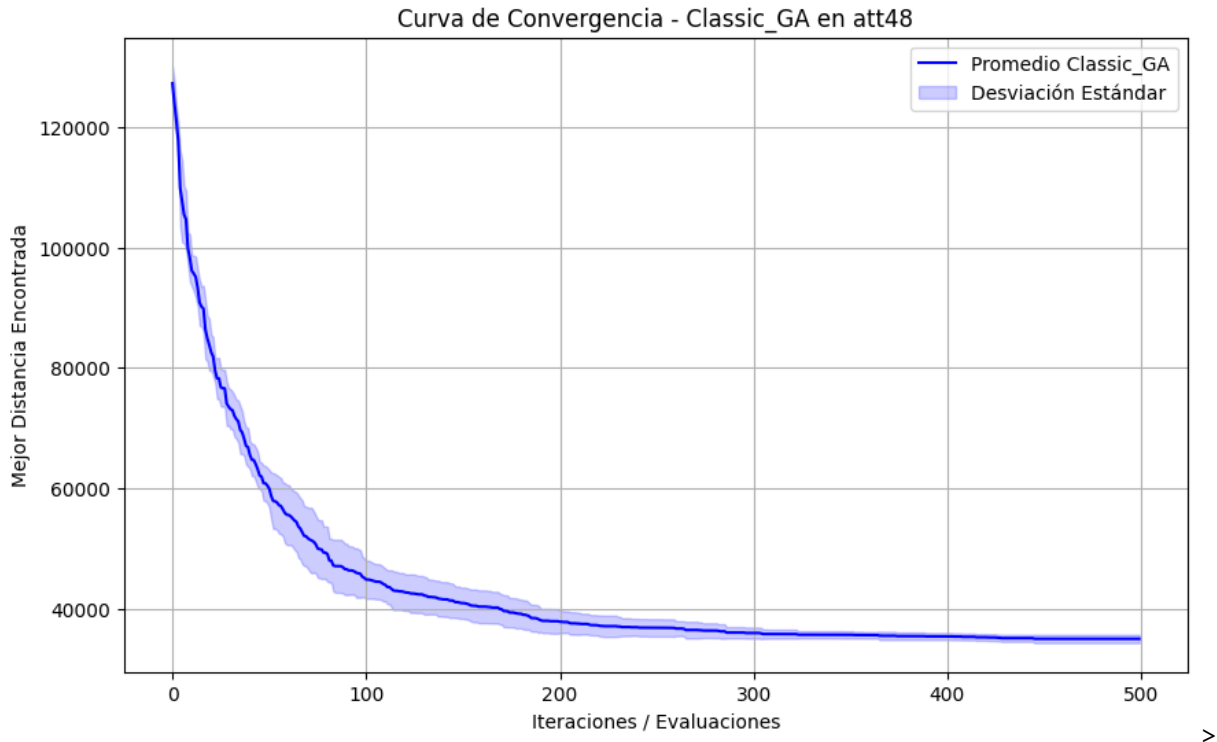
- *[Boxplot de mejores distancias GA - eil51]*



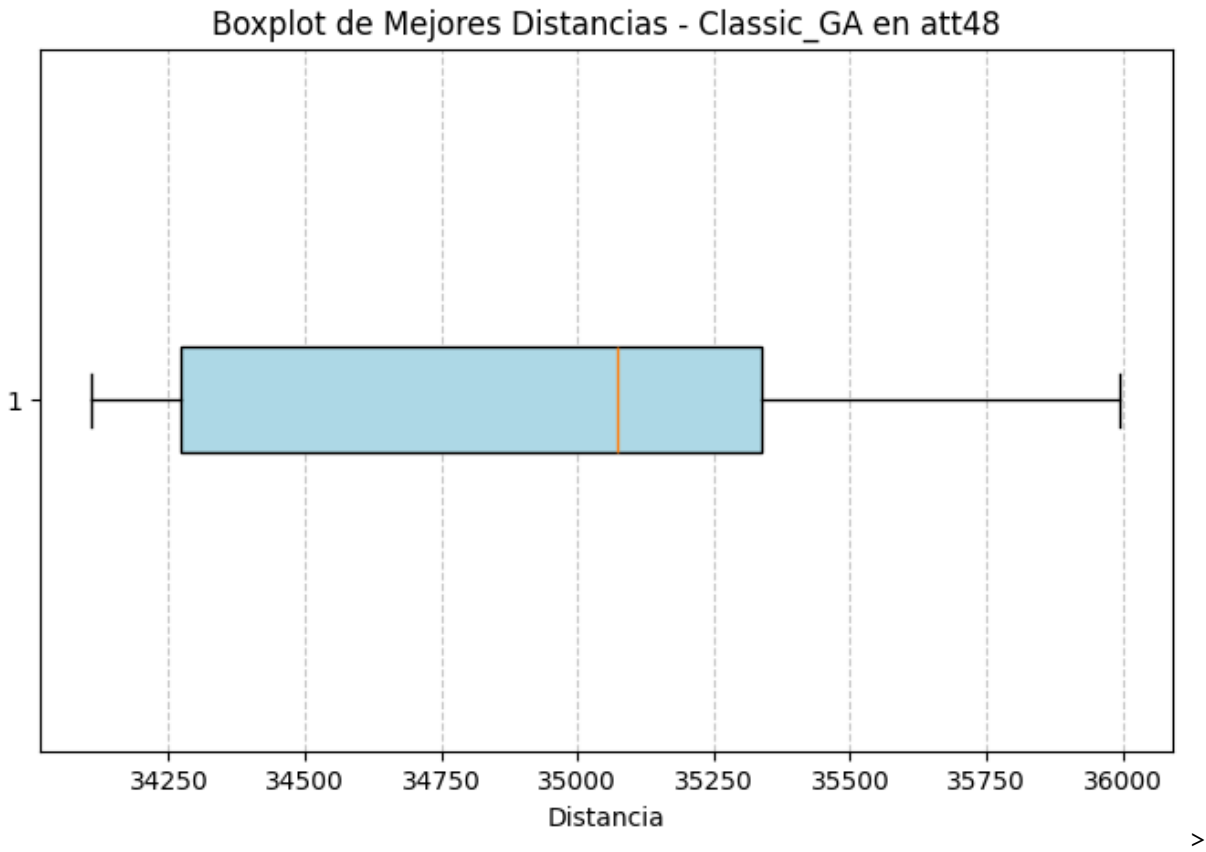
att48

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
att48	10628	34107.07	34957.42 ± 779.50	228.92	4.19

- *[Curva de convergencia promedio GA - att48]*



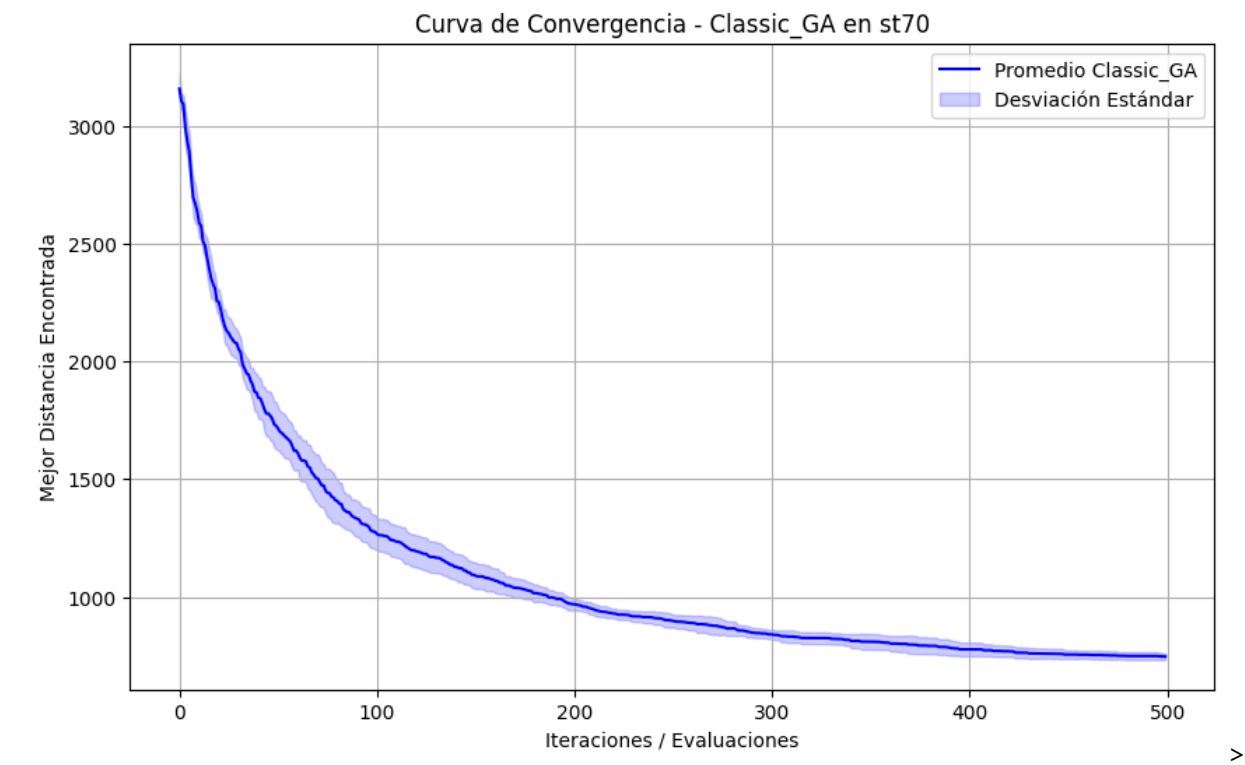
- [Boxplot de mejores distancias GA - att48]



st70

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
st70	675	723.64	747.32 ± 16.08	10.71	5.22

- [Curva de convergencia promedio GA - st70]



- [Boxplot de mejores distancias GA - st70]

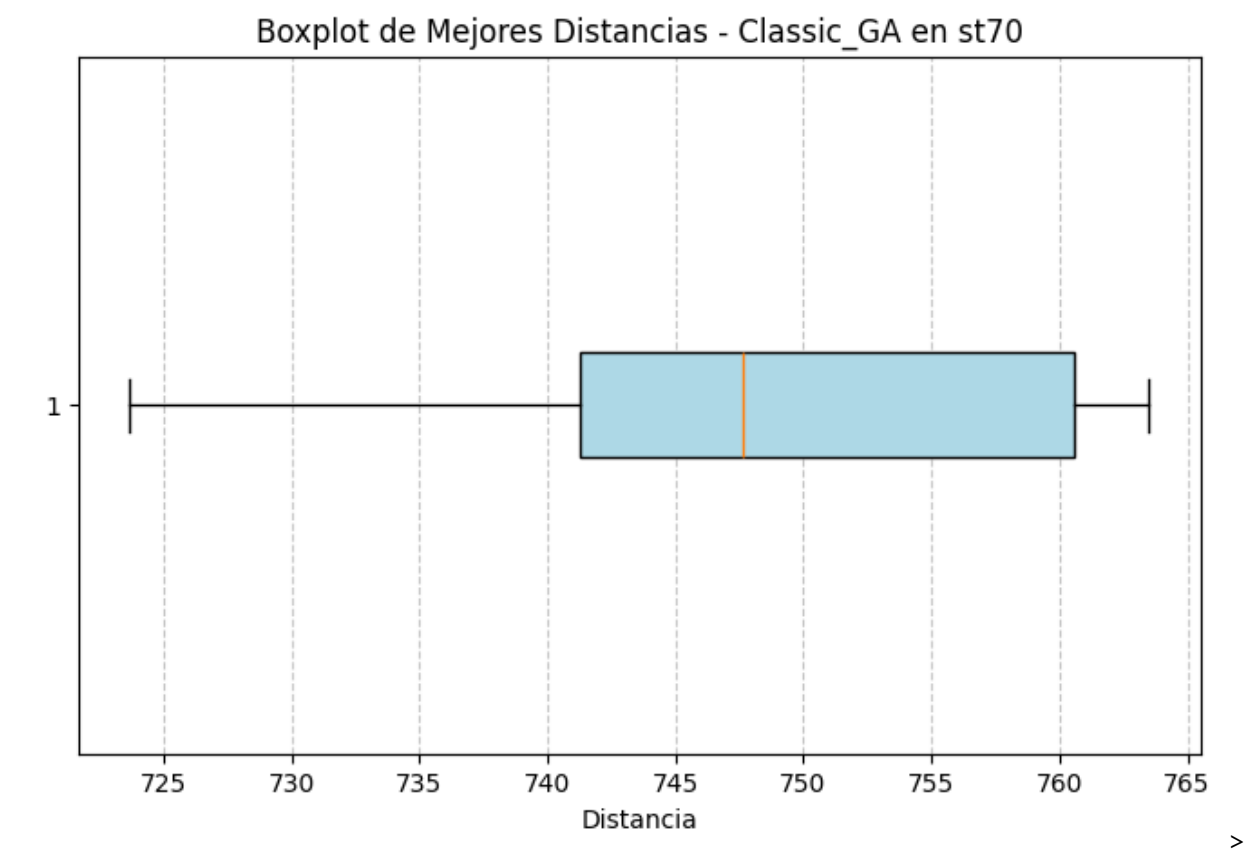


Tabla comparativa

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
-----------	--------	-----------------	--------------------	------------------	---------------------

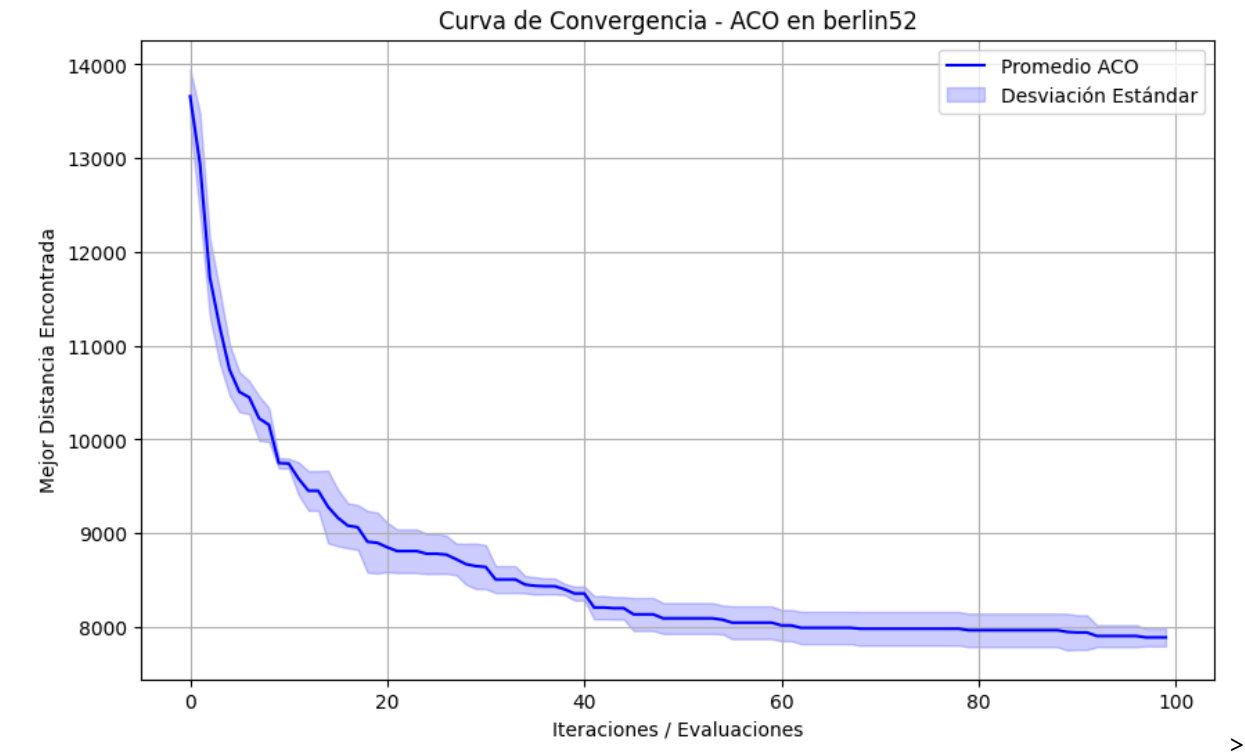
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7851.81	8264.77 ± 386.07	9.58	4.45
eil51	426	456.88	459.00 ± 2.42	7.75	4.41
att48	10628	34107.06	34957.42 ± 779.50	228.92	4.19
st70	675	723.63	747.32 ± 16.08	10.71	5.21

4.2. Ant Colony Optimization (ACO)

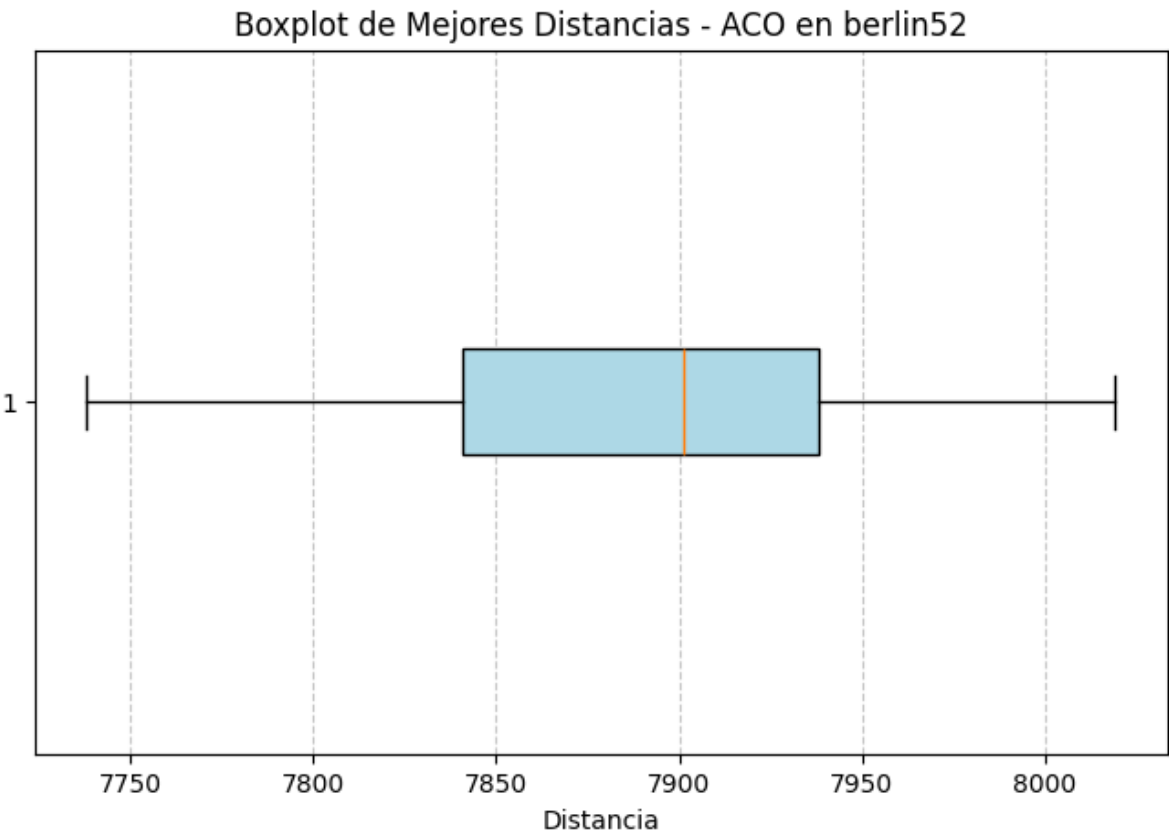
berlin52

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7737.78	7887.40 ± 105.72	4.58	10.72

- [Curva de convergencia promedio ACO - berlin52]



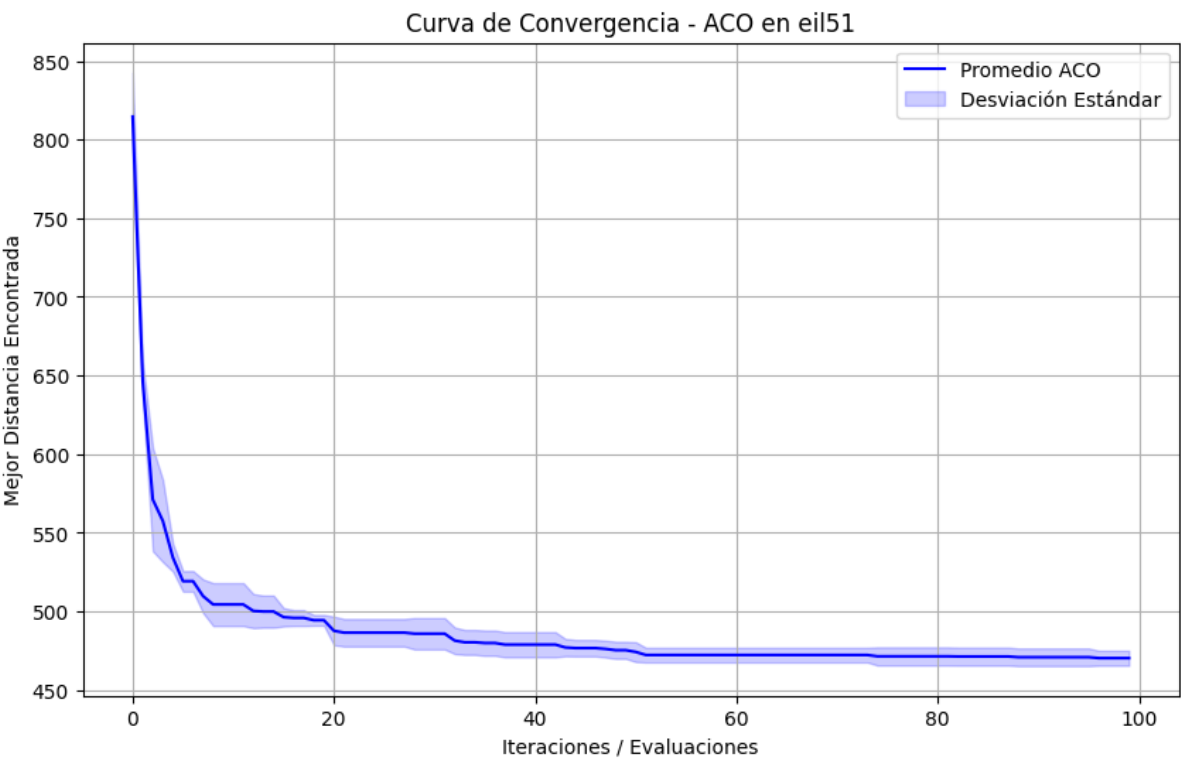
- [Boxplot de mejores distancias ACO - berlin52]



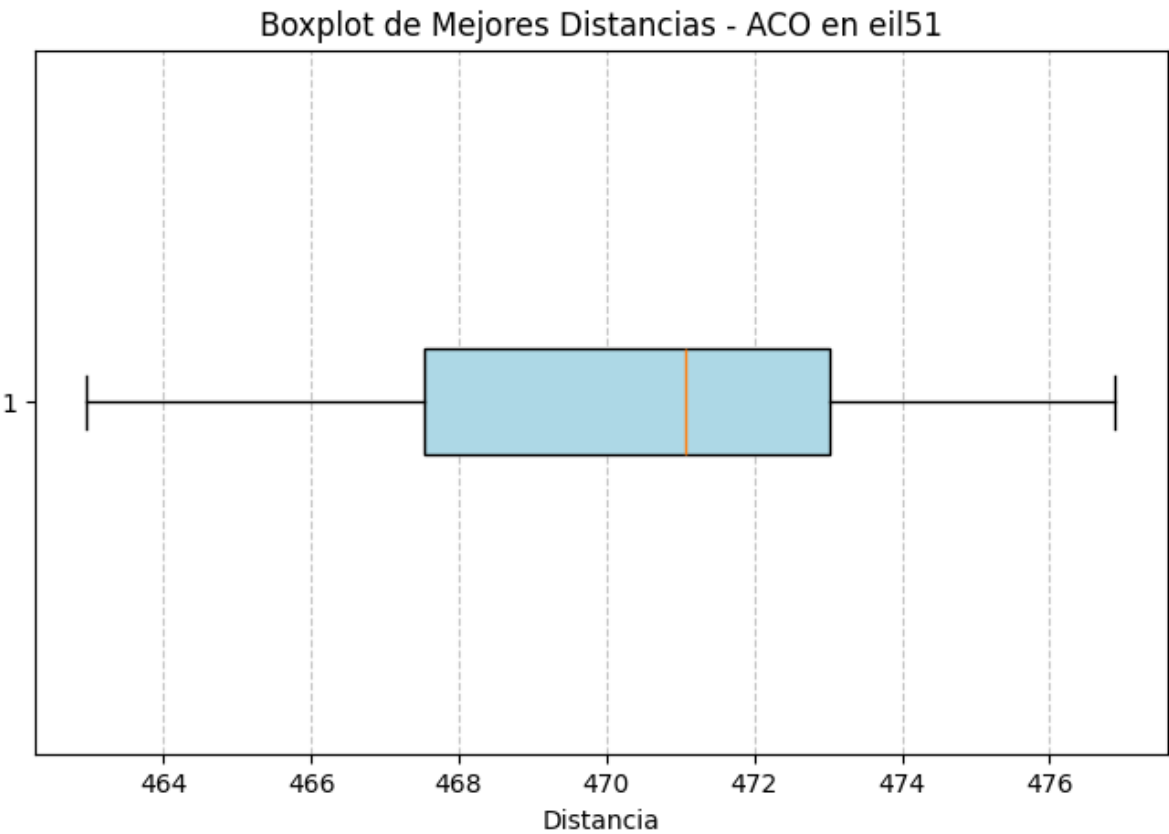
eil51

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
eil51	426	462.95	470.29 ± 5.32	10.40	8.88

- [Curva de convergencia promedio ACO - eil51]



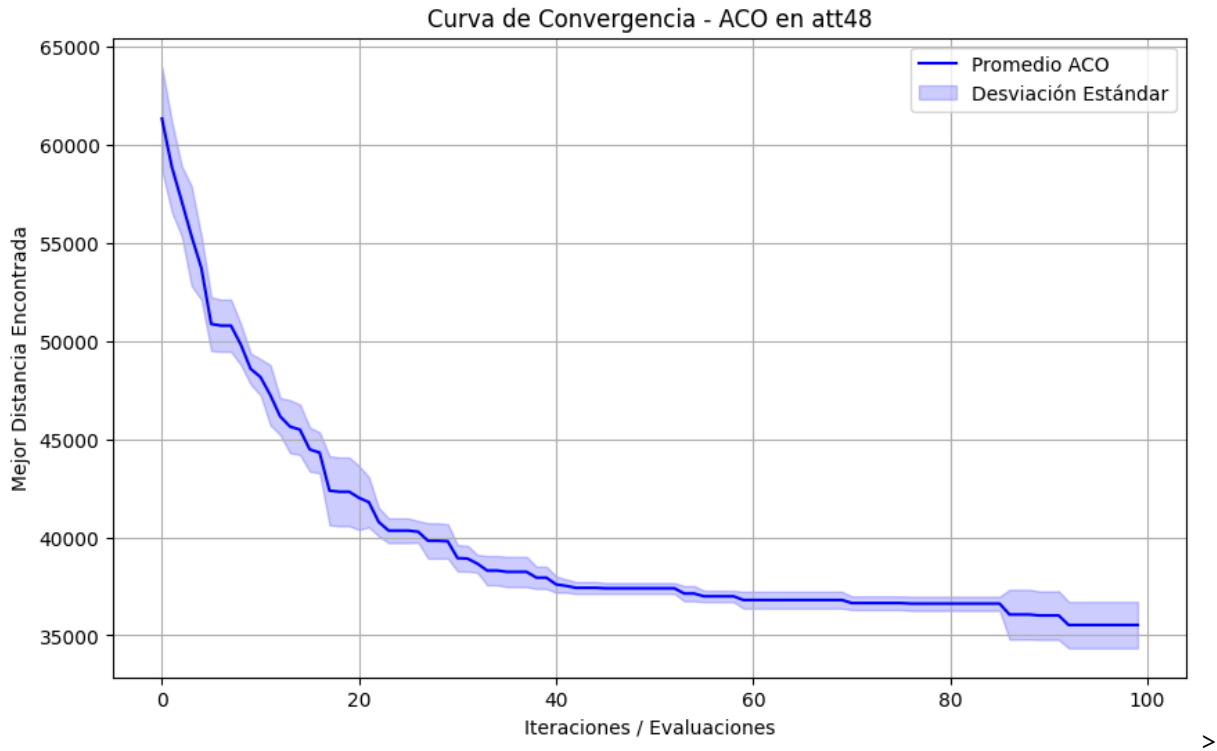
- [Boxplot de mejores distancias ACO - eil51]



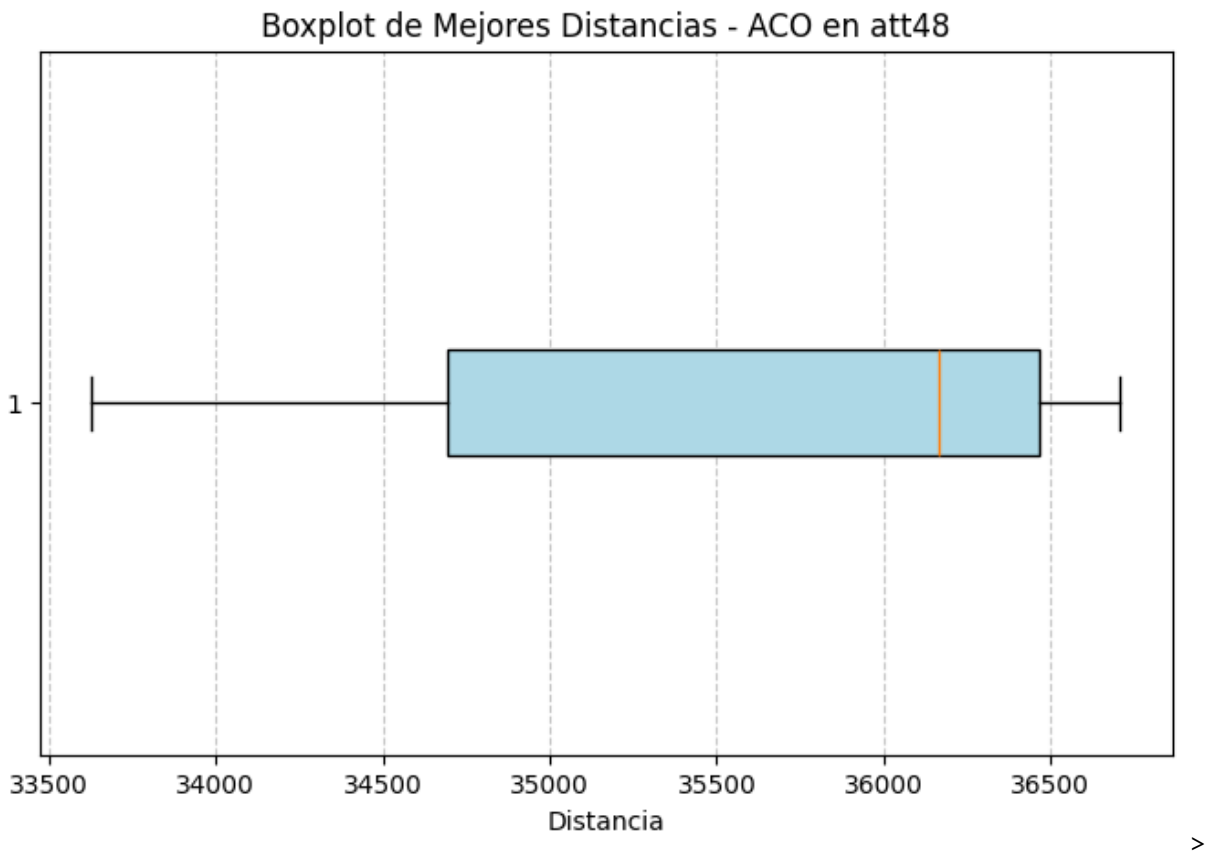
att48

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
att48	10628	33625.62	35531.95 ± 1322.09	234.32	8.52

- [Curva de convergencia promedio ACO - att48]



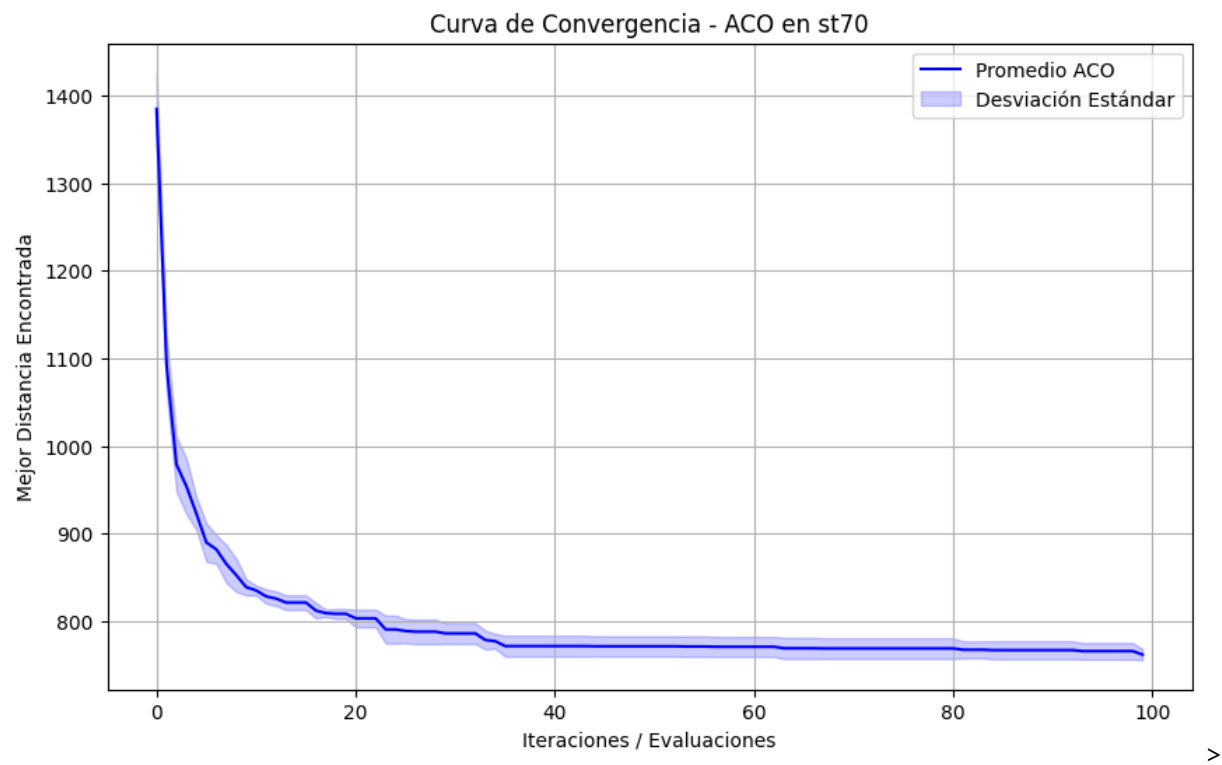
- [Boxplot de mejores distancias ACO - att48]



st70

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
st70	675	752.11	762.00 ± 7.07	12.89	14.33

- *[Curva de convergencia promedio ACO - st70]*



- *[Boxplot de mejores distancias ACO - st70]*

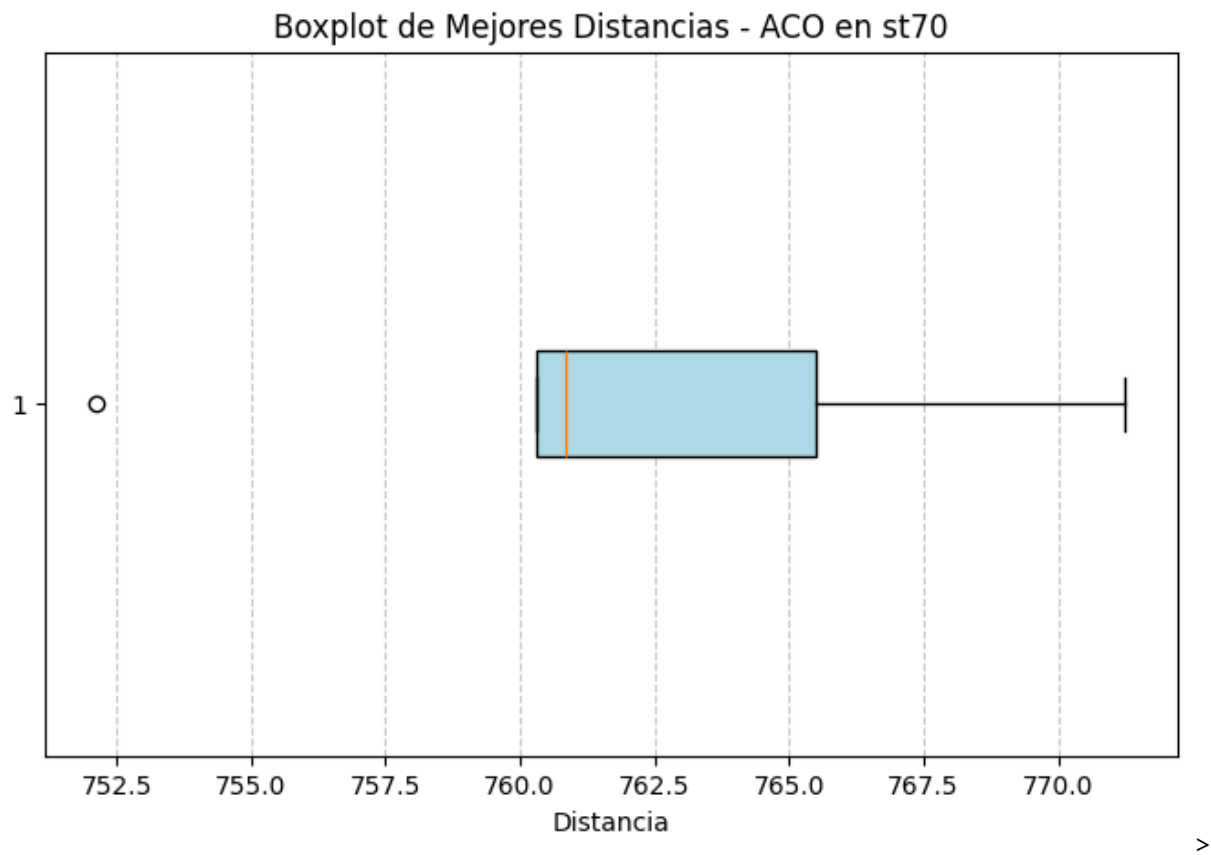


Tabla comparativa

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
-----------	--------	-----------------	--------------------	------------------	---------------------



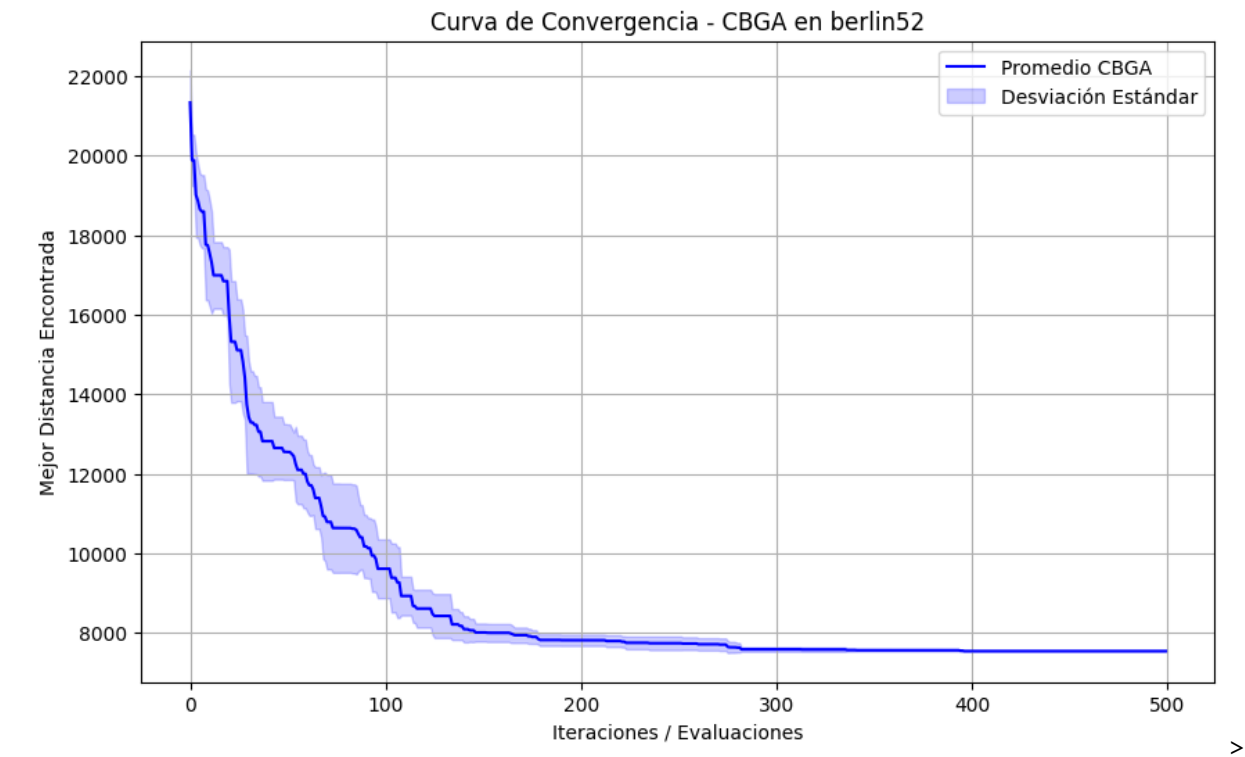
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7737.78	7887.40 ± 105.72	4.58	10.72
eil51	426	462.94	470.29 ± 5.32	10.40	8.87
att48	10628	33625.61	35531.95 ± 1322.09	234.32	8.51
st70	675	752.11	762.00 ± 7.07	12.89	14.32

4.3. Algoritmo Genético de Chu-Beasley (CBGA)

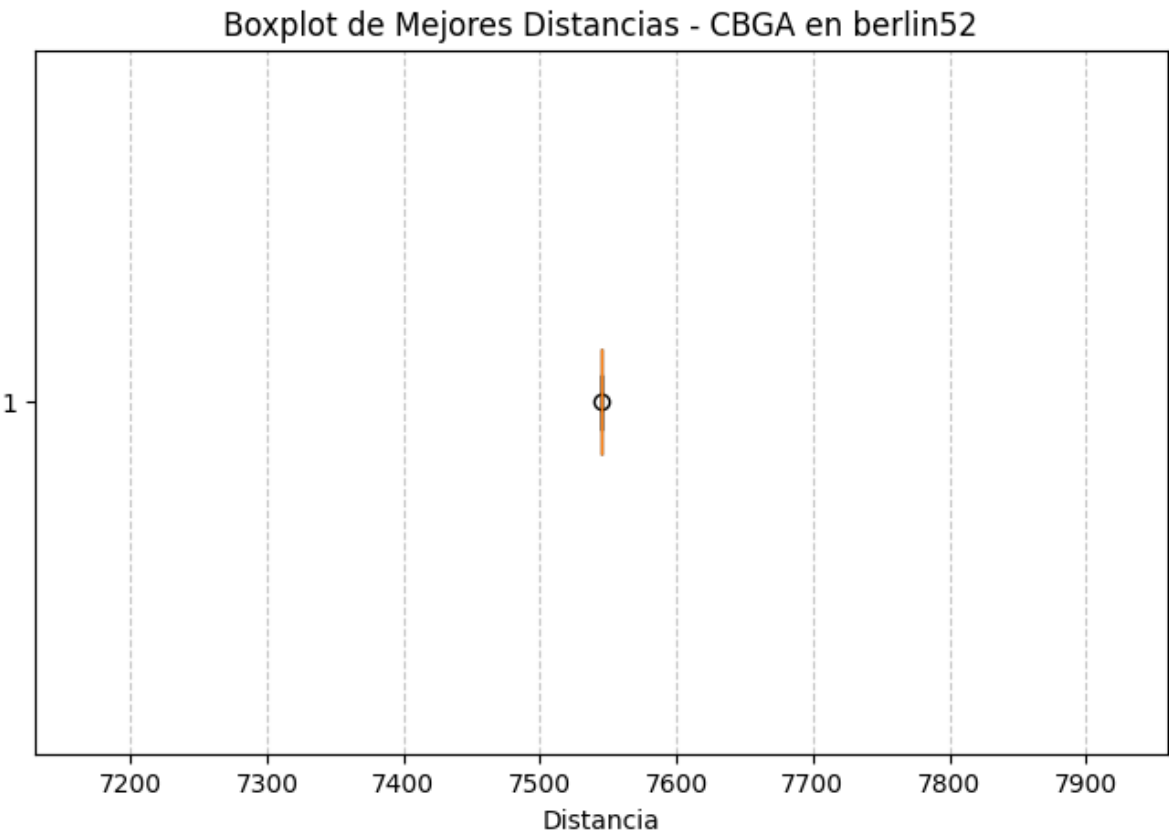
berlin52

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7544.37	7544.37 ± 0.00	0.03	3.24

- *[Curva de convergencia promedio CBGA - berlin52]*



- *[Boxplot de mejores distancias CBGA - berlin52]*

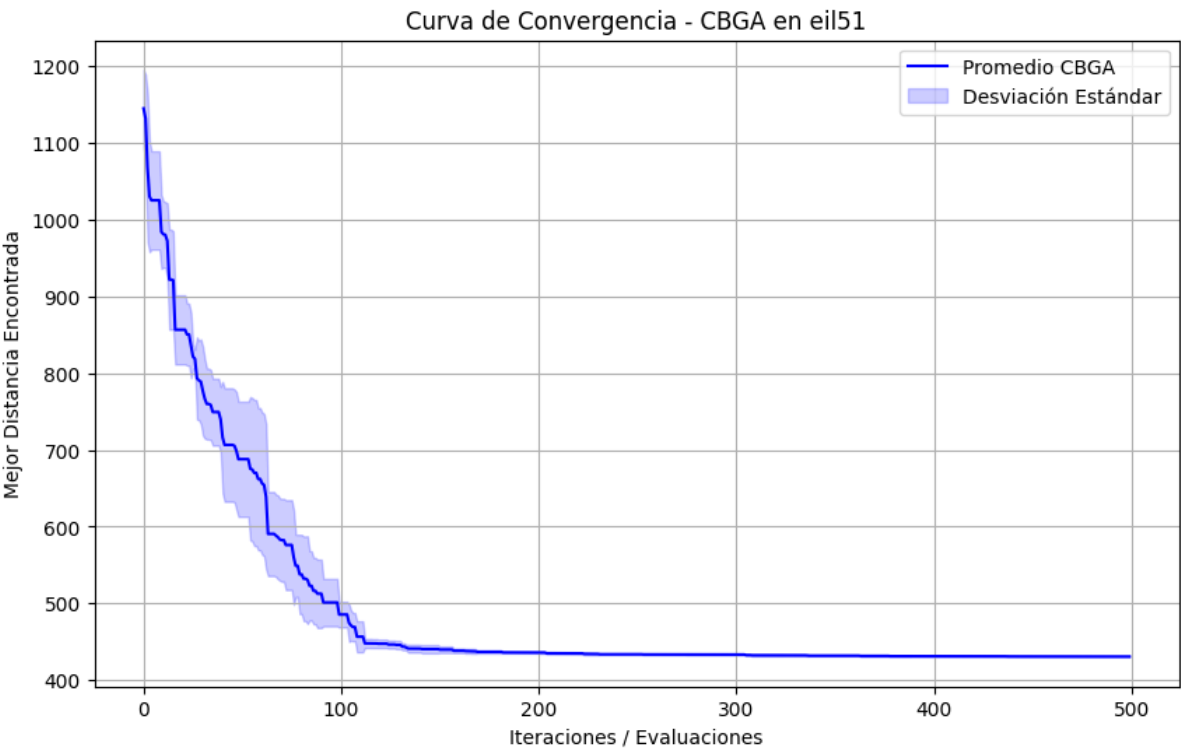


>

eil51

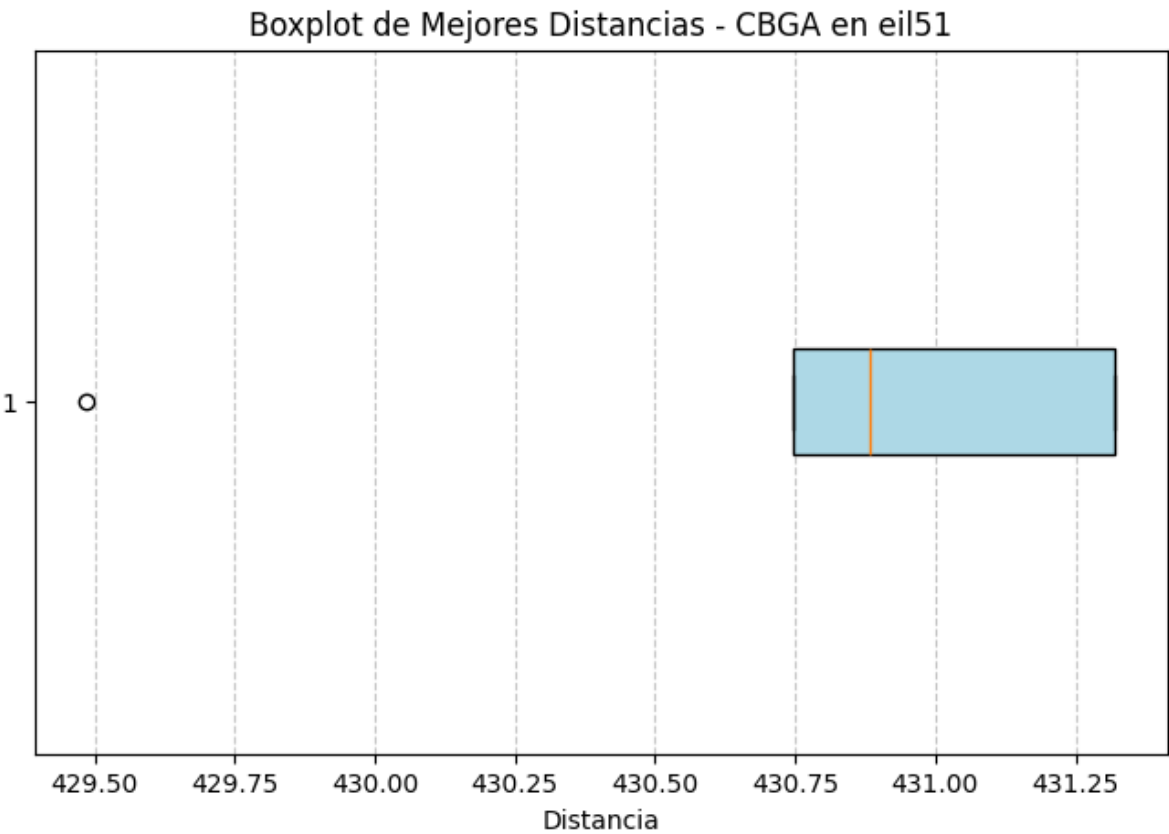
Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
eil51	426	429.48	430.75 ± 0.75	1.12	3.01

- [Curva de convergencia promedio CBGA - eil51]



>

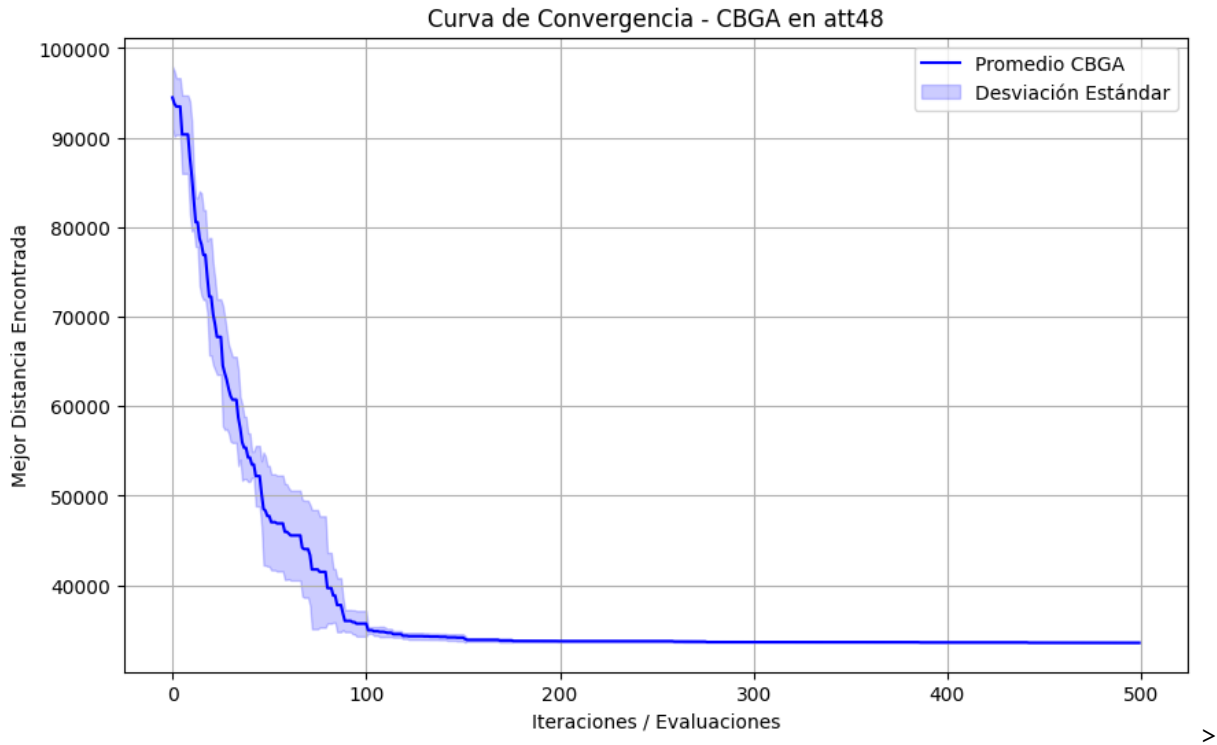
- *[Boxplot de mejores distancias CBGA - eil51]*



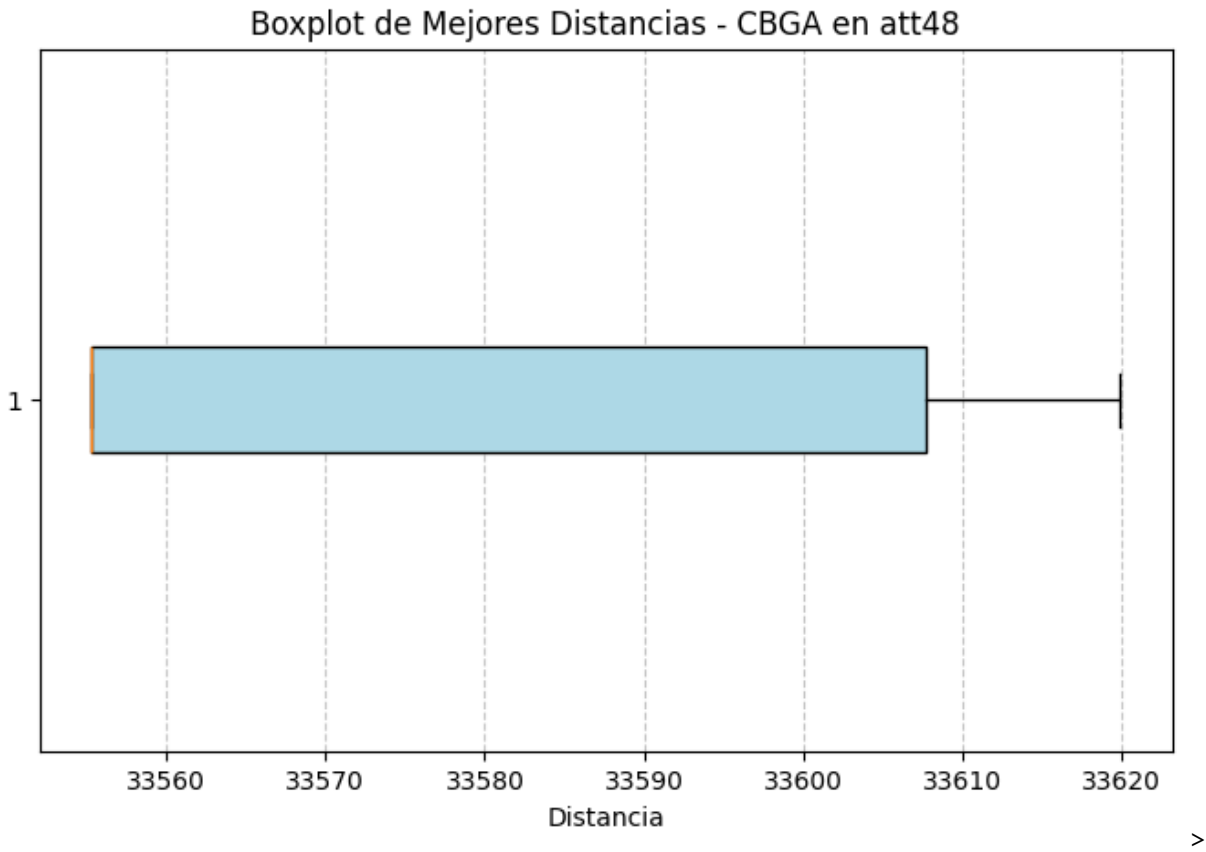
att48

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
att48	10628	33555.28	33578.68 ± 32.34	215.95	2.60

- *[Curva de convergencia promedio CBGA - att48]*



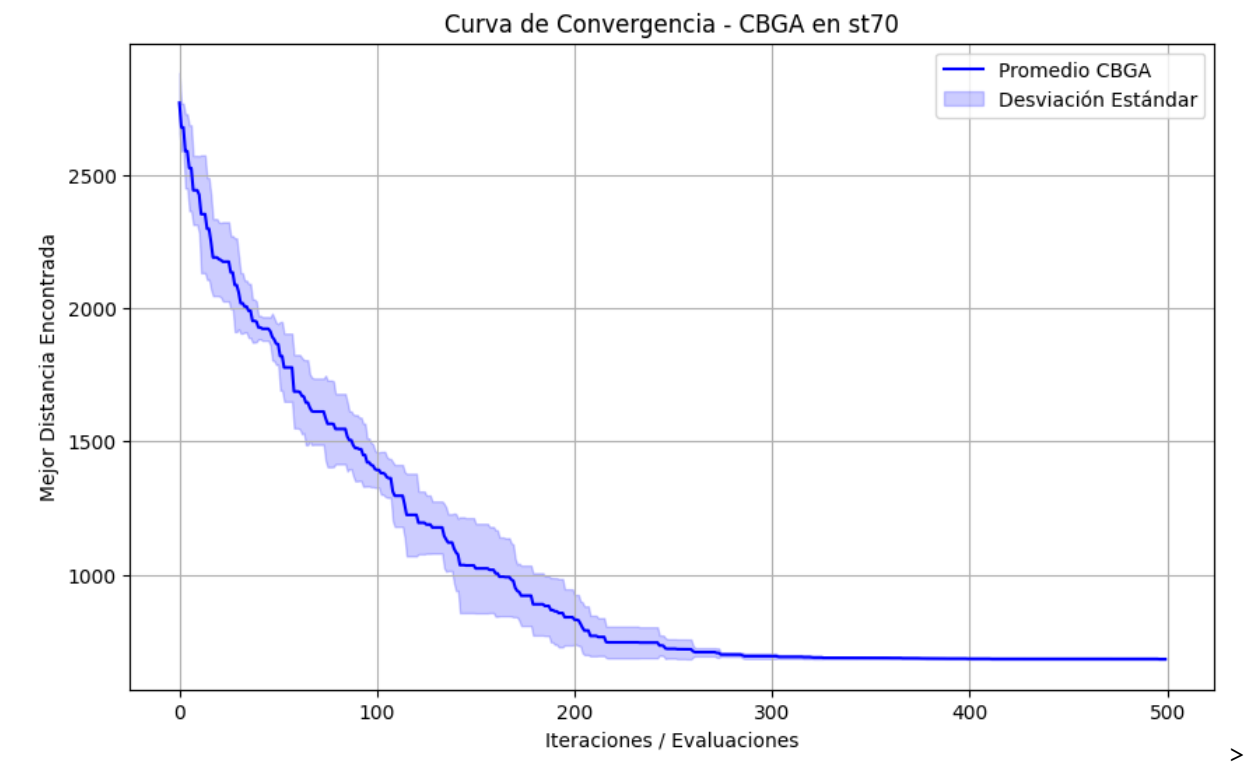
- [Boxplot de mejores distancias CBGA - att48]



st70

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
st70	675	679.65	683.75 ± 4.31	1.30	4.90

- [Curva de convergencia promedio CBGA - st70]



- [Boxplot de mejores distancias CBGA - st70]

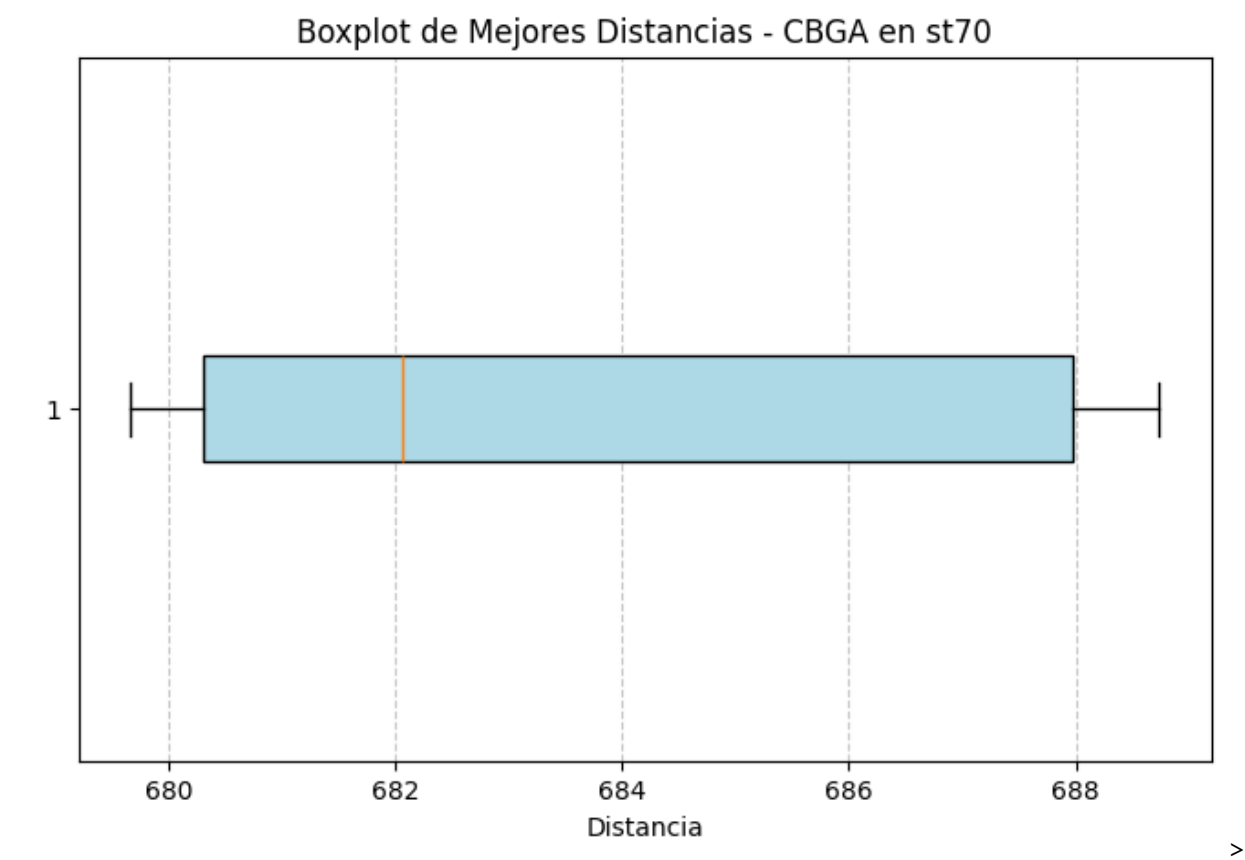


Tabla comparativa

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
-----------	--------	-----------------	--------------------	------------------	---------------------

Instancia	Óptimo	Mejor Distancia	Distancia Promedio	GAP Promedio (%)	Tiempo Promedio (s)
berlin52	7542	7544.36	7544.37 ± 0.00	0.03	3.24
eil51	426	429.48	430.75 ± 0.75	1.12	3.00
att48	10628	33555.27	33578.68 ± 32.34	215.95	2.60
st70	675	679.65	683.75 ± 4.31	1.30	4.89

## 5. Afinamiento (Tuning) de Hiperparámetros

Se realizó una búsqueda en rejilla (Grid Search) sobre la instancia **berlin52** utilizando 3 semillas para encontrar la mejor configuración de cada algoritmo.

- **Mejor GA:** `{'pop_size': 100, 'pm': 0.1, 'pc': 0.85, 'elitism_k': 2, 'iterations': 200}`. Distancia promedio: 8631.69.
- **Mejor ACO:** `{'num_ants': 40, 'alpha': 2.0, 'beta': 2.0, 'rho': 0.1, 'iterations': 50}`. Distancia promedio: 7623.97.
- **Mejor CBGA:** `{'pop_size': 50, 'threshold': 20, 'iterations': 200}`. Distancia promedio: 7793.00.

## 6. Discusión Crítica

1. **¿Quién obtiene el menor GAP promedio?** El **CBGA** obtiene, por un margen muy amplio, el menor GAP promedio. En **berlin52** logró un GAP casi perfecto de 0.03%, frente al 4.58% de ACO y el 9.58% de GA. En **st70**, CBGA logró 1.30% frente a >10% de los otros algoritmos.
2. **¿Quién es más estable (menor desviación)?** El **CBGA** demostró una estabilidad excepcional. En **berlin52**, su desviación estándar fue de 0.00 (encontró prácticamente la misma solución óptima en todas las semillas). En **eil51**, su desviación fue de apenas 0.75.
3. **¿Quién llega más rápido a buenas soluciones?** El **CBGA** es el más rápido en tiempo de ejecución (promedios de 2.6s a 4.8s), superando al GA clásico (~4-5s) y siendo significativamente más rápido que ACO (~8-14s). La inclusión de la búsqueda local 2-opt acelera drásticamente la convergencia hacia buenas soluciones.
4. **¿Qué ocurre cuando aumenta la dimensión?** Al pasar de ~50 nodos a 70 nodos (**st70**), el GA y el ACO sufren una degradación en la calidad de la solución (GAPs de 10.71% y 12.89% respectivamente). Sin embargo, el CBGA escala de manera excelente, manteniendo un GAP sumamente bajo de 1.30%.
5. **¿El tuning cambia el ganador?** No. Aunque el tuning mejoró el rendimiento de ACO (bajando su distancia promedio a 7623.97 en **berlin52**), el rendimiento base del CBGA (7544.37) sigue siendo superior a las versiones afinadas de GA y ACO.
6. **¿CBGA mejora GA? ¿En qué sentido?** Sí, de forma contundente. Mejora en **calidad** (reduce el GAP de 9.58% a 0.03% en **berlin52**), en **estabilidad** (reduce la desviación estándar a casi cero) y en **tiempo** (es más rápido al requerir una población menor, P=50 vs P=100, compensada por la intensificación del 2-opt).

## 7. Conclusiones

Tras evaluar los algoritmos bajo múltiples criterios, se concluye lo siguiente:

- **Calidad y Estabilidad (Ganador: CBGA):** La combinación de control estricto de diversidad (distancia de Hamming) y la intensificación local (2-opt) permite al CBGA evitar óptimos locales y converger a soluciones casi óptimas de manera consistente en todas las ejecuciones.
- **Velocidad (Ganador: CBGA / GA):** Los algoritmos basados en poblaciones (GA y CBGA) resultaron computacionalmente más eficientes que el enfoque constructivo de ACO, el cual requiere un alto costo computacional para la actualización y consulta de la matriz de feromonas.
- **Veredicto Final:** El **Algoritmo Genético de Chu-Beasley (CBGA)** es indiscutiblemente el mejor algoritmo para este problema. Priorizando el criterio de **calidad de la solución y escalabilidad**, el CBGA demuestra que una metaheurística híbrida (memética) que equilibra cuidadosamente la exploración (control de diversidad) y la explotación (búsqueda local) supera ampliamente a los enfoques clásicos puros.