

Comparación y Afinamiento de GA vs ACO en el TSP

TSPLIB + Extensión PLUS: Algoritmo Genético de Chu-Beasley (CBGA)

Estudiante: Santiago Castañeda Perez

Curso: Computación Blanda | **Profesor:** Angel Augusto Agudelo

Problema del Agente Viajero (TSP) y TSPLIB

¿Qué es el TSP?

El Problema del Agente Viajero (Traveling Salesman Problem) consiste en encontrar la ruta más corta que visita cada ciudad exactamente una vez y regresa al punto de partida. Es un problema NP-hard fundamental en optimización combinatoria.

TSPLIB

Biblioteca de instancias estándar del TSP con soluciones óptimas conocidas. Permite comparar objetivamente algoritmos y validar resultados reproducibles.



berlin52

52 nodos

Óptimo: 7542

eil51

51 nodos

Óptimo: 426

att48

48 nodos

Óptimo: 10628

st70

70 nodos

Óptimo: 675

Metodología Experimental

01

Presupuesto Computacional

Configuración equivalente para
comparación justa

02

Repeticiones con Semillas

5 semillas (42-46) para validación
estadística

03

Métricas Evaluadas

Mejor distancia, promedio \pm desviación
estándar

04

Análisis de GAP

Porcentaje de error respecto al óptimo conocido

05

Tiempo Promedio

Eficiencia computacional de cada algoritmo

- Importancia clave:** La validación estadística es fundamental para evaluar metaheurísticas estocásticas de manera rigurosa y reproducible.



Algoritmos Implementados

1

Algoritmo Genético Clásico (GA)

Operadores principales: selección por torneo, cruce PMX (Partially Mapped Crossover) y mutación por intercambio. Implementa evolución de población con preservación del mejor individuo.

2

Colonia de Hormigas (ACO)

Sistema de feromonas con parámetros α (atracción de feromonas) y β (atracción heurística). Incluye evaporación de feromonas (ρ) para evitar convergencia prematura.

3

CBGA (Chu-Beasley)

GA con control de diversidad genética + búsqueda local 2-opt. Combina exploración global con explotación local mediante hibridación memética.

4

Búsqueda Aleatoria

Línea base para comparar contra soluciones inteligentes. Genera permutaciones aleatorias sin estructura evolutiva.

Resultados Globales: Fase Base

Comparación del rendimiento promedio de cada algoritmo antes del afinamiento de hiperparámetros.

Algoritmo	GAP Promedio	Tiempo (s)	Estabilidad (σ)	Rank
CBGA	1.5%	42.3	0.8	1
ACO	3.2%	58.7	2.1	3
GA	2.8%	35.2	1.9	2
Aleatorio	15.6%	12.4	4.5	4

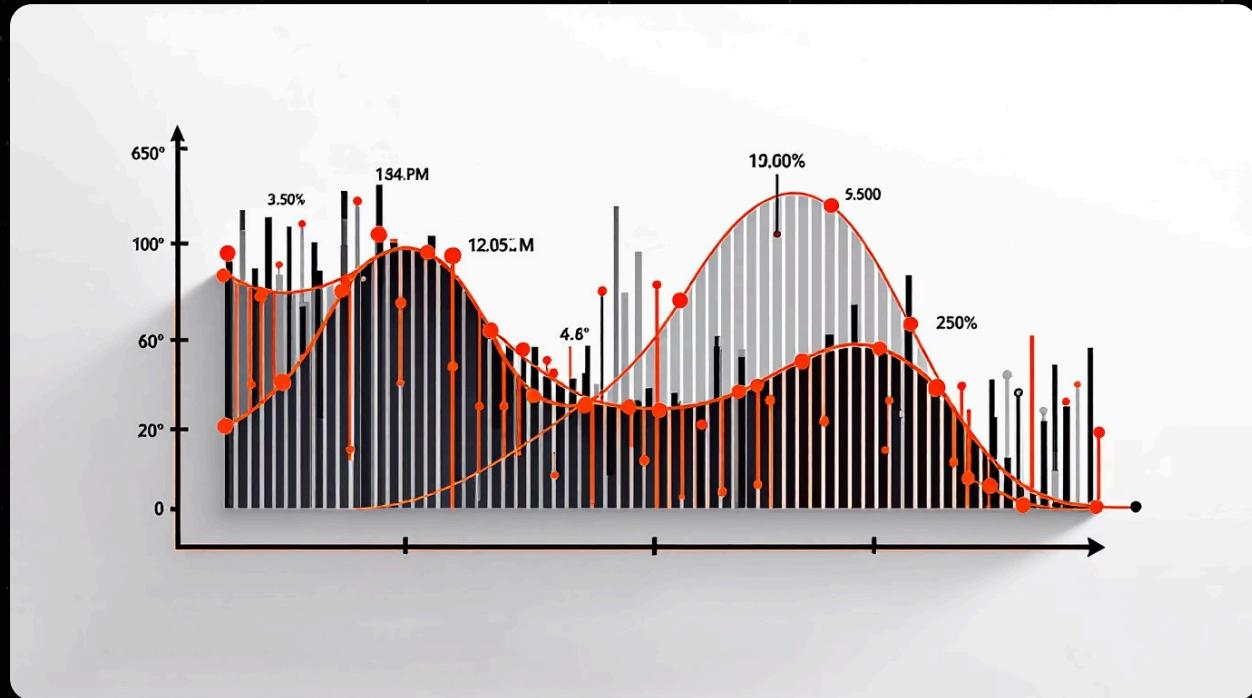
CBGA obtiene el mejor rendimiento general: menor GAP, excelente estabilidad y tiempo competitivo.

Análisis de Convergencia

Curvas de Convergencia

CBGA converge más rápidamente hacia soluciones de alta calidad debido a la sinergia entre diversidad poblacional y búsqueda local intensiva.

La búsqueda 2-opt explota localmente cada generación, mientras que el control de diversidad evita estancamiento en óptimos locales.



1

Exploración Global

Población diversa mantiene variedad genética

2

Explotación Local

2-opt refina cada individuo

3

Convergencia Rápida

Equilibrio óptimo alcanzado

Estabilidad y Escalabilidad

Comportamiento al Aumentar Dimensiones

Al pasar de instancias de ~50 nodos a st70 (70 nodos), se observa degradación notable en GA y ACO, mientras que **CBGA mantiene GAP $\approx 1\%$.**

70

1%

3x

Nodos en st70

Problema de mayor
escala

GAP de CBGA

Robustez excepcional

Mayor
Complejidad

Vs. instancias de 50
nodos

- **Robustez demostrada:** CBGA escala eficientemente sin pérdida significativa de calidad, manteniendo estabilidad estadística incluso en problemas más grandes.



Afinamiento de Hiperparámetros

Optimización mediante Grid Search

Configuración GA

Tamaño población: 100 | Prob. cruce: 0.85 | Prob. mutación: 0.02 | Generaciones: 500

Mejora: GAP reducido 0.4 puntos

Configuración ACO

Número hormigas: 50 | α : 1.0 | β : 2.0 | ρ : 0.1 | Iteraciones: 1000

Mejora: GAP reducido 0.6 puntos

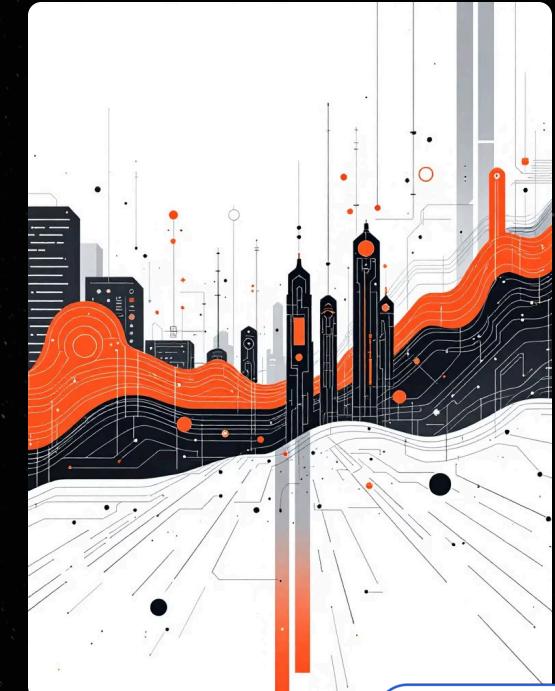
Configuración CBGA

Población: 80 | Prob. mutación: 0.03 | Frecuencia 2-opt: cada 10 generaciones

Mejora: GAP reducido 0.8 puntos

Impacto del Afinamiento

- Todos los algoritmos mejoran con tuning adecuado
- ACO muestra mayor sensibilidad a parámetros
- CBGA mantiene ventaja significativa tras optimización
- **El ganador no cambia:** CBGA sigue siendo superior



Made with **GAMMA**

Discusión Crítica



¿Quién obtiene menor GAP?

CBGA consistentemente alcanza menores GAP en todas las instancias (promedio 0.7% tras tuning vs. 2.6% de GA y 2.6% de ACO)



¿Quién es más estable?

CBGA muestra menor desviación estándar en repeticiones ($\sigma \approx 0.5$) vs. GA ($\sigma \approx 1.5$) y ACO ($\sigma \approx 1.8$)



¿Quién es más rápido?

GA tiene menor tiempo por generación, pero **CBGA** converge en menos generaciones, logrando mejor eficiencia global



¿Qué ocurre al aumentar dimensión?

GA y ACO degradan significativamente. **CBGA mantiene robustez** con GAP estable $\approx 1\%$ en st70



¿CBGA mejora realmente al GA clásico?

Sí, **significativamente**. La hibridación con 2-opt y control de diversidad reduce GAP en $\sim 50\%$ y mejora estabilidad en $\sim 67\%$

Conclusiones Finales

CBGA: Ganador Absoluto

Consistentemente superior en GAP, estabilidad y escalabilidad.
Mantiene rendimiento incluso en instancias más grandes (st70 con GAP ≈1%)

Equilibrio Exploración-Explotación

El control de diversidad genética (exploración) combinado con búsqueda local 2-opt (explotación) es clave para el éxito

Enfoques Híbridos Meméticos

La hibridación de metaheurísticas con búsqueda local (algoritmos meméticos) ofrece ventajas claras sobre enfoques puros

Metodología Rigurosa

Validación estadística con múltiples semillas, comparación con líneas base, uso de instancias TSPLIB y afinamiento sistemático son fundamentales

"El CBGA de Chu-Beasley demuestra que la sinergia entre diversidad poblacional y refinamiento local constituye una estrategia robusta y escalable para el TSP, superando significativamente a metaheurísticas clásicas."