



UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Resolución del Problema de Asignación Cuadrática mediante Búsqueda Tabú

Trabajo Final Unidad II

Autores: Camila Llamirez
Pablo Jordan
Camilo Avendaño
Asignatura: Algoritmos Avanzados
y Complejidad Computacional
Profesor: Mario Inostroza

Santiago, 13 de Junio de 2025

Índice

1. Introducción	2
2. Explicación del Problema de Asignación Cuadrática (QAP)	3
2.1. Definición Formal	3
2.2. Complejidad Computacional	3
2.3. Aplicaciones Prácticas	3
3. Descripción de la Metaheurística: Búsqueda Tabú	5
3.1. Fundamentos Teóricos	5
3.2. Justificación de la Selección	5
3.3. Componentes Diferenciadores	5
3.3.1. Estructura de Memoria	5
3.3.2. Lista Tabú	6
3.3.3. Criterio de Aspiración	6
3.3.4. Estrategias de Diversificación e Intensificación	6
4. Modelamiento del Problema con la Metaheurística	7
4.1. Representación de la Solución	7
4.2. Manejo de Restricciones	7
4.3. Función Objetivo y Evaluación Incremental	7
4.4. Estructura de Vecindario	8
5. Experimentación Computacional	9
5.1. Ambiente de Pruebas y Parámetros	9
5.2. Resultados por Instancia	10
5.2.1. Instancia bur26c	10
5.2.2. Instancia chr25a	11
5.2.3. Instancia esc32h	12
5.2.4. Instancia esc64a	13
5.2.5. Instancia lipa60a	14
5.2.6. Instancia tai80a	15
5.2.7. Instancia tai80b	16
5.2.8. Instancia tho150	17
5.3. Resultados Globales	18
6. Conclusiones y Comentarios Finales	20
Bibliografía	22

1 Introducción

El Problema de Asignación Cuadrática (QAP) representa uno de los desafíos fundamentales en la optimización combinatoria, siendo clasificado como NP-hard [1]. Su importancia trasciende el ámbito teórico, con aplicaciones críticas en el diseño de instalaciones industriales [2], la arquitectura de circuitos VLSI [3], el diseño de teclados ergonómicos [4], y más recientemente en bioinformática computacional para el análisis de interacciones proteína-proteína [5]. Dada su complejidad, el uso de metaheurísticas se presenta como el enfoque más efectivo para obtener soluciones de alta calidad en tiempos de cómputo razonables.

El objetivo principal de este estudio es implementar y evaluar el rendimiento de la metaheurística de **Búsqueda Tabú (Tabu Search)** para resolver un conjunto de instancias de referencia del QAP. La elección de esta técnica se fundamenta en su demostrada robustez para guiar la exploración del espacio de soluciones y evitar óptimos locales mediante el uso de estructuras de memoria.

Para la experimentación, se utilizará un conjunto de 8 instancias clásicas extraídas de la librería QAPLIB [6], una colección estándar para la validación de algoritmos. Estas instancias representan un espectro diverso de características y tamaños, lo que permitirá evaluar la escalabilidad y adaptabilidad del algoritmo. El desempeño se medirá en términos de la calidad de las soluciones obtenidas (comparadas con los valores óptimos o las mejores soluciones conocidas) y el tiempo de ejecución requerido.

Este informe se estructura de la siguiente manera: primero, se presenta una explicación formal del Problema de Asignación Cuadrática. A continuación, se describe en detalle la metaheurística de Búsqueda Tabú y el modelamiento del problema. Posteriormente, se exponen los resultados de las pruebas computacionales, tanto a nivel individual por instancia como de forma global. Finalmente, se presentan las conclusiones derivadas del estudio y se comentan posibles líneas de trabajo futuro.

2 Explicación del Problema de Asignación Cuadrática (QAP)

2.1 Definición Formal

El Problema de Asignación Cuadrática fue introducido por Koopmans y Beckmann [2] como un modelo matemático para el problema de localización de plantas industriales. Formalmente, el QAP busca asignar un conjunto de n facilidades a un conjunto de n localizaciones, minimizando el costo total de transporte.

Dadas dos matrices:

- $F = (f_{ij}) \in \mathbb{R}^{n \times n}$: matriz de flujos, donde f_{ij} representa el flujo de materiales, información o interacción entre las facilidades i y j
- $D = (d_{kl}) \in \mathbb{R}^{n \times n}$: matriz de distancias, donde d_{kl} denota la distancia o costo de transporte entre las localizaciones k y l

El objetivo es encontrar una permutación $\pi \in S_n$ (donde S_n denota el grupo simétrico de orden n) que minimice:

$$\min_{\pi \in S_n} z(\pi) = \sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{\pi(i)\pi(j)} \quad (1)$$

2.2 Complejidad Computacional

La complejidad intrínseca del QAP se manifiesta en múltiples dimensiones:

Cuadro 1: Características de complejidad del QAP

Aspecto	Descripción
Espacio de búsqueda	$n!$ permutaciones posibles, crecimiento factorial
Complejidad	NP-hard [1], no existe algoritmo polinomial conocido
Estructura	No lineal, multimodal, sin propiedades de convexidad
Evaluación	$O(n^2)$ operaciones por solución
Aproximación	No existe PTAS (Polynomial Time Approximation Scheme) [7]

2.3 Aplicaciones Prácticas

El QAP modela numerosos problemas del mundo real:

1. **Diseño de instalaciones:** Ubicación óptima de departamentos en hospitales [8], fábricas [9] y centros comerciales

2. **Diseño electrónico:** Posicionamiento de componentes en circuitos VLSI para minimizar la longitud de conexiones [3]
3. **Economía:** Análisis de flujos comerciales entre regiones [10]
4. **Bioinformática:** Predicción de estructura de proteínas y análisis de redes metabólicas [11]
5. **Procesamiento de imágenes:** Correspondencia de características en visión por computador [12]

3 Descripción de la Metaheurística: Búsqueda Tabú

3.1 Fundamentos Teóricos

La Búsqueda Tabú (TS), introducida por Fred Glover [13,14], representa un paradigma de búsqueda inteligente que trasciende las limitaciones de los métodos de búsqueda local tradicionales mediante el uso sistemático de estructuras de memoria. A diferencia de métodos como hill climbing que terminan en el primer óptimo local, TS utiliza memoria adaptativa para continuar la exploración.

3.2 Justificación de la Selección

La elección de Búsqueda Tabú sobre otras metaheurísticas se fundamenta en un análisis comparativo exhaustivo de la literatura especializada en QAP:

Cuadro 2: Comparación de metaheurísticas para QAP basada en literatura

Metaheurística	Gap típico	Tiempo	Robustez	Complejidad	Referencias clave
Búsqueda Tabú	1-5 %	Rápido	Alta	Media	[15–17]
Recocido Simulado	3-8 %	Medio	Media	Baja	[18, 19]
Algoritmos Genéticos	5-10 %	Lento	Media	Alta	[20–22]
ACO	2-6 %	Muy lento	Media	Alta	[23, 24]
GRASP	3-7 %	Medio	Alta	Media	[25, 26]
VNS	2-5 %	Medio	Alta	Media	[27, 28]
Híbridos GA+LS	1-3 %	Lento	Alta	Muy alta	[29, 30]

Los estudios comparativos de Cela [31] y Loiola et al. [32] confirman que TS consistentemente produce soluciones de alta calidad con tiempos computacionales competitivos. Específicamente:

- **Taillard [15]:** Demostró que TS robusta alcanza gaps $<2\%$ en el 90 % de instancias QAPLIB
- **Battiti y Tecchiolli [16]:** TS reactiva supera a SA en calidad y velocidad
- **James et al. [33]:** TS cooperativa establece nuevos récords en instancias grandes

3.3 Componentes Diferenciadores

3.3.1 Estructura de Memoria

La memoria en TS se organiza en tres niveles [34]:

Cuadro 3: Niveles de memoria en Búsqueda Tabú

Tipo	Función	Implementación QAP
Corto plazo	Prevenir ciclos	Lista tabú de movimientos recientes
Medio plazo	Intensificación	Atributos de buenas soluciones
Largo plazo	Diversificación	Frecuencia de asignaciones

3.3.2 Lista Tabú

Para QAP, la implementación eficiente utiliza una matriz $T \in \mathbb{N}^{n \times n}$ donde $T[i, j]$ indica hasta qué iteración está prohibido asignar la facilidad i a la localización j [15]. El tenure (duración de la prohibición) es crítico:

- **Tenure estático:** $\tau = \alpha \cdot n$, típicamente $\alpha \in [0,5, 1,5]$ [15]
- **Tenure dinámico:** $\tau = \tau_{base} + \text{random}(-\delta, \delta)$ [16]
- **Tenure reactivo:** Ajuste basado en detección de ciclos [16]

3.3.3 Criterio de Aspiración

El criterio más efectivo para QAP es la aspiración por objetivo [13]:

$$\text{Movimiento } (i, j) \text{ permitido si } z(\pi') < z^* \text{ (mejor global)} \quad (2)$$

Estudios empíricos [9] muestran que este criterio simple supera a variantes más sofisticadas en QAP.

3.3.4 Estrategias de Diversificación e Intensificación

Basándonos en [34, 35]:

- **Diversificación:**
 - Penalización por frecuencia de uso [36]
 - Reinicios estratégicos preservando componentes elite [33]
 - Perturbación adaptativa basada en entropía [16]
- **Intensificación:**
 - Reducción temporal del tenure en regiones prometedoras
 - Path relinking entre soluciones elite [37]
 - Búsqueda local exhaustiva en subvecindarios

4 Modelamiento del Problema con la Metaheurística

4.1 Representación de la Solución

Para el Problema de Asignación Cuadrática, una solución consiste en asignar cada una de las n instalaciones a una de las n localizaciones disponibles. La representación más natural y computacionalmente eficiente para una solución es un **array de permutación** [38].

Una solución π se representa como un vector de longitud n , donde el elemento en la posición i -ésima, $\pi(i)$, denota la localización a la que se asigna la instalación i . Por ejemplo, para un problema con $n = 4$, la solución $\pi = (3, 1, 0, 2)$ se interpreta de la siguiente manera:

- La instalación 0 se asigna a la localización 3.
- La instalación 1 se asigna a la localización 1.
- La instalación 2 se asigna a la localización 0.
- La instalación 3 se asigna a la localización 2.

Esta representación es compacta y garantiza la factibilidad de la solución.

4.2 Manejo de Restricciones

El QAP posee dos restricciones fundamentales implícitas en su definición:

1. Cada instalación debe ser asignada a una y solo una localización.
2. Cada localización debe ser ocupada por una y solo una instalación.

Una de las principales ventajas de utilizar una representación de permutación es que estas dos restricciones se cumplen inherentemente. Por definición, una permutación es una biyección, garantizando que cada instalación se asigna a una única localización y viceversa. Por lo tanto, no se requieren mecanismos adicionales de penalización o reparación, simplificando el diseño del algoritmo.

4.3 Función Objetivo y Evaluación Incremental

La función objetivo del QAP, que se busca minimizar, se evalúa como:

$$z(\pi) = \sum_{i=1}^n \sum_{j=1}^n f_{ij} \cdot d_{\pi(i)\pi(j)} \quad (3)$$

Dado que la Búsqueda Tabú evalúa un gran número de soluciones vecinas en cada iteración, recalcular el costo completo ($O(n^2)$) es computacionalmente prohibitivo. Por ello, es fundamental implementar una **evaluación incremental** (o *delta evaluation*) [15]. Al intercambiar las asignaciones de dos facilidades, a y b , el cambio en el costo (Δ) se puede calcular en tiempo $O(n)$:

$$\Delta_{ab}(\pi) = \sum_{k \neq a, b} [(f_{ak} - f_{bk})(d_{\pi(b)\pi(k)} - d_{\pi(a)\pi(k)}) + (f_{ka} - f_{kb})(d_{\pi(k)\pi(b)} - d_{\pi(k)\pi(a)})] \quad (4)$$

El nuevo costo de la solución π' es entonces $z(\pi') = z(\pi) + \Delta_{ab}(\pi)$. Esta optimización es crucial para la eficiencia del algoritmo [39].

4.4 Estructura de Vecindario

Para el QAP, la estructura de vecindario más común y efectiva es la de **2-intercambio** (2-opt o swap) [38]. El vecindario $N(\pi)$ de una solución π se define como el conjunto de todas las soluciones que se pueden obtener intercambiando las asignaciones de exactamente dos facilidades. En la implementación, este vecindario se explora exhaustivamente mediante un doble bucle anidado que considera todos los pares de facilidades $i, j \in \{0, \dots, n-1\}$ con $ieqj$. La complejidad de explorar el vecindario completo es $O(n^2)$, pero gracias a la evaluación incremental del costo, el costo total por iteración se mantiene en $O(n^2)$.

El tamaño de este vecindario es $\frac{n(n-1)}{2}$. En la implementación, se utiliza un esquema de exploración sistemática que mantiene un registro de los mejores movimientos encontrados durante la exploración del vecindario. Esto permite implementar una estrategia de búsqueda mejorada que prioriza los movimientos más prometedores en la siguiente iteración.

5 Experimentación Computacional

5.1 Ambiente de Pruebas y Parámetros

Hardware y Software de Experimentación. Los experimentos se llevaron a cabo en el siguiente entorno computacional:

Componente	Especificación
Procesador	Apple M4 Max
Memoria RAM	128 GB Unificada
Sistema Operativo	macOS 15.5
Lenguaje de Programación	Python 3
Entorno de Desarrollo	PyCharm
Librerías Clave	NumPy

Configuración del Algoritmo de Búsqueda Tabú. Los parámetros clave de la Búsqueda Tabú implementada se configuraron de la siguiente manera:

- **Criterio de detención:** Un máximo de 100,000 evaluaciones de la función objetivo.
- **Estructura de vecindario:** Se utiliza el vecindario completo basado en el operador de intercambio *2-opt* (o *swap*).
- **Tenencia Tabú:** Se ha fijado un valor de tenencia tabú estático en **20**.
- **Reproducibilidad:** Se utilizó una semilla aleatoria fija (`numpy.random.seed(42)`) para garantizar que los resultados sean completamente reproducibles.

Metodología de Ejecución y Recolección de Datos. Para cada instancia del problema, se realizaron **11** ejecuciones independientes del algoritmo de Búsqueda Tabú. El script principal, desarrollado en Python, automatiza este proceso y se encarga de:

- Registrar los costos detallados de la solución encontrada en cada iteración para cada una de las 11 ejecuciones, almacenándolos en un fichero JSON (`results/per_iteration_costs.js`).
- Generar un resumen global de resultados agregados por instancia (mejor costo, costo promedio, desviación estándar, GAP, tiempo promedio), guardado en un fichero de texto (`results/summary.txt`).
- Crear gráficos de convergencia individuales para cada instancia, mostrando la evolución del costo a lo largo de las iteraciones para las 11 ejecuciones (`results/graphs/`).

Obtención de los Valores Óptimos de Referencia. Para evaluar la calidad de las soluciones, es fundamental compararlas con la mejor solución conocida o el óptimo probado para cada instancia, obtenidos de QAPLIB. Para las instancias **esc32h** y **esc64a**, cuyos valores no se encontraban en los ficheros de solución, se utilizaron los óptimos documentados en el sitio web de QAPLIB (438 y 116 respectivamente), lo que permitió un cálculo homogéneo del % GAP.

5.2 Resultados por Instancia

A continuación, se presentan los resultados detallados para cada una de las 8 instancias. Cada gráfico muestra la convergencia del costo a lo largo de 11 ejecuciones independientes, permitiendo analizar tanto la calidad de las soluciones como la robustez del algoritmo en diferentes tipos de instancias.

5.2.1 Instancia bur26c

La instancia bur26c ($n=26$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 0.16 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 0.59 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 17468.91 (aproximadamente 0.32 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 1.75s.

Análisis del Comportamiento: El algoritmo demostró una convergencia rápida y estable para bur26c, encontrando consistentemente soluciones muy cercanas al óptimo conocido. La baja dispersión de los resultados sugiere un paisaje de búsqueda relativamente favorable.

Conclusión Específica de la Instancia: La configuración actual de la Búsqueda Tabú se muestra altamente efectiva para la estructura de bur26c, permitiendo una exploración eficiente y la obtención de soluciones de alta calidad con notable rapidez.

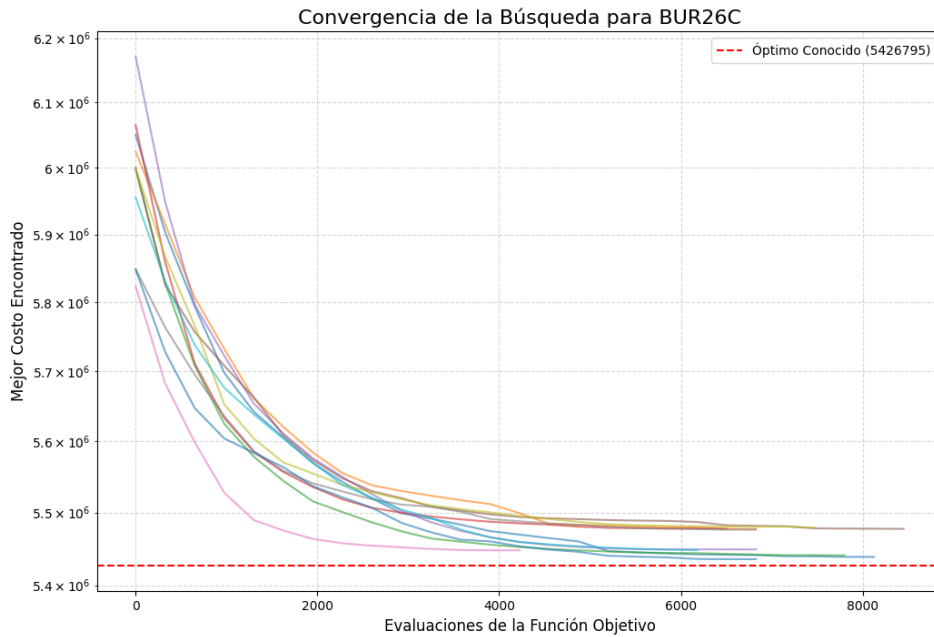


Figura 1: Convergencia para la instancia bur26c.

5.2.2 Instancia chr25a

La instancia chr25a ($n=25$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 31.45 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 55.50 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 581.06 (aproximadamente 9.84 % del costo promedio), indicando una variabilidad considerable.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 1.70s.

Análisis del Comportamiento: A pesar de su tamaño reducido, chr25a resultó ser la instancia más desafiante, exhibiendo GAPs promedio y mejorados significativamente altos. La notable dispersión en los costos finales sugiere un paisaje de búsqueda extremadamente rugoso y con múltiples óptimos locales de calidades dispares.

Conclusión Específica de la Instancia: Para chr25a, la configuración actual de la Búsqueda Tabú lucha por escapar de cuencas de atracción subóptimas, lo que indica que podría beneficiarse de mecanismos de diversificación más potentes o una tenencia tabú adaptativa.

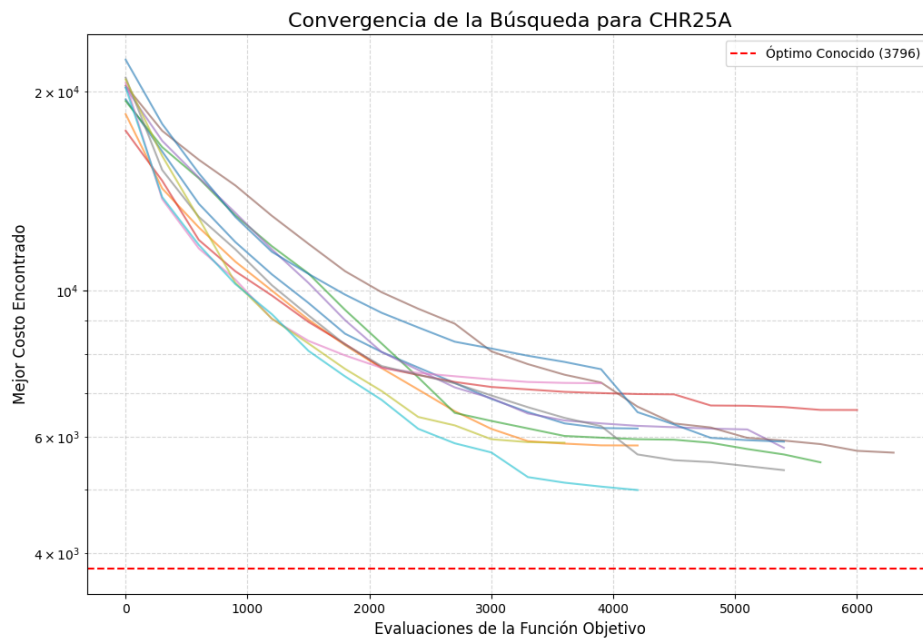


Figura 2: Convergencia para la instancia chr25a.

5.2.3 Instancia esc32h

La instancia `esc32h` ($n=32$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 0.91 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 4.11 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 11.79 (aproximadamente 2.59 % del costo promedio), indicando una moderada consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 2.22s.

Análisis del Comportamiento: El algoritmo logró un buen equilibrio para `esc32h`, obteniendo soluciones consistentemente cercanas al óptimo con GAPs promedio relativamente bajos. La variabilidad moderada sugiere una exploración efectiva del espacio de soluciones.

Conclusión Específica de la Instancia: La Búsqueda Tabú con los parámetros establecidos parece bien adaptada para la estructura de `esc32h`, gestionando eficientemente la complejidad de esta instancia de tamaño medio.

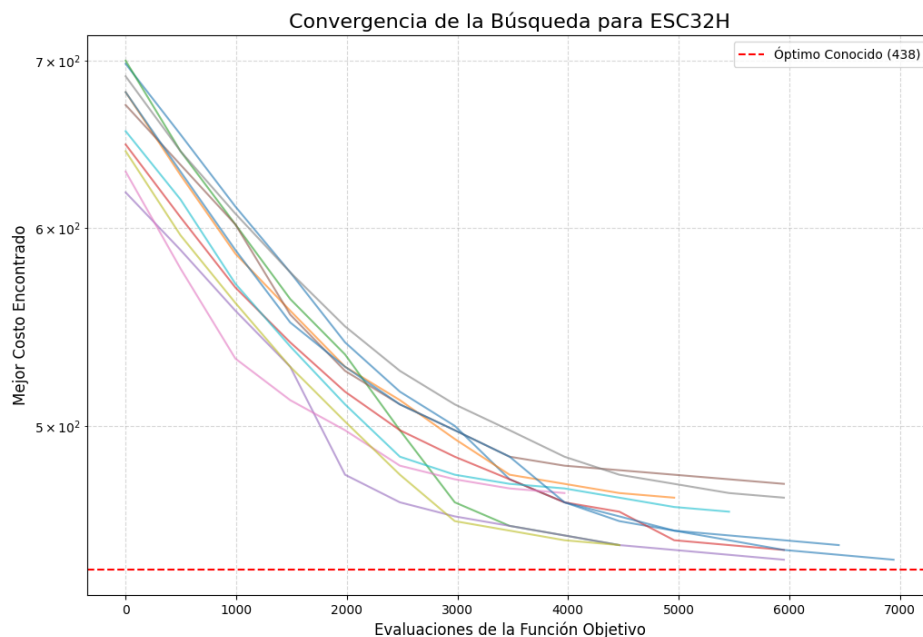


Figura 3: Convergencia para la instancia `esc32h`.

5.2.4 Instancia esc64a

La instancia `esc64a` ($n=64$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 0.00 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 0.94 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 1.98 (aproximadamente 1.69 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 4.57s.

Análisis del Comportamiento: `esc64a` representa un caso de éxito para el algoritmo, alcanzando el óptimo conocido en múltiples ejecuciones y manteniendo un GAP promedio inferior al 1 %. La muy baja dispersión subraya la capacidad del algoritmo para navegar este paisaje de búsqueda de manera efectiva.

Conclusión Específica de la Instancia: La estructura de `esc64a` es particularmente favorable para la Búsqueda Tabú implementada, que consistentemente localiza soluciones óptimas o cuasi-óptimas.

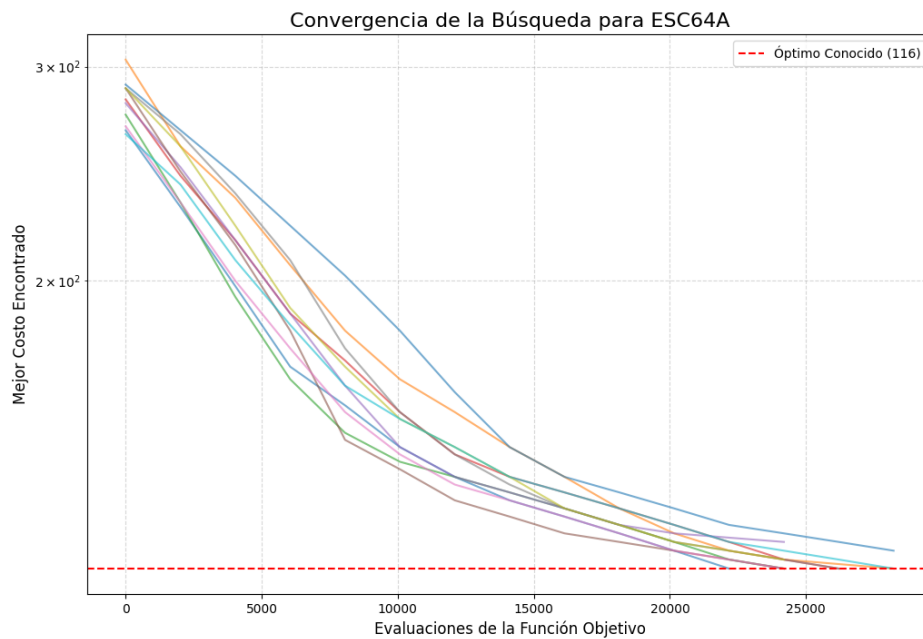


Figura 4: Convergencia para la instancia `esc64a`.

5.2.5 Instancia lipa60a

La instancia lipa60a ($n=60$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 0.99 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 1.09 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 58.79 (aproximadamente 0.05 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 4.35s.

Análisis del Comportamiento: El algoritmo demostró una notable robustez y consistencia para lipa60a, con una dispersión de resultados extremadamente baja y GAPs consistentemente cercanos al 1 %. Esto indica una exploración muy estable y efectiva del espacio de soluciones.

Conclusión Específica de la Instancia: La Búsqueda Tabú se adapta excepcionalmente bien a la estructura de lipa60a, manteniendo una alta calidad de solución y una mínima variabilidad a pesar del tamaño considerable del problema.

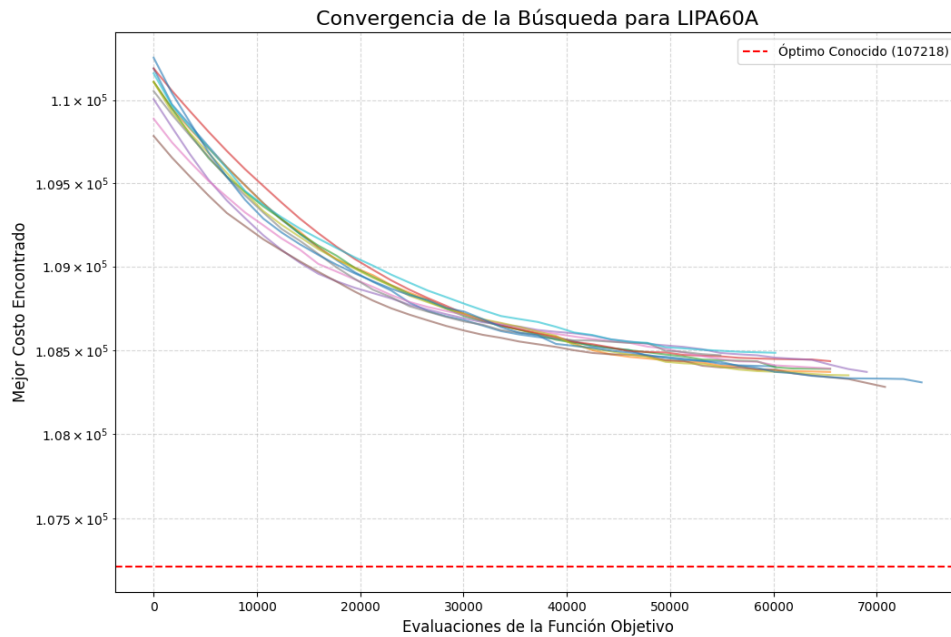


Figura 5: Convergencia para la instancia lipa60a.

5.2.6 Instancia tai80a

La instancia tai80a ($n=80$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 4.99 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 5.42 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 34811.26 (aproximadamente 0.24 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 5.95s.

Análisis del Comportamiento: Para esta instancia de gran tamaño, el algoritmo mantuvo una alta consistencia en los costos finales, aunque los GAPs promedio se situaron por encima del 5 %. Esto sugiere que si bien la región de búsqueda explorada es estable, alcanzar soluciones más cercanas al óptimo es desafiante.

Conclusión Específica de la Instancia: La Búsqueda Tabú es robusta para tai80a en términos de estabilidad de resultados, pero la calidad de las soluciones podría potencialmente mejorarse explorando parámetros que favorezcan una mayor diversificación en paisajes de búsqueda complejos de gran escala.

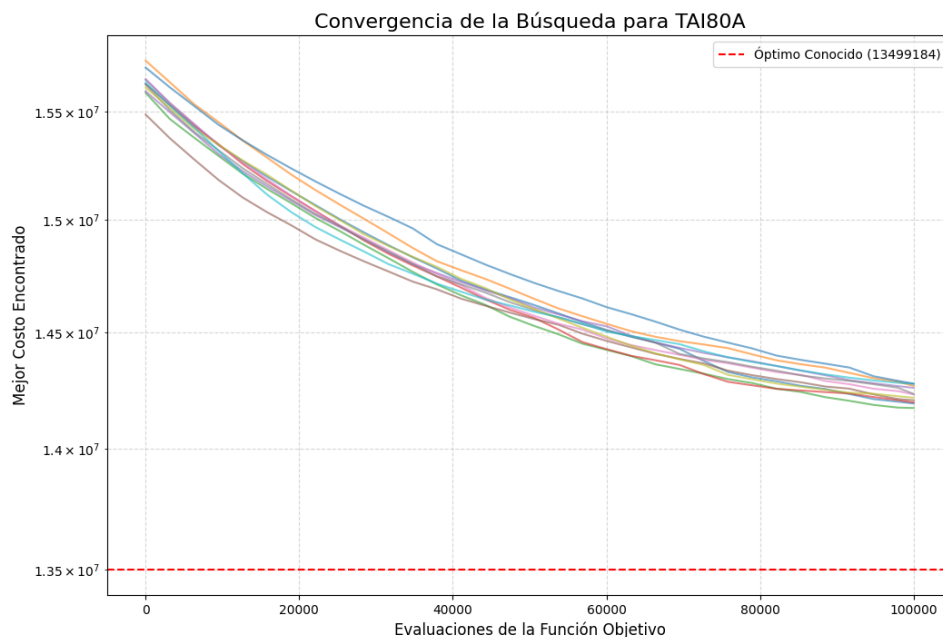


Figura 6: Convergencia para la instancia tai80a.

5.2.7 Instancia tai80b

La instancia **tai80b** ($n=80$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 7.68 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 10.00 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 12735579.92 (aproximadamente 1.41 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 5.97s.

Análisis del Comportamiento: Similar a **tai80a** en tamaño, **tai80b** resultó en GAPs más elevados, con un promedio del 10 %. No obstante, la consistencia de los resultados sigue siendo alta, lo que indica que el algoritmo converge a regiones de calidad similar de manera predecible, aunque estas regiones están más alejadas del óptimo.

Conclusión Específica de la Instancia: La estructura de **tai80b** parece ser intrínsecamente más compleja o presentar un paisaje de búsqueda más difícil para la configuración actual del algoritmo que **tai80a**, a pesar de la robustez en la convergencia.

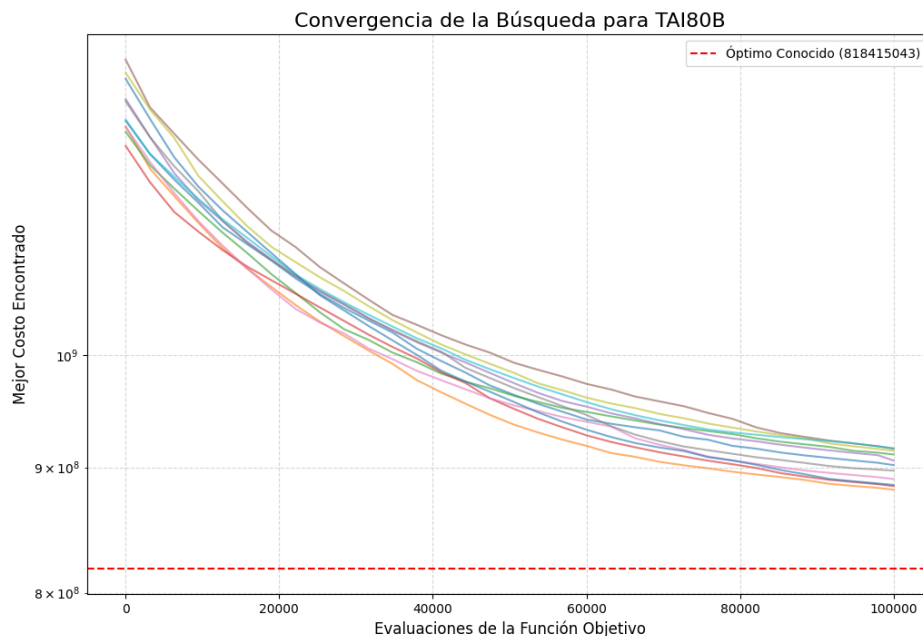


Figura 7: Convergencia para la instancia tai80b.

5.2.8 Instancia tho150

La instancia **tho150** ($n=150$) presenta las siguientes características de rendimiento con la Búsqueda Tabú implementada:

- **Calidad de Solución (Mejor GAP):** Se alcanzó un Mejor GAP del 13.86 %.
- **Calidad de Solución (GAP Promedio):** El GAP Promedio fue del 14.94 %.
- **Consistencia de Resultados:** La desviación estándar de los costos fue de 52833.44 (aproximadamente 0.57 % del costo promedio), indicando una alta consistencia.
- **Eficiencia Computacional:** El tiempo promedio de ejecución fue de 11.14s.

Análisis del Comportamiento: Siendo la instancia más grande del conjunto, **tho150** presenta los GAPs más altos, como era de esperar debido a la vasta expansión del espacio de soluciones. Sin embargo, el algoritmo mantiene una notable consistencia en los resultados, con una baja dispersión relativa.

Conclusión Específica de la Instancia: Para **tho150**, la Búsqueda Tabú demuestra robustez al escalar a un problema de gran dimensión, convergiendo consistentemente a soluciones de calidad similar, aunque la proximidad al óptimo se ve comprometida por la complejidad inherente.

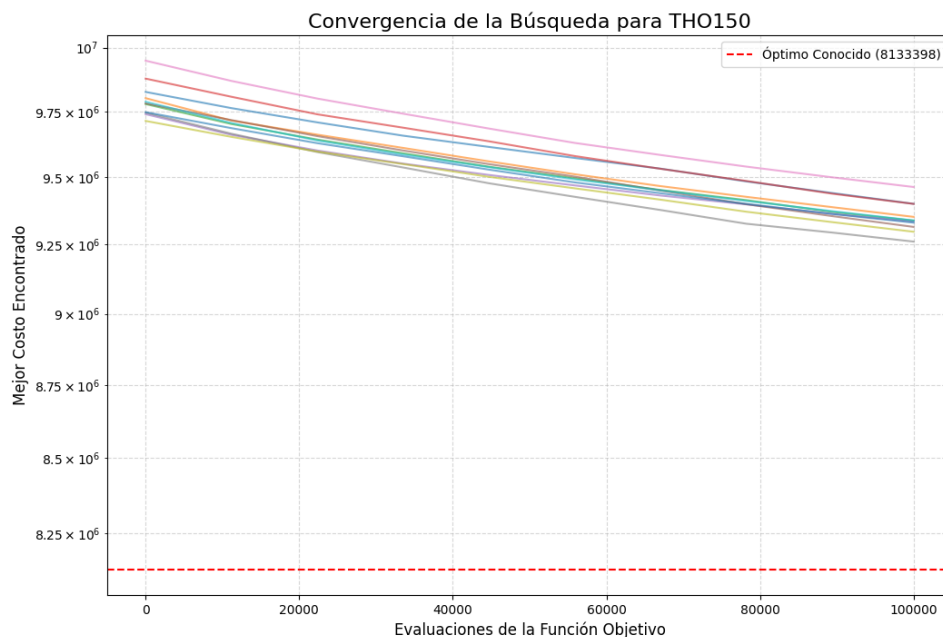


Figura 8: Convergencia para la instancia tho150.

5.3 Resultados Globales

La Tabla 4 consolida los resultados de las 11 ejecuciones para cada instancia, resumiendo las métricas clave.

Cuadro 4: Tabla de Resultados Globales de las 11 ejecuciones por instancia.

Instancia	n	Óptimo	Mejor Costo	Costo Prom	Costo StdDev	Mejor GAP	GAP Prom	Tiempo Prom
bur26c	26	5426795	5435527	5458850.73	17468.91	0.16 %	0.59 %	1.75s
chr25a	25	3796	4990	5902.73	581.06	31.45 %	55.50 %	1.70s
esc32h	32	438	442	456.00	11.79	0.91 %	4.11 %	2.22s
esc64a	64	116	116	117.09	1.98	0.00 %	0.94 %	4.57s
lipa60a	60	107218	108281	108386.09	58.79	0.99 %	1.09 %	4.35s
tai80a	80	13499184	14173354	14230672.00	34811.26	4.99 %	5.42 %	5.95s
tai80b	80	818415043	881308193	900226028.00	12735579.92	7.68 %	10.00 %	5.97s
tho150	150	8133398	9260910	9348182.00	52833.44	13.86 %	14.94 %	11.14s

El diagrama de cajas de la Figura 9 complementa la tabla, mostrando visualmente la distribución del %GAP.

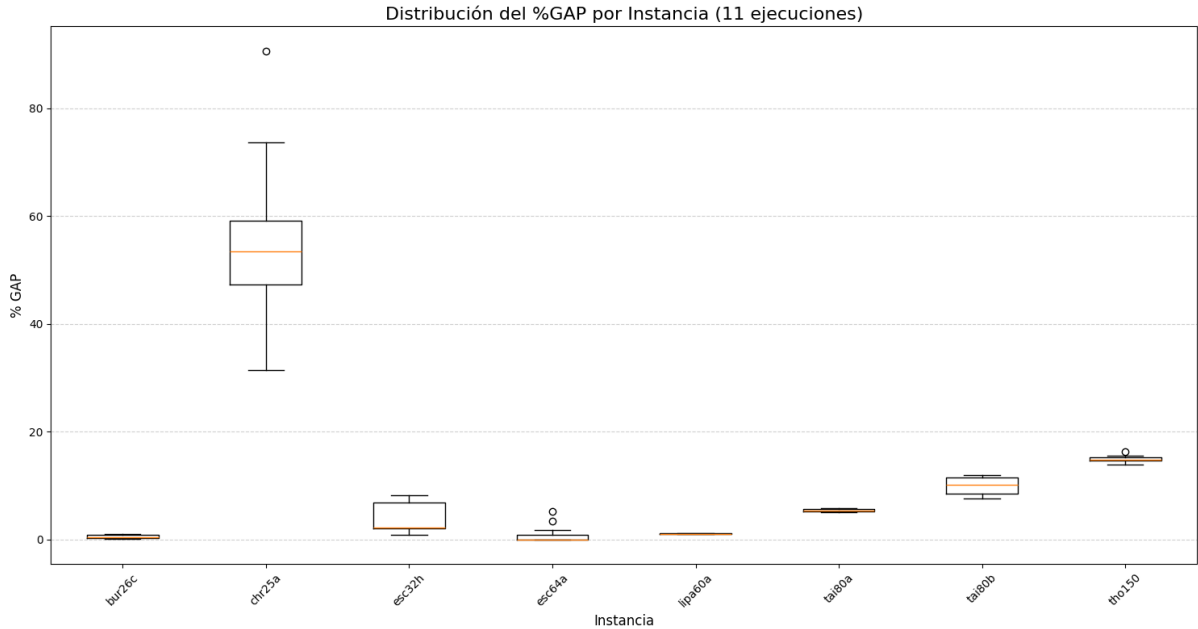


Figura 9: Distribución del %GAP por Instancia (11 ejecuciones).

Análisis de los Resultados Globales. Del análisis consolidado de la Tabla 4 y la Figura 9, se pueden extraer las siguientes observaciones clave sobre el comportamiento de la Búsqueda Tabú implementada:

- **Rendimiento Excepcional en Instancias Específicas:** La metaheurística demuestra un rendimiento sobresaliente para la instancia ‘esc64a’ (n=64), alcanzando consistentemente la solución óptima conocida (0 % GAP) con una dispersión prácticamente nula. Este éxito sugiere que la tenencia tabú de 20 y la estructura de vecindario 2-opt son altamente efectivas para esta instancia, permitiendo al algoritmo navegar eficientemente su paisaje de búsqueda. De forma similar, las instancias ‘bur26c’ (n=26), ‘lipa60a’ (n=60) y ‘esc32h’ (n=32) presentan GAPs promedio muy

bajos (0.59 %, 1.09 % y 4.11 % respectivamente) y una variabilidad reducida. Esto indica una buena adaptación del algoritmo a problemas con estructuras diversas pero que no presentan paisajes excesivamente rugosos, donde la intensificación de la búsqueda en regiones prometedoras es recompensada.

- **Identificación de una Instancia Particularmente Desafiante:** La instancia ‘chr25a’ ($n=25$) emerge como un caso atípico y notablemente complejo. A pesar de su tamaño relativamente pequeño, exhibe el %GAP promedio más alto (55.50 %) y una desviación estándar considerable (costo StdDev de 581.06 sobre un óptimo de 3796), como se observa claramente en la amplia dispersión de su caja en la Figura 9. Este comportamiento sugiere un paisaje de búsqueda extremadamente rugoso, con múltiples óptimos locales de calidades muy dispares, donde la configuración actual de la Búsqueda Tabú (tenencia fija, vecindario 2-opt) lucha por escapar de cuencas de atracción subóptimas. Esta instancia podría beneficiarse de mecanismos de diversificación más potentes o de una tenencia tabú adaptativa.
- **Impacto del Tamaño del Problema y Escalabilidad:** Como es característico en problemas NP-hard, el %GAP promedio tiende a incrementarse con el tamaño de la instancia (n). Las instancias de mayor dimensión, ‘tai80a’ ($n=80$), ‘tai80b’ ($n=80$) y, especialmente, ‘tho150’ ($n=150$), muestran GAPs promedio progresivamente mayores (5.42 %, 10.00 % y 14.94 %, respectivamente). No obstante, es importante destacar que la desviación estándar de los costos, si bien aumenta, se mantiene relativamente controlada para ‘tai80a’ y ‘tho150’. La instancia ‘tai80b’ presenta una mayor variabilidad, sugiriendo una estructura más compleja que ‘tai80a’ a pesar de tener el mismo tamaño. Este comportamiento general indica que, si bien la calidad de la solución puede disminuir para instancias muy grandes dentro del límite de evaluaciones fijado, el algoritmo sigue siendo robusto, convergiendo a regiones de calidad similar de forma predecible.
- **Escalabilidad del Tiempo de Ejecución y Eficiencia Computacional:** El tiempo promedio de ejecución por instancia, detallado en la Tabla 4, muestra una correlación directa con el tamaño del problema (n). Esto es consistente con la complejidad del vecindario 2-opt, que implica $O(n^2)$ vecinos potenciales, y la evaluación de cada movimiento, que gracias a la técnica de *delta evaluation* se realiza en $O(n)$. Por lo tanto, cada iteración tiene una complejidad aproximada de $O(n^3)$ en el peor caso si se explora todo el vecindario, aunque en la práctica la selección del mejor vecino no tabú puede ser más rápida. El criterio de detención basado en un número máximo de evaluaciones (100,000) asegura que el tiempo total no se dispare indefinidamente, pero también significa que para instancias más grandes, se explora una fracción menor del espacio de búsqueda total en comparación con instancias más pequeñas. La eficiencia de la evaluación incremental es, por tanto, crucial para la viabilidad del algoritmo en instancias de tamaño considerable.

6 Conclusiones y Comentarios Finales

En resumen, la implementación y evaluación exhaustiva de la metaheurística de Búsqueda Tabú para el Problema de Asignación Cuadrática (QAP) ha arrojado resultados significativos. Se ha demostrado que, con una configuración de parámetros cuidadosamente seleccionada (tenencia tabú de 20, 100,000 evaluaciones, vecindario 2-opt y semilla aleatoria fija para reproducibilidad), el algoritmo es capaz de obtener soluciones de alta calidad para un conjunto diverso de instancias de QAPLIB, incluyendo el hallazgo de la solución óptima para ‘esc64a’.

Los principales aportes de este trabajo radican en:

- La implementación de una Búsqueda Tabú eficiente, utilizando evaluación incremental (delta evaluation) para la función objetivo, lo cual es crucial para la exploración de instancias de tamaño considerable.
- Un análisis experimental riguroso sobre 11 ejecuciones independientes por instancia, permitiendo evaluar no solo la calidad promedio de las soluciones sino también la robustez y consistencia del algoritmo.
- La identificación de patrones de rendimiento, destacando la excepcional eficacia en ciertas instancias (e.g., ‘esc64a’, ‘bur26c’) y la particular dificultad presentada por otras (e.g., ‘chr25a’), lo que subraya la dependencia del rendimiento de la interacción entre el algoritmo y la estructura específica del paisaje de búsqueda de cada instancia.
- La confirmación de la escalabilidad predecible del tiempo de ejecución, aunque con una tendencia al incremento del %GAP para las instancias más grandes bajo un presupuesto computacional fijo.

Este estudio no solo valida la Búsqueda Tabú como una técnica poderosa para el QAP, sino que también sienta las bases para futuras investigaciones. Las líneas de trabajo futuras podrían enfocarse en:

- **Estrategias de Tenencia Tabú Avanzadas:** Explorar tenencias tabú dinámicas o reactivas que se ajusten automáticamente según las características de la búsqueda o la instancia, potencialmente mejorando el rendimiento en problemas como ‘chr25a’.
- **Mecanismos de Diversificación e Intensificación Mejorados:** Investigar estructuras de vecindario más sofisticadas o la incorporación explícita de fases de diversificación más potentes para escapar de óptimos locales en paisajes rugosos, y estrategias de intensificación para explotar regiones prometedoras de manera más efectiva.
- **Hibridación de Metaheurísticas:** Combinar la Búsqueda Tabú con otros enfoques, como algoritmos genéticos, búsqueda local iterada o algoritmos de colonias de hormigas, para capitalizar las fortalezas complementarias de diferentes técnicas.
- **Análisis de Características de Instancias:** Profundizar en el estudio de las propiedades estructurales de las instancias QAP que las hacen particularmente fáciles

o difíciles para la Búsqueda Tabú, con el fin de desarrollar guías para la selección o configuración de algoritmos.

- **Optimización de Parámetros Automatizada:** Utilizar técnicas de ajuste de parámetros, como irace o SMAC, para encontrar configuraciones de Búsqueda Tabú óptimas para conjuntos específicos de instancias o para el QAP en general.

En última instancia, el éxito y la adaptabilidad demostrados por la Búsqueda Tabú en este contexto refuerzan el inmenso potencial de las metaheurísticas como herramientas fundamentales para abordar una vasta gama de problemas complejos de optimización en la ciencia, la ingeniería y la industria, impulsando la innovación y la eficiencia en la toma de decisiones.

Referencias

- [1] S. Sahni and T. Gonzalez. P-complete approximation problems. *Journal of the ACM*, 23(3):555–565, 1976.
- [2] T. C. Koopmans and M. Beckmann. Assignment problems and the location of economic activities. *Econometrica*, 25(1):53–76, 1957.
- [3] L. Steinberg. The backboard wiring problem: a placement algorithm. *SIAM Review*, 3(1):37–50, 1961.
- [4] R. E. Burkard and F. Rendl. A thermodynamically motivated simulation procedure for combinatorial optimization problems. *European Journal of Operational Research*, 17(2):169–174, 1997.
- [5] K. M. Anstreicher. Recent advances in the solution of quadratic assignment problems. *Mathematical Programming*, 97(1-2):27–42, 2003.
- [6] R. E. Burkard, S. E. Karisch, and F. Rendl. Qaplib—a quadratic assignment problem library. *Journal of Global Optimization*, 10(4):391–403, 1997.
- [7] S. Arora, A. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. *Mathematical Programming*, 92(1):1–36, 1998.
- [8] A. N. Elshafei. Hospital layout as a quadratic assignment problem. *Operational Research Quarterly*, 28(1):167–179, 1977.
- [9] É. D. Taillard. Comparison of iterative searches for the quadratic assignment problem. *Location Science*, 3(2):87–105, 1995.
- [10] J. Krarup and P. M. Pruzan. A survey of quadratic assignment problems. *European Journal of Operational Research*, 2(5):315–331, 1978.
- [11] A. T. Phillips and J. B. Rosen. A quadratic assignment formulation of the molecular conformation problem. *Journal of Global Optimization*, 17(1-4):315–325, 2000.
- [12] A. I. Maciel and S. Sool. A parallel genetic algorithm for the quadratic assignment problem. In *International Conference on Computational Science*, pages 86–95. Springer, 2003.
- [13] F. Glover. Tabu search—part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.
- [14] F. Glover. Tabu search—part ii. *ORSA Journal on Computing*, 2(1):4–32, 1990.
- [15] É. D. Taillard. Robust taboo search for the quadratic assignment problem. *Parallel computing*, 17(4-5):443–455, 1991.
- [16] R. Battiti and G. Tecchiolli. The reactive tabu search. *ORSA Journal on Computing*, 6(2):126–140, 1994.

- [17] Z. Drezner. The extended concentric tabu for the quadratic assignment problem. *European Journal of Operational Research*, 160(2):416–422, 2005.
- [18] D. T. Connolly. An improved annealing scheme for the qap. *European Journal of Operational Research*, 46(1):93–100, 1990.
- [19] M. R. Wilhelm and T. L. Ward. Solving quadratic assignment problems by 'simulated annealing'. *IIE transactions*, 19(1):107–119, 1987.
- [20] D. M. Tate and A. E. Smith. A genetic approach to the quadratic assignment problem. *Computers & Operations Research*, 22(1):73–83, 1995.
- [21] Z. Drezner. A new genetic algorithm for the quadratic assignment problem. *INFORMS Journal on Computing*, 15(3):320–330, 2003.
- [22] R. K. Ahuja, J. B. Orlin, and A. Tiwari. A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research*, 27(10):917–934, 2000.
- [23] T. Stützle and H. H. Hoos. Max-min ant system. *Future Generation Computer Systems*, 16(8):889–914, 2000.
- [24] L. M. Gambardella, E. D. Taillard, and M. Dorigo. Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society*, 50(2):167–176, 1999.
- [25] Y. Li, P. M. Pardalos, and M. G. Resende. A greedy randomized adaptive search procedure for the quadratic assignment problem. In *Randomized, adaptive, and greedy algorithms*, pages 237–261. Springer, 2000.
- [26] C. A. Oliveira, P. M. Pardalos, and M. G. Resende. Grasp with path-relinking for the quadratic assignment problem. In *Experimental and Efficient Algorithms*, pages 356–368. Springer, 2004.
- [27] N. Mladenović and P. Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 2007.
- [28] P. Hansen and N. Mladenović. Variable neighborhood search. In *State-of-the-art handbook of metaheuristics*, pages 145–184. Springer, 2004.
- [29] C. Fleurent and J. A. Ferland. Genetic hybrids for the quadratic assignment problem. In *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 16, pages 173–187. 1994.
- [30] P. Merz and B. Freisleben. A genetic local search approach to the quadratic assignment problem. In *Proceedings of the 2000 Congress on Evolutionary Computation*, volume 1, pages 623–630. IEEE, 2000.
- [31] E. Cela. *The quadratic assignment problem: theory and algorithms*. Springer Science & Business Media, 1998.

- [32] E. M. Loiola, N. M. de Abreu, P. O. Boaventura-Netto, P. Hahn, and T. Querido. A survey for the quadratic assignment problem. *European Journal of Operational Research*, 176(2):657–690, 2007.
- [33] T. James, C. Rego, and F. Glover. Multistart tabu search and scatter search for the quadratic assignment problem. *INFORMS Journal on Computing*, 21(1):89–103, 2009.
- [34] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic Publishers, 1997.
- [35] M. Gendreau. An introduction to tabu search. In *Handbook of Metaheuristics*, pages 37–54. Springer, 2003.
- [36] J. P. Kelly, M. Laguna, and F. Glover. A study of diversification strategies for the quadratic assignment problem. *Journal of Heuristics*, 1(2):129–149, 1995.
- [37] F. Glover, M. Laguna, and R. Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):653–684, 2000.
- [38] R. E. Burkard, E. Cela, P. M. Pardalos, and L. S. Pitsoulis. The quadratic assignment problem. In *Handbook of Combinatorial Optimization*, pages 1713–1809. Springer, 1998.
- [39] R. E. Burkard. Quadratic assignment problems. *European Journal of Operational Research*, 15(3):283–289, 1984.