Informe Técnico – Taller Práctico 1: Procesamiento Paralelo y el Problema del Viajante

cesar mora - Johan García

1. Título y Objetivos

Título: Comparación de Rendimiento entre Algoritmos Secuencial y Paralelo para el Problema del Viajante (TSP) mediante Fuerza Bruta
Objetivo: Implementar y comparar la eficiencia de un algoritmo secuencial y otro paralelo (con múltiples procesos) para resolver el TSP por fuerza bruta, evaluando el speedup (aceleración) obtenido al incrementar el número de ciudades.

2. Marco Teórico

El Problema del Viajante (TSP) 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, cuya complejidad crece factorialmente: para N ciudades, existen (N-1)!/2 rutas posibles en el caso simétrico.

Aunque existen heurísticas eficientes, la fuerza bruta explora todas las combinaciones, lo que la hace ideal para estudiar paralelismo: cada ruta se puede evaluar de forma independiente, permitiendo una división directa del trabajo entre procesos. Sin embargo, el overhead de crear procesos puede superar los beneficios si el problema es pequeño.

3. Metodología

Datos
Se utilizó la siguiente lista fija de 8 ciudades (coordenadas en el plano):

[[0, 0], [1, 2], [3, 1], [5, 4], [2, 6], [4, 3], [6, 7], [8, 2]]

Además, se probaron tamaños aleatorios (N = 6, 7, 8) con semilla fija en el script de comparación.

Hardware

Núcleos lógicos: 20

Sistema operativo: Windows

CPU: Procesador de alto rendimiento (ej. Intel i9 o Ryzen 9)

Software

Lenguaje: Python 3.12

Bibliotecas: itertools, multiprocessing, math, random

Entorno: Ejecución local desde terminal

Algoritmos Desarrollados

Secuencial: Genera todas las permutaciones de ciudades (fijando la ciudad 0 como inicio), calcula la distancia de cada ruta y retorna la mínima.

Paralelo: Divide las permutaciones entre múltiples procesos usando multiprocessing.Pool. Cada proceso evalúa un subconjunto de rutas y retorna su mejor candidato. El proceso principal selecciona el mínimo global.

4. Resultados

Ejecución con ciudades fijas (N = 8)

Algoritmo	Mejor Ruta	Distancia	Tiempo (s)
Secuencial	[0, 1, 4, 6, 7, 3, 5, 2]	26.2856	0.0097
Paralelo	[0, 1, 4, 6, 7, 3, 5, 2]	26.2856	0.2298

Comparación con ciudades aleatorias

N	Tiempo Secuencial (s)	Tiempo Paralelo (s)	Speedup	Distancia
6	0.0003	0.1648	0.00×	316.83
7	0.0018	0.1478	0.01×	317.55
8	0.0140	0.1596	0.09×	322.83

Nota: Speedup = Tiempo secuencial / Tiempo paralelo. Valores < 1.0 indican que el paralelo es más lento.

5. Análisis de Rendimiento

El algoritmo secuencial es significativamente más rápido para $N \leq 8$.

El paralelo introduce overhead debido a:
• Creación y gestión de procesos (costoso en Windows)
• Serialización/deserialización de datos (cities y permutaciones)
• Sincronización al recolectar resultados

Para problemas pequeños, el tiempo de cómputo puro es muy bajo (milisegundos), por lo que el overhead domina el tiempo total.

El speedup negativo es esperado en este rango de N. El paralelismo solo es rentable cuando el tiempo de cómputo por ruta es mucho mayor que el overhead (ej. $N \geq 10$ o funciones de distancia complejas).

Conclusión técnica: el paralelismo no siempre acelera un programa. Depende del tamaño del problema, el costo de la unidad de trabajo y la arquitectura del sistema.

6. Conclusiones

Se implementaron correctamente ambos algoritmos de fuerza bruta para el TSP, con y sin paralelismo.

Se verificó que ambos producen la misma solución óptima, garantizando la corrección.

Se observó que, para instancias pequeñas ($N \leq 8$), el algoritmo secuencial supera al paralelo debido al overhead de la creación de procesos.

Este resultado no invalida el paralelismo, sino que ilustra un principio clave en computación paralela: la granularidad del trabajo importa.

En problemas reales con $N > 10$ o con funciones de costo más costosas (ej. distancia geodésica, APIs externas), el paralelismo sí mostraría speedup positivo.

Este taller demuestra de forma práctica cuándo y por qué el paralelismo puede (o no) mejorar el rendimiento