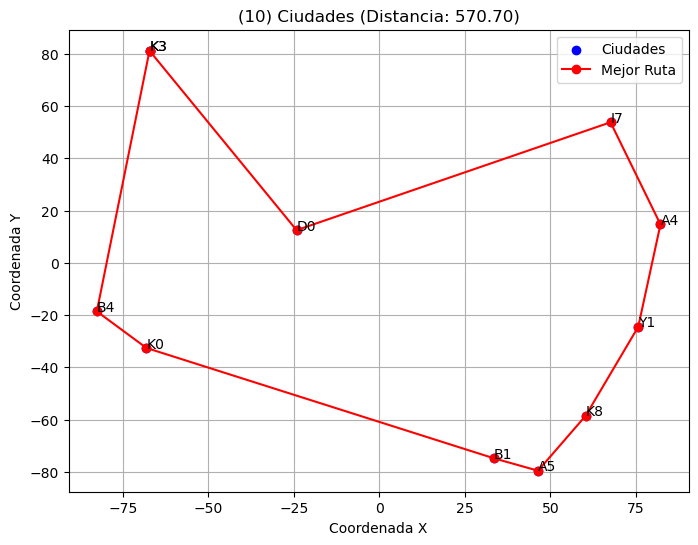
1. **Analizar el código propuesto**



A graph with red lines and numbers

AI-generated content may be incorrect.

A graph with red lines and numbers

AI-generated content may be incorrect.

A graph with red lines and numbers

AI-generated content may be incorrect.

A graph with red lines and numbers

AI-generated content may be incorrect.

Como se puede observar en las corridas anteriores. El número de ciudades es el factor diferencial al momento de tomar tiempos ya que aumenta el procesamiento de la información. Como el tiempo límite sigue establecido a 30 segundos, todos los ejemplos tienen el mismo límite de tiempo.

Al no usarse ninguna heurística, es probable que el algoritmo base sea aleatorio, Greedy sin mejora, o alguna forma de recorrido sin optimización. Esto reduce notablemente la calidad de la solución.

Al escalar el algoritmo del TSP a 10, 20, 30, 40 y 50 ciudades, el impacto sobre los tiempos de ejecución, la calidad de la solución y la viabilidad del enfoque cambia drásticamente, especialmente si no se usan heurísticas.

Usando la heurística de 'vecino cercano' se puede observar una mejora en los tiempos de ejecución. Se sigue sin encontrar la mejor ruta en con las heurísticas aplicadas. Sin embargo, en algunos casos la distancia total recorrida con el resultado de la heurística aplicada es más grande que en la corrida inicial. Por otra parte, los tiempos de ejecución con las primeras ciudades mejora bastante con la heurística aplicada.

1. **Analizar parámetro tee**

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

El parámetro tee ayuda a monitorear el progreso de optimización. También se puede ver el tiempo límite antes de encontrar una solución óptima.

interpretación:

* Se leyeron 4973 restricciones y 4970 variables
* Algunas advertencias por redefinición de límites de variables.
* Se encontró una solución óptima al LP relajado (964.009).
* El tiempo límite fue alcanzado antes de encontrar una solución óptima.

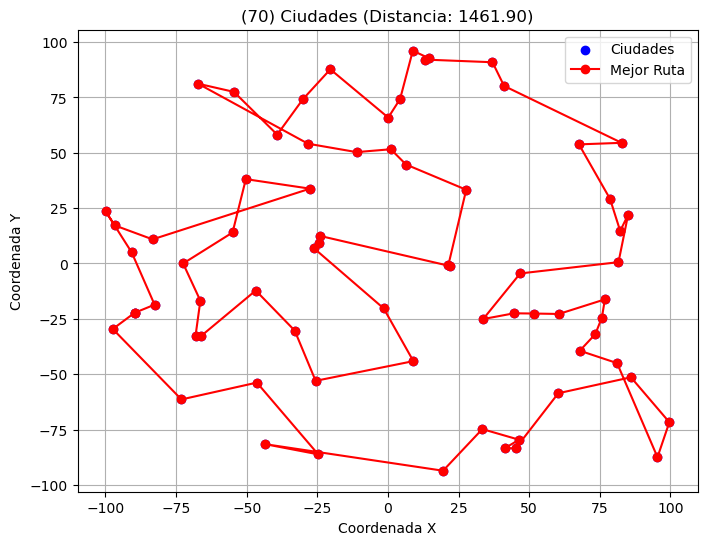
Gap al final:

* El mejor lower bound fue 995.39
* La mejor solución entera (mip) fue 1461.90
* El gap era de 31.9% al finalizar, por eso no es una solución óptima.
* Distancia total recorrida (solución actual): 1461.90
* Distancia mínima posible (teórica o relajada): 887.24
* Distancia máxima posible: 2129.37
* Heurística usada: 'limitar\_funcion\_objetivo': probablemente un recorte para guiar la búsqueda del solver más rápidamente hacia soluciones prometedoras.

Conclusión: El parámetro tee te permitió ver estos logs completos del solver.

El resultado final: No se encontró una solución óptima, pero sí una factible bastante buena. La búsqueda fue interrumpida por tiempo límite.

1. **Aplicar heurística de límites a la función objetivo**



**Con heurística:**

**A black text on a white background

AI-generated content may be incorrect.**

**Sin Heurística:**

A white background with black numbers

AI-generated content may be incorrect.

**¿Cuál es la diferencia entre los dos casos?**

La aplicación de la heurística permitió encontrar una mejor solución (más corta) en el mismo tiempo de ejecución.

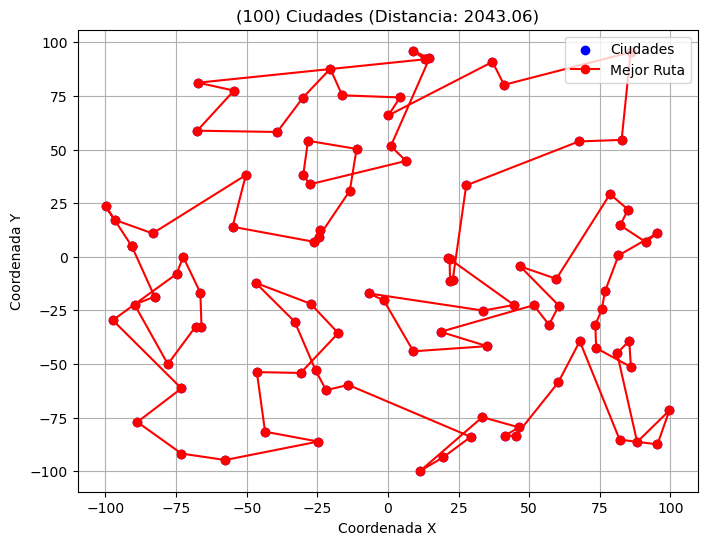
**¿Sirve esta heurística para cualquier caso? ¿Cuál pudiera ser una razón?**

La heurística limitar\_funcion\_objetivo puede ser útil en algunos casos, pero no garantiza mejoras en todos los contextos. Esto se debe a que:

1. Las heurísticas suelen estar diseñadas para reducir el espacio de búsqueda, enfocándose en soluciones más prometedoras, pero:
2. Pueden omitir soluciones óptimas si las restricciones impuestas son demasiado fuertes.

En este caso, funcionó bien porque probablemente acotó las soluciones malas, permitiendo al algoritmo concentrarse en rutas más eficientes.

1. **Aplicar heurística de vecinos cercanos**



**Con heurística:**

A black and white text

AI-generated content may be incorrect.

**Sin heurística:**

A white background with black text

AI-generated content may be incorrect.

**¿Cuál es la diferencia entre los dos casos?**

Si bien es cierto que tanto los tiempos de ejecución y los valores estadísticos son los mismos, la diferencia principal es que cuando se aplicó la heurística del vecino\_cercano se pudo llegar a una ruta con distancia total (no óptima).

**¿Sirve esta heurística para cualquier caso?**

La heurística vecina\_cercano no garantiza la mejor solución, pero sí ayuda a obtener una solución válida en muchos casos.

**Conclusiones:**

* El TSP es un problema NP-duro, lo que significa que el tiempo de resolución crece exponencialmente con el número de ciudades.
* Para pocas ciudades (10 o 20), se puede analizar la mayoría del espacio de soluciones sin heurísticas.
* A partir de 30 ciudades, la cantidad de rutas posibles se vuelve muy difícil de cuantificar, por lo que los métodos exactos sin heurísticas fallan.
* Las heurísticas permiten encontrar soluciones viables en tiempos razonables
* Usar heurísticas aumentan la capacidad para resolver instancias con muchas ciudades
* Aunque no óptimas, las soluciones suelen ser suficientemente buenas para la mayoría de las aplicaciones prácticas.
* Hay diferentes tipos de heurísticas por lo que la calidad depende del tipo de heurística usada