Traveling Salesman Problem with Time Windows

Algoritmos y Representación de Datos

Daniel F. Rincón[1]

[1] *Escuela Colombiana de ingenieria Julio Garavito, Bogotá, Colombia*

Fecha: 06/03/2021

Resumen— Solución aproximada al problema "Traveling Salesman Problem with Time Windows"por medio de un sencillo algoritmo voraz, este algoritmo se ejecuta en tiempos razonables para casos pequeños, en casos más grandes con un límite bajo de tiempo no encuentra soluciones posibles. La ventaja de este algoritmo es su fácil implementación y entendimiento.

Palabras clave— tsptw, algoritmo, voraz.

Abstract— Approximate solution to the "Traveling Salesman Problem with Time Windows"by means of a simple voracious algorithm, this algorithm runs in reasonable times for small cases, in larger cases with a low time limit it does not find possible solutions. The advantage of this algorithm is its easy implementation and understanding.

Keywords— tsptw, algorithm, greedy

Presentación del Problema

El problema del agente viajero con ventanas de tiempo es el problema de encontrar una ruta de coste mínimo que visite un conjunto de ciudades exactamente una vez, donde cada ciudad debe ser visitada dentro de una ventana de tiempo específica. Cada ciudad i tiene:

- Un tiempo de servicio β_i: es el tiempo necesario para visitar la ciudad.
- Un tiempo de preparación a_i (a veces llamado tiempo de liberación): no se puede visitar la ciudad antes de su tiempo de preparación.
- Un tiempo de vencimiento b_i (a veces llamado plazo): debe visitar la ciudad antes de su tiempo de vencimiento.

Sólo puede (y debe) visitar cada ciudad una vez. Los costes de los arcos representan los tiempos de viaje (y a veces también los tiempos de servicio). El coste total de un recorrido es la suma de los costes de los arcos utilizados en el recorrido. Los tiempos de preparación y de entrega de una ciudad i definen una ventana de tiempo $[a_i, b_i]$ dentro de la cual la ciudad debe ser visitada. Se puede visitar la ciudad antes de la hora lista, pero habrá que esperar hasta la hora lista para poder atenderla. Los tiempos de espera deben ser respetados y los recorridos que no visitan las ciudades antes de que su plazo finalice se consideran inviables.

Estrategia de Solución

Estrategias encontradas

Se investigaron diferentes técnicas definidas por diversos autores, la mayoría sumamente complejas, y no correspondían a la escencia simple del programa. Algunas de estas técnicas se listan continuación:

- Templado simulado comprimido (Blum, 2021)
- Búsqueda anidada de Monte-Carlo con adaptación de políticas (Edelkamp, 2021)
- Propagación de la programación de restricciones + Técnicas de investigación operativa (Lodi, 2021)

- Programación lineal entera ampliada en el tiempo (Boland, 2021)

Estrategia implementada

Se implementó una estrategia voraz sencilla, en donde se establecen aleatoriamente los nodos visitados bajo ciertas reglas, y se siguen intentando posibilidades hasta que el tiempo finalice. El algoritmo utilizado se puede describir con los siguientes pasos:

- Basados en el tiempo actual, las ventanas de tiempo de los nodos no visitados y los tiempos de viaje a cada uno de ellos, encontrar los nodos factibles para viajar (aquellos a los que se pueda llegar antes de que finalice su ventana de tiempo). Si no hay ningún nodo factible, volver al nodo anterior, calcular los nodos factibles sin tener cuenta el recién visitado, además agregar uno al contador de caminos fallidos.
- Elegir un nodo aleatorio del pool de nodos factibles, removerlo de los nodos por visitar y agregarlo a los nodos visitados.
- Repetir para el camino con el nuevo nodo visitado.
- Si el contador de caminos fallidos llega al máximo, comenzar de nuevo con la solución.
- Si la lista de nodos por visitar se encuentra vacía, se tiene una solución, retornarla.

Como podemos ver, la solución es muy sencilla, y aunque no es eficiente para casos grandes, si se trabaja con casos pequeños (<=20) se pueden obtener soluciones en tiempos sorpresivamente cortos.

RESULTADOS

El programa es funcional para casos pequeños con números de ciudades aproximadamente menores a 25, en estos casos encuentra soluciones razonables en tiempos menores a un minuto, en casos más grandes y complejos, no encuentra soluciones en tiempos razonables. Con mayores restricciones en el algoritmo quizá se podrían obtener mejores resultados, sin embargo, se decidió mantener simple, ya que esa es la escencia del algoritmo.

REFERENCIAS

[1] Blum, C. (2021). "The travelling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization". Tomado de https://www.sciencedirect.com/science/article/abs/pii/S1568494613001658.

[2] Boland, N. (2021). "Solving the traveling salesman problem with time windows through dynamically generated time-expanded networks". Tomado de https://link.springer.com/chapter/10.1007/978-3-319-59776-8_21.

[3] Edelkamp, S. (2021). "Algorithm and knowledge engineering for thetsptw problem". Tomado de https://www.researchgate.net/publication/261450637_Algorithm_and_knowledge_engineering_for_the_TSPTW_problem.

[4] Lodi, A. (2021). "A hybrid exact algorithm for the tsptw". Tomado de https://www.researchgate.net/publication/220669297_A_hybrid_exact_algorithm_for_the_TSPTW.