Conjuntos k-Dominantes Mínimos en grafos Ponderados

Anonymous Author(s)
Affiliation
Address
email

Abstract

En el artículo [1] recientemente publicado proponemos una generalización del conjunto k-dominante en un grafo, cuando consideramos un grafo ponderado. Dado un grafo con pesos en sus aristas, un conjunto de vértices es un conjunto $k-$dominante en un grafo ponderado si para cada vértice fuera del conjunto, la suma de los pesos desde él a los vértices en el conjunto es mayor o igual que k. El número de k-dominación en un grafo ponderado es la cardinalidad mínima entre todos los conjuntos k-dominantes en un grafo ponderado. Dado que el problema de encontrar el número de k-dominación en un grafo ponderado es $\mathcal{NP}$-difícil, se han comparado dieciséis variantes de un algoritmo exacto Iterated Greedy. Los resultados computacionales muestran que la propuesta es capaz de encontrar soluciones óptimas o casi óptimas en un tiempo computacional pequeño.

1 Introducción

El problema de encontrar un conjunto de dominación mínimo en un grafo tiene diversas aplicaciones en redes sociales , redes inalámbricas, y optimización logística . Este artículo presenta una generalización denominada problema del conjunto k-dominante mínimo en grafos ponderados (k-WDSP por sus siglas en inglés, k-Weighted Dominating Set Problem), donde se consideran pesos en las aristas y se requiere que la suma de los pesos de las aristas adyacentes a un vértice sea al menos *k*.

Concretamente en este trabajo publicado (ver [1]),

- Generalizamos el conjunto k-dominante cuando en lugar de grafos no ponderados consideramos grafos ponderados dando lugar a un nuevo problema, el conjunto k-dominante ponderado (k-WDS).
- Se han incluido diferentes procedimientos de destrucción y reconstrucción en el algoritmo considerado aprovechando el conocimiento del problema.
- Proponemos y comparamos las 16 variantes resultantes de un algoritmo Iterated Greedy para abordar el k-WDSP, que surge al combinar los procedimientos de destrucción y reconstrucción.
- Resolvemos el problema utilizando un algoritmo capaz de resolver grafos con diferentes tamaños y estructuras.
- Hemos comparado nuestro algoritmo con un procedimiento exacto para comprobar el potencial de nuestra propuesta y somos capaces de resolver instancias grandes en las que el algoritmo exacto se atasca.

Submitted to XVI Congreso EspaÃśol de Metaheuristicas, Algoritmos Evolutivos y Bioinsiprados (MAEB 2025). Do not distribute.

2 Definición del Problema

Dado un grafo ponderado $G = (V, E, w)$, donde $w : E \to \mathbb{R}^+$ asigna un peso positivo a cada arista, un conjunto $D \subseteq V$ es un conjunto k-dominante en un grafo ponderado si para cada vértice $v \notin D$, se cumple que $\sum_{u \in D} w(uv) \geq k$. El número de dominación k en un grafo ponderado, $\gamma_k(G)$, es el tamaño mínimo de tal conjunto.

Para modelar matemáticamente el k-WDSP, se incluye una formulación del problema de Programación Lineal Entera (ILP). Dado un grafo ponderado $G = (V, E, w)$, con vértices $V = \{v_1, v_2, \ldots, v_n\}$, denotamos por $W = (w_{ij})$ la matriz de pesos, una matriz cuadrada de $n \times n$ donde w_{ij} representa el peso de la arista $v_i v_j$, es decir, $w_{ij} = w(v_i v_j)$. Sin pérdida de generalidad, a lo largo de este manuscrito consideraremos que $w_{ij} \in [0, 1]$ para reescalar fácilmente las instancias que se resolverán. La formulación ILP utiliza variables binarias x_i que toman el valor 1 si el vértice v_i es seleccionado para ser parte del conjunto y 0 en caso contrario. El problema se puede formular de la siguiente manera:

$$\min \quad \sum_{i=1}^{n} x_i \tag{1}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} w_{ij} x_i \geq k(1 - x_j) \quad \forall j = 1, \ldots, n, \tag{2}$$

$$x_i \in \{0, 1\} \quad \forall i = 1, \ldots, n. \tag{3}$$

La función objetivo (1) representa el número total de vértices seleccionados, es decir, la cardinalidad de un k-WDS, mientras que las restricciones (2) aseguran que cada vértice no seleccionado reciba un peso total de al menos k unidades de los vértices seleccionados en el k-WDS. Finalmente, las restricciones (3) definen la naturaleza binaria de las variables de decisión.

Se ha demostrado que este problema es NP-difícil, lo que motiva y justifica el desarrollo de métodos heurísticos para su resolución.

3 Metodología

Se desarrollaron dieciséis variantes de un algoritmo iterativo codicioso (IG, por sus siglas en inglés, Iterated Greedy) para resolver el problema. Estas variantes incluyen técnicas de construcción y reconstrucción adaptadas específicamente para resolver el k-WDS.

3.1 Construcción de la Solución Inicial

El IG se inicia con un conjunto vacío y se agregan vértices iterativamente, eligiendo aquel que maximice la cantidad de nodos dominado. Se podría partir de una solución aleatoria, pero en este trabajo se propone una construcción codiciosa.

3.2 Fase de Destrucción y Reconstrucción

En la fase de destrucción se elimina un subconjunto de vértices de la solución actual, y en la fase de reconstrucción se vuelven a introducir vértices siguiendo diferentes estrategias o criterios. Concretamente seguimos cuatro estrategias diferentes para destruir y reconstruir una solución.

- Aleatoria
- Codiciosa basándonos en el grado del vértice
- Codiciosa basándonos en los pesos del vértice
- Codiciosa basándonos en cómo se completa la solución

Al combinar estas estrategias surgen 16 variantes del algoritmo IG.

4 Resultados Computacionales

Se compararon 16 variantes del algoritmo IG evaluando 156 instancias de grafos con tamaños de hasta 4000 vértices.

Hemos realizado varios experimentos. Como era de esperar, los resultados preliminares muestran que las variantes de IG que aprovechan más las características del problema dan mejores resultados. Se han probado soluciones iniciales aleatorias y basadas en el problema y los resultados evidencian que empezar con una solución aleatoria lleva en torno a un 32% menos de tiempo, pero consigue soluciones de peor calidad (22% de desviación). Se han realizado análisis paramétricos para ajustar los principales parámetros de entrada. Además, se han probado diferentes patrones de peso de aristas para comprobar su impacto en la función objetivo resultante, y observamos que la dificultad del problema aumenta cuando aumentan las diferencias de peso de las aristas.

Hemos realizado dos experimentos finales con todas las instancias, el primero comparando el mejor IG con el algoritmo exacto para comprobar los límites del procedimiento exacto y el rendimiento del IG. Para las instancias de tamaño pequeño, hasta casi 200 vértices, el algoritmo exacto encuentra soluciones casi óptimas, con una brecha de optimalidad media del 0,41%, pero consume mucho tiempo, alcanzando el límite de 30 minutos para instancias con más de 400 vértices. Dado que, hasta donde sabemos, no existen otros algoritmos del estado del arte para resolver el k-WDSP, el segundo experimento final compara los resultados de nuestra metodología con la aplicada para resolver un caso particular del problema propuesto, el problema del conjunto mínimo dominante (ver [2]). Aunque los tiempos de computación no son comparables sin

5 Conclusiones

El IG demostró ser una herramienta eficaz para resolver el k-WDSP, reduciendo significativamente el tiempo de cómputo sin sacrificar la calidad de la solución. Futuras investigaciones pueden enfocarse en variantes metaheurísticas híbridas para mejorar aún más su desempeño.

References

[1] Barrena, E, Bermudo, S., Hernández-Díaz, A.G., López-Sánchez, A.D., Zamudio, J.A. (2025). Finding the minimum k-weighted dominating sets using heuristic algorithms. *Mathematics and Computers in Simulation* **228**:485-497.

[2] Casado, A, Bermudo, S., López-Sánchez, A.D., Sánchez-Oro, J. (2023). An iterated greedy algoritm for finding the minimum dominating set in gragh. *Mathematics and Computers in Simulation* **207**:41-58.