El Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas

Lidia Cruz-Pérez[1], Sergio Pérez-Peló[1], Ana D. López-Sánchez[2], and Eva Barrena[2]

[1] Universidad Rey Juan Carlos, Dept. Informática y Estadística, C/ Tulipán S/N, Móstoles, 28933, Madrid, España
[2] Universidad Pablo de Olavide, Dept. Economía, Métodos Cuantitativos e Historia Económica, Ctra. Utrera km 1, Sevilla, 41013, Sevilla, España

Abstract. En este trabajo se propone una nueva variante del problema de dominación en grafos, denominada el Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas (VCEW-DSP, del inglés *Vertex-costs and Edge-weights Dominating Set Problem*). A diferencia de la versión clásica, el VCEW-DSP considera costes en los vértices y pesos en las aristas del grafo. El objetivo del problema es seleccionar un subconjunto de vértices de coste mínimo de forma que cada vértice no perteneciente al conjunto reciba de este un peso entrante al menos igual a un valor k. Este peso entrante se define como la suma de los pesos de las aristas que lo conectan con el conjunto. En este trabajo se define formalmente el VCEW-DSP y se presenta un modelo de Programación Lineal Entera para su resolución exacta. De forma complementaria, se explora la metaheurística Búsqueda Local Iterativa, con el fin de estudiar alternativas escalables para instancias de mayor tamaño.

Keywords: Optimización combinatoria · Grafos ponderados · Dominación en grafos · Búsqueda Local Iterativa

1 Introducción

Los problemas de dominación en grafos conforman un conjunto de problemas ampliamente estudiados en optimización combinatoria debido a sus numerosas aplicaciones en redes sociales ([11]), redes de comunicación ([8]) y localización de instalaciones ([5]), entre otros. En su formulación básica, un conjunto dominante es un subconjunto de vértices tal que todo vértice no perteneciente a dicho conjunto es adyacente a al menos uno de ellos. El problema asociado, conocido como el Problema del Conjunto Dominante Mínimo (MDSP, del inglés *Minimum Dominating Set Problem*), consiste en determinar un conjunto dominante de tamaño mínimo. Desde un punto de vista práctico, los conjuntos dominantes representan grupos de elementos clave capaces de controlar, influir o proporcionar servicio al resto de elementos del grafo, por lo que la reducción de sus tamaños resulta relevante para la optimización de recursos.

El concepto de dominación en grafos tiene su origen en el trabajo de Ore ([13]) y fue posteriormente consolidado y ampliado en los estudios de Haynes et

al ([9]). A partir de entonces, se ha desarrollado una extensa familia de variantes (ver [10]), motivadas por la necesidad de representar de forma más precisa distintos escenarios aplicados. En particular, una línea de investigación relevante es la dominación en grafos ponderados, en los que se asignan pesos a vértices, aristas o ambos. Los pesos en vértices suelen interpretarse como costes de selección o consumo de recursos, mientras que los pesos en aristas permiten modelar la intensidad, fiabilidad o calidad de las conexiones. Un ejemplo es el Problema del Conjunto Dominante de Peso Mínimo (MWDSP, del inglés *Minimum Weight Dominating Set Problem*), en el que los vértices tienen asociado un peso y se busca minimizar el peso total del conjunto dominante en lugar de su cardinalidad ([1]). En modelos con pesos en aristas destaca el Problema del Conjunto Dominante k-ponderado Mínimo (k-WDSP, del inglés *k-Weighted Dominating Set Problem*), en el que se exige que cada vértice no seleccionado reciba un peso mínimo k a través de las aristas que lo conectan al conjunto dominante ([3]).

El problema propuesto, denominado el Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas (VCEW-DSP, del inglés *Vertex-costs and Edge-weights Dominating Set Problem*), combina la optimización de costes en vértices, propia del MWDSP, con restricciones de dominación basadas en pesos de aristas, como en el k-WDSP. De este modo, la consideración conjunta de ambos tipos de pesos permite capturar de forma más flexible situaciones en las que se desean conjuntos de coste reducido, pero sin perder un nivel mínimo de interacción en la grafo. Por ejemplo, en redes inalámbricas de sensores y comunicaciones, los costes de los vértices pueden representar el despliegue, la activación o el mantenimiento de los dispositivos, mientras que los pesos de las aristas reflejan la calidad de la comunicación. En este contexto, el umbral k asegura un nivel mínimo de conectividad entre cada dispositivo y los sensores o estaciones de retransmisión elegidos. De forma análoga, en problemas de localización de instalaciones y diseño de redes de servicio, los costes de los vértices representan la apertura de instalaciones, mientras que los pesos de las aristas modelan la accesibilidad, la capacidad de transporte o la calidad del servicio.

Desde el punto de vista computacional, el VCEW-DSP es un problema $\mathcal{NP}$-duro, al generalizar variantes clásicas del problema de dominación. En consecuencia, los métodos exactos no resultan escalables para instancias de gran tamaño, lo que motiva el uso de enfoques metaheurísticos. En este trabajo se propone un algoritmo basado en la metaheurística Búsqueda Local Iterativa (ILS, del inglés *Iterated Local Search*) para obtener soluciones óptimas o sub-óptimas en tiempos razonables.

2 Definición del problema

Sea $G = (V, E, c, w)$ un grafo simple, no dirigido y conexo con costes en los vértices y pesos en las aristas, donde $c : V \to \mathbb{R}_{\geq 0}$ asigna un coste no negativo a cada vértice $i \in V$ y $w : E \to \mathbb{R}_{>0}$ asigna un peso positivo a cada arista $ij \in E$.

Definition 1 (k-conjunto dominante ponderado). *Un conjunto $S \subseteq V$ es un k-conjunto dominante ponderado (k-WDS, del inglés k-weighted dominating*

set) si cada vértice $i \in V \setminus S$ recibe de S un peso entrante mayor o igual a k, es decir,

$$\sum_{j \in S} w(ij) \geq k, \quad \forall i \in V.$$

Definition 2 (Coste de un conjunto de vértices). *El coste de un conjunto $S \subseteq V$ se define como la suma de los costes de los vértices del conjunto, es decir,*

$$C(S) = \sum_{i \in S} c(i).$$

Definition 3 (Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas). *Dado un grafo $G = (V, E, c, w)$ y un número $k \in \mathbb{R}_{>0}$, el Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas (VCEW-DSP) consiste en encontrar un k-WDS, $S \subseteq V$, que minimice $C(S)$, es decir,*

$$\min_{S \subseteq V} C(S) \quad \textit{sujeto a} \quad \sum_{j \in S} w(ij) \geq k, \quad \forall i \in V \setminus S.$$

El coste óptimo del problema se denota por $\gamma^k_{VE}(G)$ y se define como

$$\gamma^k_{VE}(G) = \min \left\{ C(S) : S \subseteq V, \ \sum_{j \in S} w(ij) \geq k, \ \forall i \in V \setminus S \right\}.$$

La Figura 1 muestra un ejemplo de solución del VCEW-DSP con $k = 1$. Las etiquetas de los vértices representan los costes, c, mientras que los pesos de las aristas, w, se indican sobre estas. Los vértices resaltados forman un 1-WDS, $S = \{A, E, F, H, I\}$, ya que para cada vértice no perteneciente a S, la suma de los pesos de las aristas que lo conectan a S es al menos 1. En particular, este peso entrante es igual a 1.00 para B, 1.30 para C, 1.10 para D, 1.90 para G y 1.10 para J. El coste de la solución, $C(S)$, es 25 y, además, dicha solución es óptima ($\gamma^1_{VE}(G) = 25$).

El VCEW-DSP generaliza varios problemas de dominación en grafos. En particular, si todas las aristas tienen peso al menos k, el problema se reduce al MWDSP; si todos los vértices tienen el mismo coste, se obtiene el k-WDSP, y, si se cumplen ambas condiciones simultáneamente, se tiene el MDSP.

2.1 Formulación matemática

Sea $G = (V, E, c, w)$ un grafo con costes en los vértices y pesos en las aristas, donde $V = \{1, 2, \ldots, n\}$. Los pesos de las aristas se representan mediante una matriz simétrica $W = (w_{ij}) \in \mathbb{R}^{n \times n}_{\geq 0}$, donde w_{ij} denota el peso de la arista $ij \in E$, con $w_{ij} = 0$ si $ij \notin E$. Además, a cada vértice $i \in V$ se le asocia un coste $c_i \in \mathbb{R}_{\geq 0}$.

El modelo de Programación Lineal Entera (ILP, del inglés *Integer Linear Programming*) propuesto define, para cada vértice $i \in V$, una variable binaria

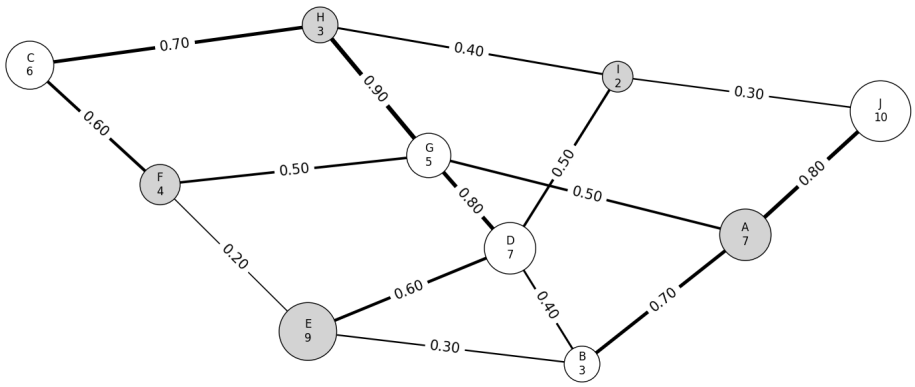


Fig. 1. Ejemplo de una solución óptima para el VCEW-DSP con $k = 1$ ($\gamma^1_{VE}(G) = 25$).

x_i, que toma valor 1 si el vértice i se selecciona como parte de la solución y 0 en caso contrario. La formulación es la siguiente:

$$(F1)\ \min \sum_{i \in V} c_i x_i \tag{1}$$

$$\text{s.a.} \sum_{i \in V} w_{ij} x_i \geq k(1 - x_j) \quad \forall j \in V \tag{2}$$

$$x_i \in \{0, 1\} \quad \forall i \in V. \tag{3}$$

La función objetivo (1) minimiza la suma de los costes de los vértices seleccionados. El conjunto de restricciones (2) garantizan que cada vértice no seleccionado reciba un peso entrante de al menos k desde los vértices seleccionados en la solución. Por último, las restricciones (3) definen la naturaleza binaria de las variables de decisión.

El VCEW-DSP, al ser una generalización del MDSP, el cual es $\mathcal{NP}$-duro [7], también es $\mathcal{NP}$-duro. Para abordar instancias de gran tamaño, se propone un enfoque metaheurístico adaptado al problema.

3 Propuesta algorítmica

La Búsqueda Local Iterativa (ILS, del inglés *Iterated Local Search*) es una metaheurística trayectorial, propuesta en [12], que emplea la búsqueda local como mecanismo principal. La búsqueda local explora el entorno de una solución mediante el desplazamiento secuencial hacia soluciones vecinas con mejor valor de la función objetivo. Este proceso continúa hasta alcanzar un óptimo local. Aunque es una técnica eficaz para mejorar soluciones, presenta la limitación de quedar atrapada en óptimos locales. ILS aborda este problema mediante la aplicación de una perturbación sobre la solución atrapada, lo que permite escapar de óptimos

locales y explorar nuevas regiones del espacio de soluciones desde las que reiniciar el proceso de búsqueda local. De este modo, ILS alterna entre intensificación, mediante búsqueda local, y diversificación, mediante perturbaciones.

El Algoritmo 1 presenta el esquema general de la metaheurística ILS.

Algorithm 1 ILS

```
1: S_0 ← ConstrucciónSoluciónInicial()
2: S ← BúsquedaLocal(S_0)
3: while no se ha alcanzado el criterio de parada do
4:     S' ← Perturbación(S)
5:     S' ← BúsquedaLocal(S')
6:     if f(S') ≤ f(S) then
7:         S ← S'
8:     end if
9: end while
10: Devolver S
```

3.1 Construcción de una solución inicial

Para la construcción de la solución inicial se emplea la fase constructiva de la metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) [6]. Este procedimiento construye una solución, S, de forma incremental, añadiendo vértices iterativamente. En cada paso, se genera una lista restringida de candidatos (RCL) formada por los vértices no pertenecientes a la solución y que presentan una mejor puntuación según una función de evaluación, g,

$$RCL = \{i \in V \setminus S : g(i) \geq g_{max} - \alpha(g_{max} - g_{min})\},$$

donde $\alpha \in [0, 1]$ controla el grado de aleatoriedad de la construcción. Posteriormente, se selecciona aleatoriamente un vértice de la RCL y se añade a la solución parcial, S. El proceso finaliza cuando se obtiene una solución factible, es decir, cuando S es un k-WDS. Por último, se eliminan los vértices redundantes de la solución, entendiendo por tales aquellos cuya eliminación no compromete la factibilidad.

La función de evaluación, g, utilizada para valorar cada vértice $i \in V \setminus S$ se define como:

$$g(i) = \frac{\left(\sum_{j \in N(i)} w(ij)\right) / \deg(i)}{c(i)},$$

donde $N(i)$ representa el conjunto de vértices adyacentes a i y $\deg(i)$ su grado. Esta función prioriza vértices con un peso medio elevado de sus aristas y penaliza simultáneamente aquellos con mayor coste.

3.2 Búsqueda local

Para el VCEW-DSP se propone una búsqueda local basada en el movimiento *swap1x*, que consiste en eliminar un vértice, i, de la solución actual, S y, en caso

necesario, añadir un subconjunto de vértices, $X \subseteq V \setminus S \cup \{i\}$, para restablecer la factibilidad de $S \setminus \{i\}$. La vecindad inducida por este movimiento se define como

$$\mathcal{N}_{swap1x}(S) = \{(S \setminus \{i\}) \cup X \ : \ i \in S,\ X \subseteq V \setminus S \cup \{i\} \text{ restaura factibilidad}\}.$$

Nótese que el conjunto X puede ser vacío cuando el vértice eliminado, i, resulta redundante.

La búsqueda local utiliza una estrategia de *primera mejora*, en la que se acepta la primera solución vecina que mejora el valor de la función objetivo. Esta elección se justifica por el elevado tamaño de la vecindad inducida por el movimiento *swap1x*, cuyo tamaño está acotado superiormente por $|S| \cdot 2^{|V|-|S|}$.

3.3 Perturbación

La idea principal de la perturbación es modificar la solución lo suficiente como para escapar del óptimo local en el que se encuentra, preservando al mismo tiempo parte de su estructura.

Para el VCEW-DSP se proponen dos mecanismos de perturbación. El primero consiste en una destrucción parcial y aleatoria de la solución, seguida de la recuperación de la factibilidad mediante la incorporación, también aletoria, de vértices. El segundo mecanismo actúa en sentido contrario, añadiendo de forma aleatoria vértices redundantes a la solución. En ambos casos, la intensidad de la perturbación se controla mediante un parámetro $\rho \in (0, 1)$, que determina la proporción de vértices a añadir o eliminar en relación con el tamaño de la solución a perturbar.

4 Resultados preliminares

Los experimentos se han ejecutado en un AMD Ryzen 9 5950X con 32 hilos y 128 GB de memoria RAM. La metaheurística ILS se ha implementado en Java 23.0.1, mientras que el modelo ILP se ha resuelto mediante Gurobi 12.0.3, estableciendo un tiempo máximo de ejecución de 1800 segundos.

El conjunto de instancias está compuesto por 160 grafos con tamaños entre 100 y 5000 nodos y densidades comprendidas entre 0.2 y 0.5. Las instancias se han generado a partir de grafos de la literatura [2, 4], a los que se les han asignado pesos aleatorios en las aristas dentro del intervalo $(0, 1]$ y costes enteros en los vértices entre 1 y 10. En los experimentos preliminares se ha fijado el umbral de dominación en $k = 1$. Para la calibración de las distintas componentes del ILS (Tablas 1–3), se ha utilizado un subconjunto preliminar de 40 instancias, correspondiente al 25% del conjunto total.

La Tabla 1 compara las distintas funciones de evaluación empleadas en la fase de construcción de la solución inicial (columna 1). Para ello, se muestran la calidad de la solución, es decir, la función objetivo (F.O.), y el tiempo computacional, medido en segundos, en las columnas 2 y 3, respectivamente. Los

resultados muestran que las funciones basadas exclusivamente en la estructura del grafo (g_1 y g_2) generan soluciones de elevado coste, al no considerar el valor asociado a los vértices. Por el contrario, la función centrada únicamente en el coste (g_3) mejora notablemente la calidad de las soluciones, al estar directamente alineada con la función objetivo del problema. Sin embargo, los mejores resultados se obtienen con g_4, que combina información estructural del grafo con el coste de los vértices. En cuanto los tiempos de ejecución, todas las funciones presentan tiempos muy reducidos y prácticamente idénticos, del orden de centésimas de segundo.

Table 1. Comparación de distintas funciones de evaluación para la construcción de la solución inicial.

Función de evaluación	F.O.	Tiempo
Grado (g_1)	112.35	**0.06**
Suma de los pesos de las aristas (g_2)	112.60	0.07
Coste (g_3)	64.08	**0.06**
Ratio entre el peso medio de las aristas y el coste (g_4)	**34.35**	**0.06**

Respecto a la búsqueda local, la Tabla 2 presenta la comparación de cuatro variantes definidas a partir de la combinación de distintas estrategias de selección de vértices a eliminar e insertar para el movimiento *swap1x*. En particular, se consideran dos criterios de selección: aleatorio y heurístico. En el caso del heurístico, durante la eliminación se priorizan los vértices de mayor coste, mientras que en la inserción se favorecen los de menor valor.

La tabla muestra que la estrategia de inserción tiene un mayor impacto en la calidad de las soluciones que la de eliminación, observándose mejores resultados cuando se emplea inserción heurística. En particular, la combinación de eliminación aleatoria e inserción heurística obtiene el mejor rendimiento global. Este comportamiento sugiere que la aleatoriedad en la eliminación favorece la diversificación, mientras que la inserción guiada contribuye a mejorar la calidad de las soluciones. En cuanto a los tiempos de cómputo, las diferencias son reducidas. La configuración completamente heurística es la más rápida, aunque no proporciona los mejores resultados, lo que sugiere una posible convergencia prematura hacia óptimos locales. Por su parte, la estrategia completamente aleatoria presenta el peor comportamiento tanto en tiempo como en calidad de solución.

Por último, la Tabla 3 compara los dos mecanismos de perturbación diseñados para distintos valores de intensidad ρ. Los resultados indican que la eliminación de vértices no constituye una estrategia efectiva, ya que su rendimiento se deteriora a medida que aumenta la intensidad. Esta ineficacia se debe a la pérdida de estructura de la solución y al posterior proceso aleatorio de restauración de la factibilidad. En cambio, la inserción de vértices muestra un comportamiento más robusto, alcanzando su mejor rendimiento para $\rho = 0.20$, el segundo valor más bajo considerado. En términos generales, salvo en el nivel de intensidad más

Table 2. Comparación de distintas estrategias de eliminación e inserción en la búsqueda local *swap1x*.

F.O. (Tiempo)		
Eliminación \ Inserción	Aleatoria	Heurística
Aleatoria	28.23 (0.67)	**21.40** (0.51)
Heurística	22.48 (0.55)	21.50 **(0.35)**

bajo, la inserción supera de forma consistente a la eliminación tanto en calidad de solución como en tiempo de ejecución, siendo aproximadamente la mitad de costosa. Esto se debe a que evita el proceso de restauración de la factibilidad asociado a la eliminación. Además, se observa que valores menores de ρ conllevan tiempos de ejecución más reducidos, al implicar modificaciones más suaves sobre la solución.

Table 3. Comparación de distintos mecanismos de perturbación para diferentes niveles de intensidad, ρ.

	Inserción		Eliminación	
ρ	F.O.	Tiempo	F.O.	Tiempo
0.1	21.28	**0.15**	21.03	0.32
0.2	**20.98**	0.23	21.28	0.42
0.3	21.00	0.24	21.30	0.57
0.4	21.10	0.27	21.33	0.73

Una vez determinada la mejor configuración del ILS (función de evaluación g_4 para la construcción inicial, búsqueda local con eliminación aleatoria e inserción heurística y perturbación por inserción con $\rho = 0.20$), se ha realizado una comparativa frente al modelo ILP. La Tabla 4 recoge las medias de la función objetivo y el tiempo de cómputo así como el número de veces que cada método alcanza la mejor solución encontrada, agrupadas por tamaño de instancia (16 instancias por grupo).

Los resultados muestran que el modelo ILP alcanza soluciones óptimas en las instancias de hasta 400 vértices, ya que el gap obtenido por el *solver* es nulo. A partir de ese tamaño, el gap aumenta progresivamente, lo que indica que Gurobi alcanza el límite de tiempo de 1800 segundos. Para estas instancias en las que el modelo exacto obtiene soluciones óptimas, el ILS también consigue encontrarlas en prácticamente todos los casos (salvo dos excepciones), con tiempos comparables en las instancias de 100 y 200 vértices y significativamente inferiores en las de 400 y 500 vértices. En particular, en las instancias de 500 vértices se obtienen las 16 soluciones óptimas, con una reducción del tiempo de cómputo, concretamente del 97.45%.

Para las instancias en las que el modelo ILP no alcanza la optimalidad dentro del tiempo permitido, el rendimiento del ILS también se ve afectado, aunque mantiene tiempos significativamente inferiores y más estables que el enfoque exacto.

Table 4. Comparación entre modelo ILP e ILS en instancias de distintos tamaños.

	ILP				ILS			
$\|V\|$	F.O	Tiempo	Gap	#Mejores	F.O	Tiempo	#Mejores	Desv. (%)
100	**20.31**	**0.17**	0.00	16	**20.31**	0.28	16	0.00
200	**17.94**	0.77	0.00	16	18.00	**0.71**	15	0.20
300	**17.63**	5.72	0.00	16	17.69	**1.33**	15	0.69
400	**16.69**	87.05	0.00	16	**16.69**	**2.23**	16	0.00
500	**17.06**	754.32	0.03	16	17.38	**3.31**	11	2.72
1000	**18.13**	1800.09	0.30	14	18.38	**9.50**	10	2.40
2000	**20.38**	1800.24	0.54	16	20.75	**21.86**	10	1.78
3000	**22.13**	1800.57	0.63	14	22.19	**34.05**	13	0.21
4000	**23.00**	1801.33	0.66	15	23.13	**51.29**	13	0.61
5000	**23.94**	1802.42	0.67	15	24.13	**63.90**	12	1.23
Media	19.72	985.27	0.28	-	19.87	18.85	-	0.98

5 Conclusiones y trabajo futuro

En este trabajo se ha introducido el Problema del Conjunto Dominante con Costes en Vértices y Pesos en Aristas (VCEW-DSP), una extensión del problema clásico que permite modelar escenarios más realistas al considerar simultáneamente costes asociados a los vértices y calidad en las conexiones, representada mediante pesos en las aristas. Se ha presentado su definición formal junto con una primera aproximación a su análisis estructural y modelización. Como método de resolución, se ha propuesto una metaheurística basada en la Búsqueda Local Iterativa (ILS), diseñada específicamente para explotar las características del problema. Los resultados preliminares, obtenidos sobre instancias de tamaño pequeño y medio, muestran que el enfoque ILS es capaz de encontrar soluciones de buena calidad en tiempos de cómputo reducidos. Como trabajo futuro, se plantea ampliar el estudio experimental a instancias de mayor tamaño, así como explorar mejoras en el diseño del ILS y su posible integración con otras técnicas para potenciar su rendimiento y robustez.

Acknowledgments.

Disclosure of Interests. Los autores declaran que no tienen intereses en conflicto que sean relevantes para el contenido de este trabajo.

Bibliography

[1] Albuquerque, M., Vidal, T.: An efficient matheuristic for the minimum-weight dominating set problem. Applied Soft Computing **72**, 527–538 (2018). https://doi.org/10.1016/j.asoc.2018.06.052

[2] Álvarez-Miranda, E., Sinnl, M.: Exact and heuristic algorithms for the weighted total domination problem. Computers & Operations Research **127**, 105157 (2021). https://doi.org/10.1016/j.cor.2020.105157

[3] Barrena, E., Bermudo, S., Hernández-Díaz, A., López-Sánchez, A., Zamudio, J.: Finding the minimum k-weighted dominating sets using heuristic algorithms. Mathematics and Computers in Simulation **228**, 485 – 497 (2025). https://doi.org/10.1016/j.matcom.2024.09.010

[4] Casado, A., Bermudo, S., López-Sánchez, A., Sánchez-Oro, J.: An iterated greedy algorithm for finding the minimum dominating set in graphs. Mathematics and Computers in Simulation **207**, 41–58 (2023). https://doi.org/10.1016/j.matcom.2022.12.018

[5] Corcoran, P., Gagarin, A.: Heuristics for k-domination models of facility location problems in street networks. Computers & Operations Research **133**, 105368 (2021). https://doi.org/10.1016/j.cor.2021.105368

[6] Feo, T.A., Resende, M.G.C., Smith, S.H.: A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set. Operations Research **42**(5), 860 – 878 (1994). https://doi.org/10.1287/opre.42.5.860

[7] Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. A Series of Books in the Mathematical Sciences, W. H. Freeman and Co., San Francisco, Calif. (1979)

[8] Hassani Karbasi, A., Ebrahimi Atani, R.: Application of dominating sets in wireless sensor networks. International Journal of Security and its Applications **7**, 185 – 202 (2013)

[9] Haynes, T.W., Hedetniemi, S., Slater, P.: Fundamentals of Domination in Graphs. CRC Press, Boca Raton, 1 edn. (1998). https://doi.org/10.1201/9781482246582

[10] Haynes, T.W., Hedetniemi, S.T., Henning, M.A.: Models of Domination in Graphs. In: Topics in Domination in Graphs, vol. 64, pp. 13 – 30. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51117-3_2

[11] Kelleher, L.L., Cozzens, M.B.: Dominating sets in social network graphs. Mathematical Social Sciences **16**, 267 – 279 (1988). https://doi.org/10.1016/0165-4896(88)90041-8

[12] Lourenço, H., Martin, O., Stützle, T.: Iterated Local Search, pp. 321 – 353. Kluwer Academic Publishers (01 2003). https://doi.org/10.1007/0-306-48056-5_11

[13] Ore, O.: Theory of Graphs, vol. 38. American Mathematical Society Colloquium Publications, Povidence, Rhode Island (1962)