Un marco general para el diseño de heurísticas constructivas para problemas de embebidos de grafos

Sergio Cavero
Dpto. de Informática y Estadística
Universidad Rey Juan Carlos
Móstoles, España
sergio.cavero@urjc.es

Eduardo G. Pardo
Dpto. de Informática y Estadística
Universidad Rey Juan Carlos
Móstoles, España
eduardo.pardo@urjc.es

Mauricio G. C. Resende
University of Washington
Seattle, WA, USA
mgcr@berkeley.edu

***Resumen*—Los problemas de embebido de grafos (GLP, por sus siglas en inglés, *Graph Layout Problems*) son una familia de problemas de optimización combinatoria que buscan asignar los vértices de un grafo de entrada a los vértices de un grafo huésped, satisfaciendo restricciones específicas y optimizando una función matemática. Debido a su complejidad computacional, se suelen utilizar algoritmos aproximados, como las heurísticas. Este artículo presenta una revisión de las heurísticas constructivas que se utilizan para generar soluciones de partida para algunos de los GLP más estudiados de la literatura. A partir de la revisión realizada, se propone un marco general para la propuesta de heurísticas constructivas para esta familia de problemas u otros problemas relacionados, así como un conjunto de posibles trabajos futuros.**

***Palabras clave*—*graph layout problems*, embebido de grafos, heurísticas, algoritmos constructivos, optimización combinatoria**

I. Introducción

Los problemas de embebidos de grafos, más conocidos en inglés como *Graph Layout Problems* (GLP) son una familia de problema de optimización combinatoria que tiene como objetivo encontrar una disposición o diseño específico para un grafo de entrada. Estos problemas tienen una amplia gama de aplicaciones en diversos campos, como el dibujado de grafos, la migración de redes de telecomunicaciones, el diseño de circuitos VLSI, la programación de tareas, el diseño biológico y la distribución de instalaciones, entre otras [10], [11].

La resolución de un GLP implica asignar los vértices de un grafo de entrada a los de un grafo huésped. Aunque cualquier estructura puede ser un grafo huésped. Entre ellos, los de topología conocida (como son los caminos, ciclos, árboles, cuadrículas/rejillas) son los más estudiados [11], [36]. Esta asignación, conocida como etiquetado, embebido o numeración, define una solución de un GLP. El objetivo es de los GLP es encontrar una solución que optimice una función matemática, como minimizar los cruces de las aristas del grafo de entrada o las distancias, medidas en el grafo huésped, entre dos vértices adyacentes del grafo de entrada, entre otras.

Los GLP son, en su mayoría, NP-difíciles [14], [15], [35] por lo que encontrar la solución óptima es una tarea realmente compleja y, generalmente, para la mayoría de las aplicaciones prácticas es suficiente encontrar soluciones factibles con una calidad subóptima. Por lo tanto, se suelen emplear técnicas aproximadas para abordarlos, entres las que se encuentran las heurísticas. Estas heurísticas pueden ser de dos tipos: constructivas y de mejora [50], que habitualmente se combinan en dos fases. Durante la fase de construcción, las soluciones se construyen paso a paso utilizando una estrategia específica del problema, que equilibra la eficiencia computacional y la calidad de la solución [48]. Las heurísticas constructivas son particularmente útiles para encontrar rápidamente soluciones factibles para grafos de grandes proporciones. En la fase de mejora, el método más común es la búsqueda local. Este procedimiento consiste en explorar un subconjunto de soluciones del espacio de búsqueda con la finalidad de encontrar mejores soluciones [31].

Este trabajo está motivado por el creciente interés de la comunidad científica en el estudio de los GLP y problemas similares, cuyos algoritmos y estrategias son fácilmente extrapolables. Por ejemplo, resultan de especial interés los problemas de dominancia [1], [4] o los problemas de localización de instalaciones [12], [29]. En concreto, en la literatura, se puede encontrar una amplia variedad de artículos que estudian los GLP desde diferentes perspectivas. Los primeros trabajos relacionados con los GLP se remontan a finales del siglo XX. En el momento en que se escribió este artículo, se identificaron casi 50.000 publicaciones que estudian los GLP, sus aplicaciones o problemas relacionados[1].

Este trabajo ha sido parcialmente financiado por los proyectos PID2021-125709OA-C22 y PID2021-126605NB-I00, financiados por MCI-N/AEI/10.13039/501100011033 y "ERDF A way of making Europe"; el proyecto CIAICO/2021/224, financiado por la Generalitat Valenciana; el proyecto M2988, financiado por la convocatoria "Proyectos Impulso de la Universidad Rey Juan Carlos 2022"; la "Cátedra de Innovación y Digitalización Empresarial entre Universidad Rey Juan Carlos y Second Episode" (Ref. MCA06); y la "Red Española de optimización heurística 4.0 digitalización" (Ref. RED2022-134480-T).

[1]https://www.webofscience.com/wos/woscc/summary/ebb5950b-66dd-4aac-a45f-97854680b05b-62b12cce/relevance/1

Entre ellas, en este trabajo, se ha centrado la atención en el estudio de aquellas investigaciones en las que se proponen heurísticas constructivas para problemas de embebidos de grafos. Considerando la naturaleza extensa e intrincada de la familia de GLP, esta investigación se focaliza en la revisión de propuestas que abordan GLP donde el embebido se realiza en un grafo con estructura conocida (grafos camino, ciclos, árboles, rejillas, etc.) ya que son aquellos problemas que han ganado una atención significativa en la última década [7], [36]. En concreto, se revisan diversas estrategias heurísticas, evaluando sus méritos, limitaciones e idoneidad para diferentes tipos de grafos.

El resto del artículo contiene una breve formalización de los GLP (Sección II), una revisión general de las heurísticas constructivas consideradas en esta investigación (Sección III) y un novedoso marco para proponer nuevas heurísticas constructivas basado en la revisión bibliográfica realizada (Sección IV), finalizando con las conclusiones y posibles líneas de investigación futura para potenciar los hallazgos de esta investigación (Sección V).

II. Definición y formalización de los GLP

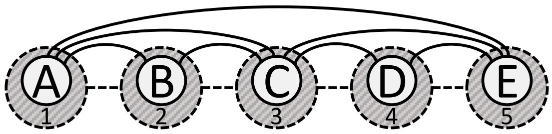
En primer lugar, se definen los GLP y algunos conceptos asociados. Siguiendo las definiciones y notaciones estándar de la familia GLP, sea $G(V_G, E_G)$ un grafo tal que V_G es el conjunto de vértices y E_G el conjunto de aristas. De manera similar, sea $H(V_H, E_H)$ un grafo huésped donde V_H representa el conjunto de vértices y E_H representa el conjunto de aristas. Un GLP consiste en encontrar un embebido (también conocido como disposición o incrustación) de un grafo dado G en un grafo huésped H. Un embebido se define formalmente como la función matemática $\varphi : V_G \rightarrow V_H$ tal que si $(u, v) \in E_G$, entonces $p(\varphi(u), \varphi(v))$ es el camino más corto compuesto por aristas de E_H que conecta u y v.

A modo de ejemplo, sea G un grafo de entrada con 5 vértices ($V_G = \{\text{A, B, C, D, E}\}$) y 7 aristas ($E_G = \{(\text{A,B}), (\text{A,C}), (\text{A,E}), (\text{B,C}), (\text{C,D}), (\text{C,E}), (\text{D,E})\}$). A continuación, se ilustra un posible embebido de G en un grafo huésped camino, H_P (Figura 1a) y un grafo huésped ciclo H_C (Figura 1b). Independientemente del grafo huésped, la función de asignación φ se define de la siguiente manera: $\varphi(A) = 1$, $\varphi(B) = 2$, $\varphi(C) = 3$, $\varphi(D) = 4$, y $\varphi(E) = 5$.

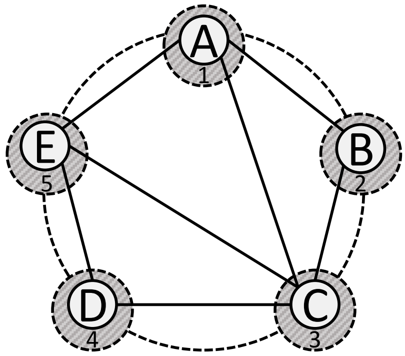
Dada una disposición φ de un grafo G, los investigadores han propuesto varias métricas o medidas, definidas como funciones matemáticas calculadas sobre φ, para evaluar la calidad de un embebido. Algunas de las métricas más relevantes son el corte de aristas (más conocida como *cutwidth* [9], [45]), la longitud de aristas (*bandwidth* [32], [38] o *antibandwidth* [3], [26]), la bisectriz de aristas o vértices (e*dge/vertex bisection* [19], [21]) y el perfil (*profile* [33], [44]). Para una revisión extensa de estas métricas, véase [11].

III. Revisión de las heurísticas constructivas para los GLP

Las heurísticas constructivas son métodos que se utilizan para construir gradualmente una solución factible para un problema dado. Estos métodos comienzan con una solución parcial vacía y añaden componentes de manera incremental hasta que se logra una solución completa. En el contexto de los GLP, esto implica establecer una asignación φ entre los vértices de V_G y V_H.



a Embebdido de G en un grafo camino H_P.



b Embebido de G en un grafo ciclo H_C.

Figura 1: Ejemplos de embebidos de un mismo grafo de entrada G en grafos huésped con diferente estructura.

La creación de una solución para los GLP puede ser bastante sencilla y directa. Un método común y simple implica seleccionar aleatoriamente un vértice de V_G y asignarlo a un vértice de V_H. Este procedimiento se repite hasta que cada vértice en G ha sido asignado a un vértice en H. Aunque este método puede producir soluciones de calidad inferior a otros métodos más avanzados, puede ser útil en ciertos escenarios, lo cual se discutirá más adelante.

Partiendo de un constructivo aleatorio, los investigadores han tratado de aportar conocimiento a la construcción de una solución por medio de criterios heurísticos, también conocidos como criterios voraces [28]. En el estudio realizado se han detectado dos fines principales de los criterios voraces: aquellos cuyo objetivo es seleccionar un vértice del grafo de entrada para asignarlo a la solución o, por el contrario, aquellos que, dado un vértice del grafo de entrada, tratan de encontrar el mejor vértice del grafo huésped para asignarlo. También, en esta investigación, se ha identificado una clasificación de los constructivos en función del número de vértices que añaden a la solución en cada una de sus iteraciones: constructivos que añaden un único vértice en cada paso del método, o constructivos que añaden un conjunto de vértices en cada iteración. A continuación, se presentan los constructivos más relevantes en el campo de los GLP para esta clasificación realizada.

En primer lugar, se revisan los criterios voraces para la selección de vértices del grafo de entrada. A su vez, para este fin, se pueden encontrar dos tipos de criterios, aquellos que buscan colocar los vértices adyacentes del grafo de entrada manera próxima en el embebido o, por el contrario, aquellos que tratan de posicionarlos lo más alejados posible.

Para generar soluciones en las que los vértices del grafo de entrada estén localizados próximamente en el embebido, el enfoque predominante es el propuesto por McAllister en 1999 [30]. En concreto, McAllister presentó un método que determina el vértice de V_G que tiene una mayor afinidad con los vértices adyacentes que ya están en la solución. Para ello, estableció una función matemática de evaluación de cada vértice candidato. En concreto, para cada iteración del constructivo y, para cada vértice $u \in V_G$, define dos conjuntos de vértices. Uno de los conjuntos contiene los vértices adyacentes a v que forman parte de la solución, mientras que el otro contiene los vértices adyacentes a v que no forman parte de la solución. La diferencia de las cardinalidades de ambos conjuntos resulta en un valor numérico que cuantifica, la "urgencia" o "prioridad" que tiene un vértice para ser añadido a la solución. Este concepto ha sido ampliamente utilizado, especialmente para aquellos problemas relacionados con la minimización del *cutwidth* o del *bandwdith* [9], [42], [43]. También, ha sido adaptado, añadiendo una parametrización extra, que permite ajustar el peso que tiene cada una de los factores que influyen en la evaluación de los vértices, lo que permite un ajuste diferente dependiendo de la función objetivo a optimizar y los grafos de entrada [6], [8]. Dentro de esta subfamilia de criterios, algunos investigadores proponen ordenar los vértices del grafo de entrada en función de su grado para, posteriormente, asignarlos secuencialmente [24], [34].

A diferencia del criterio de McAllister, que incorpora un único vértice a la solución en cada iteración, también se pueden encontrar en la literatura constructivos que añaden más de un vértice en cada iteración con el mismo fin. Por ejemplo, en [36], [41] se propone usar cliqués para incorporar conjuntos de vértices altamente conectados a la solución. De manera similar en [21] se propone la identificación de clústeres jerárquicos para generar una solución parcial inicial y, posteriormente, aplicar búsquedas locales que exploran cada clúster de manera independiente. Por último, en [20], [47] se presentan tres heurísticas constructivas (denotadas como H1, H2 y H3) que se basan en la generación de subconjuntos de vértices a partir del cálculo de los ciclos máximos del grafo.

A continuación, se presentan los constructivos cuyo fin es colocar los vértices adyacentes del grafo de entrada lo más alejados posibles los unos de los otros en el grafo huésped. Esta estrategia, se puede considerar una variantes de la anterior y da lugar a una nueva subfamilia de algoritmos constructivos. No obstante, este tipo de algoritmos han sido significativamente menos estudiados en la literatura. En este sentido, la propuesta predominante es la de Díaz *et al.*, que proponen un método para ordenar los vértices de V_G garantizando, siempre que sea posible, que no haya dos vértices adyacentes en el grafo de entrada que sean asignados secuencialmente en el embebido [11]. La puesta en práctica de la idea anterior, ha sido posteriormente aplicada en diferentes trabajos [3], [5], [25], [46]. Para ello, varios investigadores han propuesto realizar una ordenación de los vértices del grafo de entrada en niveles, utilizando los algoritmos *Breadth-First Search* (BFS) y *Depth-First Search* (DFS) [51]. En estas ordenaciones, los vértices de niveles no consecutivos nunca serán adyacentes entre sí.

La segunda familia de criterios para los constructivos se centra en la selección de un vértice del grafo huésped que albergará un vértice seleccionado anteriormente del grafo de entrada. Por un lado, se han identificado múltiples propuestas en las que se utiliza la función objetivo del problema para guiar la selección de dicho vértice [5], [34], [42]. Aunque el uso de la función objetivo como estrategia de guía es una de las propuestas que mejores soluciones obtiene, generalmente tiene dos inconvenientes: i) computar la función objetivo suele ser una tarea computacionalmente compleja y costosa, por lo que calcularla para cada vértice del grafo huésped e iteración suele ser aún más ineficiente, y ii) se intensifica la búsqueda en una región concreta del espacio de soluciones, es decir, las soluciones generadas son poco diversas. Por otro lado, existen otras propuestas más simples y directas de computar, que consisten en la definición de un orden lógico o secuencial basado en la estructura del grafo huésped. Por ejemplo, si el grafo es un camino, se comenzarían a asignar los vértices del grafo huésped por un extremo para, a continuación, asignar el vértice adyacente al anteriormente seleccionado, terminando en el extremo opuesto. A modo de ejemplo, en [6] se plantean diferentes algoritmos de secuenciación para grafos de tipo rejilla. Entre las opciones de selección de vértices del grafo huésped se pueden encontrar propuestas basadas en geometría, por ejemplo: en la asignación de un vértice en la bisectriz, mediatriz o punto medio de algunos de sus vértices adyacentes. En concreto en [20] se proponen tres heurísticas constructivas (denotadas como H3, H4 y H5) que exploran esta estrategia. También, en [24] se propone una función para computar la distancia, medida en un grafo huésped rejilla, entre el vértice del grafo de entrada a asignar y sus vértices ayacentes que ya forman parte de la solución.

Más allá de los criterios a utilizar en la construcción, se pueden identificar dos propiedades que pueden ser comunes a todos ellos: adaptabilidad y aleatoriedad.

Respecto a la adaptabilidad de un constructivo se distinguen constructivos estáticos y dinámicos. Las heurísticas estáticas usan criterios fijos y funcionan bien cuando las características del problema son estables. Las heurísticas dinámicas ajustan sus criterios basándose en la solución parcial que está siendo construida, lo que suele dar mejores resultados en problemas complejos o cambiantes. Por ejemplo, los constructivos basados en la propuesta de McAllister o, en la función objetivo del problema, actualizan el criterio voraz para cada vértice en cada iteración [9], [42], [43].

Respecto a la aleatoriedad de los constructivos, existen tres enfoques principales: determinista, aleatorizado y semialeatorizado. Los métodos deterministas toman siempre decisiones basadas en criterios predefinidos [7], [9]. Los métodos aleatorizados introducen aleatoriedad en una de las decisiones a la hora de seleccionar un vértice [38]. Por último, los enfoques semialeatorizados combinan decisiones deterministas y aleatorias [13], [34], [45]. Por lo general, se ha observado una tendencia de métodos aleatorios o semialeatorios para obtener un conjunto diverso de soluciones que inicialice la población de la familia de algoritmos poblacionales, que han sido también utilizados en el contexto de los GLP [25], [49]. Por último, los métodos deterministas se utilizan cuando se requieren soluciones de alta calidad rápidamente [7], [9], [20]. Una de las metaheurísticas que mejor aprovecha esta propiedad es GRASP (del inglés, *Greedy Randomized Adaptive Search Procedure*) [34], [39], [40]. GRASP aleatoriza los procedimientos en base a una distribución estadística, combinando los principios voraces y de aleatoriedad para explorar el espacio de soluciones de manera efectiva [13], [45].

Además de las propiedades fundamentales identificadas anteriormente, las heurísticas constructivas han sido combinadas, en el contexto de los GLP, con otras técnicas avanzadas para mejorar su rendimiento. Por ejemplo, mediante la incorporación de algoritmos de búsqueda local [7], [9], [49] o la integración de ideas más complejas establecidas en marcos metaheurísticos. Un ejemplo de esto es el uso de las ideas provinientes de la metaheurística *Tabu Search* [16], que mantiene una lista de soluciones (o propiedades de estas) exploradas recientemente y prohíbe temporalmente generar soluciones similares a las recientemente construidas [8], [16], [37]. Otra metaheurística, *Iterated Greedy*, aplica repetidas fases de construcción, destrucción y reconstrucción para estudiar la diversidad del espacio de búsqueda [6], [27]. También, en [33] se presenta un constructivo que hibrida la metaheurística multiarranque con *Simulated Annealing* [18]. Por último, cabe mencionar la propuesta de constructivos que, inspirados en ideas de la metodología de *Strategic Oscillation* [8], [17] generan soluciones infactibles y, posteriormente, utilizan operadores de reparación para generar una solución factible [5].

Por último, cabe mencionar la existencia de constructivos específicos para problemas concretos fundamentados en teoremas matemáticos. Por ejemplo, en [22], [23] se presenta un método de secuenciación espectral para construir una solución para grafos huésped camino. Este método se basa en las propiedades espectrales de la matriz de adyacencia del grafo, concretamente computando los autovalores y autovectores. Los autovalores representan la "importancia" de cada nodo en el grafo, mientras que los autovectores proporcionan información sobre la conectividad de cada uno de ellos.

IV. Marco General para la Propuesta de Heurísticas Constructivas

Extrayendo algunas de las ideas observadas en la revisión de las heurísticas constructivas para los GLP, se propone un marco general y flexible que engloba los aspectos fundamentales de estas heurísticas. Este marco se estructura en torno a cuatro componentes principales: 1) selección de vértices, 2) adaptabilidad, 3) aleatoriedad y 4) uso de iteraciones.

La **selección de vértices** es el primer aspecto a considerar al plantear un método de construcción de una solución. Este proceso puede implicar la selección de un solo vértice o de varios vértices simultáneamente, ya sea del grafo de entrada o del grafo huésped. Las decisiones sobre qué vértices seleccionar y dónde colocarlos pueden ser interdependientes, es decir, la elección de un vértice en V_G puede influir en la elección de un vértice en V_H y viceversa.

La **adaptabilidad** se refiere a si los criterios para la selección de vértices dependen del estado parcial de la solución, o son completamente independientes. En caso de que dependan de la solución, los criterios de selección de los vértices deberán ser recalculados en cada iteración del constructivo. Las heurísticas estáticas, que utilizan criterios fijos, suelen funcionar bien cuando las características del problema son estables. Por otro lado, las heurísticas dinámicas, que ajustan sus criterios en función de la solución en construcción, suelen dar mejores resultados en problemas complejos o cambiantes.

La **aleatoriedad** consiste en la introducción de elementos aleatorios en el proceso de construcción de la solución. Esta aleatoriedad puede introducirse en diferentes puntos del marco propuesto, ya sea en aspectos concretos como la selección de cada uno de los vértices, o de manera más general en el procedimiento de construcción.

Las **iteraciones** hacen referencia al número de veces que se ejecuta el constructivo. En algunos casos, puede ser beneficioso construir múltiples soluciones diversas si se dispone de tiempo para ello, mientras que, en otros casos, puede ser preferible optar por una única solución de partida. En este caso, el criterio de finalización puede ser un número máximo de iteraciones, un numero de iteraciones sin encontrar una solución mejor, un tiempo de corte, o técnicas más avanzadas, como estrategias probabilísticas como las que se presentan en [2].

En este trabajo se realiza una propuesta algorítmica que recoge las ideas anteriores. Esta propuesta se presenta en Algoritmo 1, cuyo pseudocódigo ilustra la aplicación de este marco general para la construcción de soluciones para un GLP. Concretamente, este algoritmo es un ejemplo de cómo se pueden combinar los componentes de selección de vértices, adaptabilidad, aleatoriedad e iteraciones para formar una heurística constructiva eficaz y flexible. Nótese que este algoritmo no es específico para un problema concreto ya que su implementación puede variar dependiendo de las características concretas del problema estudiado.

Algoritmo 1: Marco de construcción general

Require: Grafo de entrada G, grafo huésped H, número de iteraciones n

Inicializar una solución vacía S, que alberga la mejor solución encontrada
for $i = 1$ to n **do**
  Inicializar una solución vacía S'
  while S' no sea una solución completa **do**
    if es necesario **then**
      Recalcular los criterios de selección de vértices
    end if
    if se introduce aleatoriedad **then**
      Modificar los criterios de selección de acuerdo con una distribución estadística
    end if
    Seleccionar j vértices $u_j \in V_G$ y $v_j \in V_H$ según los criterios de selección y en el orden establecido
    Actualizar S' realizando las asignaciones $\varphi(v_j) = u_j$
  end while
  if S es mejor que S' **then**
    Actualizar S con S'
  end if
end for
return S

V. Conclusiones y líneas de trabajo futuras

Este estudio ha proporcionado una visión detallada de la situación actual de las heurísticas constructivas existentes para los GLP, destacando la importancia de la selección de vértices, la adaptabilidad, la aleatoriedad y las iteraciones en la construcción de soluciones. Además, se ha propuesto un marco general y flexible que engloba estos aspectos fundamentales, proporcionando una base sólida para el desarrollo de futuras heurísticas constructivas. Los algoritmos y estrategias desarrollados para estos problemas son fácilmente extrapolables a otros contextos, lo que amplía aún más su relevancia y aplicabilidad. De manera similar, constructivos o criterios heurísticos empleados en otras áreas podrían ser adaptados en el contexto de los GLP.

Como futuras líneas de investigación, en primer lugar, se propone la evaluación experimental del marco teórico propuesto. Concretamente, sería interesante observar si los mejores algoritmos del estado del arte para problemas GLP encajan en este marco teórico. Por otro lado, sería interesante proponer nuevos algoritmos constructivos basados en el marco teórico, para diversos problemas GLP y compararlos con constructivos ya existentes. Al margen de esto, se han identificado otras posibles áreas de trabajos futuros. Para empezar, sería conveniente estudiar la relación entre los distintos problemas dentro de la familia de los GLP para identificar si las soluciones de un problema son también soluciones de calidad para otros problemas. De esta manera, el conocimiento de un problema podría utilizarse para abordar el otro. Otra línea de investigación poco estudiada y con gran potencial es el estudio de métodos constructivos, o procedimientos heurísticos en general, capaces de trabajar con soluciones infactibles y su posterior reparación para obtener la factibilidad.

Otra dirección prometedora es combinar los distintos algoritmos constructivos propuestos para generar una población diversa de soluciones que será utilizada por algoritmos poblacionales. Esto podría mejorar la exploración del espacio de soluciones y potencialmente conducir a soluciones de mayor calidad.

La combinación de técnicas de *Machine Learning* con algoritmos heurísticos y metaheurísticos es otro posible trabajo futuro. En particular, se podrían utilizar para extraer características y propiedades de los grafos de entrada con el fin de determinar, para una función objetivo, cuál es el constructivo más eficaz, o incluso ajustar los parámetros de un algoritmo constructivo concreto. Esta integración de métodos de aprendizaje automático podría proporcionar una mejora significativa en la eficiencia y efectividad de los algoritmos heurísticos, tanto actuales como futuros.

Por último, se propone la investigación y desarrollo de un constructivo general que sea capaz de comportarse adecuadamente para la mayoría de los problemas de tipo GLP y que pueda utilizarse como punto de referencia para otras técnicas heurísticas. Este constructivo general debería ser versátil y adaptable a una amplia gama de situaciones, lo que lo convertiría en una herramienta interesante para los investigadores en el campo de los GLP.

Referencias

[1] Aggarwal, H., Reddy, P.V.S.: Meta-heuristic algorithms for double roman domination problem. Applied Soft Computing **154**, 111306 (2024)

[2] Aiex, R.M., Resende, M.G., Ribeiro, C.C.: Ttt plots: a perl program to create time-to-target plots. Optimization Letters **1**, 355–366 (2007)

[3] Bansal, R., Srivastava, K.: A memetic algorithm for the cyclic antibandwidth maximization problem. Soft Computing **15**, 397–412 (2011)

[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)

[5] Cavero, S., Pardo, E.G., Duarte, A.: A general variable neighborhood search for the cyclic antibandwidth problem. Computational Optimization and Applications **81**(2), 657–687 (2022)

[6] Cavero, S., Pardo, E.G., Duarte, A.: Efficient iterated greedy for the two-dimensional bandwidth minimization problem. European Journal of Operational Research **306**(3), 1126–1139 (2023)

[7] Cavero, S., Pardo, E.G., Duarte, A., Rodriguez-Tello, E.: A variable neighborhood search approach for cyclic bandwidth sum problem. Knowledge-Based Systems **246**, 108680 (2022)

[8] Cavero, S., Pardo, E.G., Glover, F., Martí, R.: Strategic oscillation tabu search for improved hierarchical graph drawing. Expert Systems With Applications **243**, 122668 (2024)

[9] Cavero, S., Pardo, E.G., Laguna, M., Duarte, A.: Multistart search for the cyclic cutwidth minimization problem. Computers & Operations Research **126**, 105116 (2021)

[10] Cohoon, J.P., Sahni, S.: Heuristics for backplane ordering. Journal of VLSI and computer systems **2**(1-2), 37–60 (1987)

[11] Díaz, J., Petit, J., Serna, M.: A survey of graph layout problems. ACM Computing Surveys (CSUR) **34**(3), 313–356 (2002)

[12] Drira, A., Pierreval, H., Hajri-Gabouj, S.: Facility layout problems: A survey. Annual reviews in control **31**(2), 255–267 (2007)

[13] Duarte, A., Martí, R., Resende, M.G., Silva, R.M.: Grasp with path relinking heuristics for the antibandwidth problem. Networks **58**(3), 171–189 (2011)

[14] Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified np-complete problems. In: Proceedings of the sixth annual ACM symposium on Theory of computing. pp. 47–63 (1974)

[15] Gavril, F.: Some np-complete problems on graphs. In: Proc. Conf. on Inform. Sci. and Systems, 1977. pp. 91–95 (1977)

[16] Glover, F., Campos, V., Martí, R.: Tabu search tutorial. a graph drawing application. Top **29**(2), 319–350 (2021)

[17] Glover, F., Hao, J.K.: The case for strategic oscillation. Annals of Operations Research **183**, 163–173 (2011)

[18] Henderson, D., Jacobson, S.H., Johnson, A.W.: The theory and practice of simulated annealing. Handbook of metaheuristics pp. 287–319 (2003)

[19] Herrán, A., Colmenar, J.M., Duarte, A.: A variable neighborhood search approach for the vertex bisection problem. Information Sciences **476**, 1–18 (2019)

[20] Jain, P., Srivastava, K., Saran, G.: Minimizing cyclic cutwidth of graphs using a memetic algorithm. Journal of Heuristics **22**, 815–848 (2016)

[21] Jin, Y., Xiong, B., He, K., Hao, J.K., Li, C.M., Fu, Z.H.: Clustering driven iterated hybrid search for vertex bisection minimization. IEEE Transactions on Computers **71**(10), 2370–2380 (2021)

[22] Juvan, M., Mohar, B.: Optimal linear labelings and eigenvalues of graphs. Discrete Applied Mathematics **36**(2), 153–168 (1992)

[23] Kardam, Y.S., Srivastava, K.: Tabu-embedded simulated annealing algorithm for profile minimization problem. In: Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Volume 1, pp. 165–179. Springer (2020)

[24] Khandelwal, A., Srivastava, K., Saran, G.: Grid bandwidth Minimization Problem: Simulated Annealing Approach. Mapana Journal of Sciences **22**, 101–128 (2023). https://doi.org/https://doi.org/10.12723/mjs.sp1.9

[25] Lim, A., Lin, J., Xiao, F.: Particle swarm optimization and hill climbing for the bandwidth minimization problem. Applied Intelligence **26**, 175–182 (2007)

[26] Lozano, M., Duarte, A., Gortázar, F., Martí, R.: Variable neighborhood search with ejection chains for the antibandwidth problem. Journal of Heuristics **18**, 919–938 (2012)

[27] Lozano, M., Rodriguez-Tello, E.: Population-based iterated greedy algorithm for the s-labeling problem. Computers & Operations Research **155**, 106224 (2023)

[28] Martí, R., Reinelt, G., Martí, R., Reinelt, G.: Heuristic methods. The linear ordering problem: exact and heuristic methods in combinatorial optimization pp. 17–40 (2011)

[29] Martín-Santamaría, R., Cavero, S., Herrán, A., Duarte, A., Colmenar, J.M.: A Practical Methodology for Reproducible Experimentation: An Application to the Double-Row Facility Layout Problem. Evolutionary Computation **32**(1), 69–104 (2024)

[30] Mcallister, A.J.: A new heuristic algorithm for the linear arrangement problem. Tech. rep., New Brunswick, CA: University of New Brunswick (1999)

[31] Michiels, W., Aarts, E.H., Korst, J.: Theory of local search. Handbook of heuristics pp. 299–339 (2018)

[32] Mladenovic, N., Urosevic, D., Pérez-Brito, D., García-González, C.G.: Variable neighbourhood search for bandwidth reduction. European Journal of Operational Research **200**(1), 14–27 (2010)

[33] Palubeckis, G.: A variable neighborhood search and simulated annealing hybrid for the profile minimization problem. Computers & Operations Research **87**, 83–97 (2017)

[34] Pantrigo, J.J., Martí, R., Duarte, A., Pardo, E.G.: Scatter search for the cutwidth minimization problem. Annals of Operations Research **199**, 285–304 (2012)

[35] Papadimitriou, C.H.: The np-completeness of the bandwidth minimization problem. Computing **16**(3), 263–270 (1976)

[36] Pardo, E.G., Martí, R., Duarte, A.: Linear layout problems. In: Handbook of Heuristics, pp. 1025–1049. Springer (2018)

[37] Pastore, T., Martínez-Gavara, A., Napoletano, A., Festa, P., Martí, R.: Tabu search for min-max edge crossing in graphs. Computers & Operations Research **114**, 104830 (2020)

[38] Ren, J., Hao, J.K., Rodriguez-Tello, E., Li, L., He, K.: A new iterated local search algorithm for the cyclic bandwidth problem. Knowledge-Based Systems **203**, 106136 (2020)

[39] Resende, M.G., Ribeiro, C.C.: Greedy randomized adaptive search procedures: Advances, hybridizations, and applications. Handbook of metaheuristics pp. 283–319 (2010)

[40] Resende, M.G., Ribeiro, C.C.: Greedy randomized adaptive search procedures: Advances and extensions. Handbook of metaheuristics pp. 169–220 (2019)

[41] Robles, M., Cavero, S., Pardo, E.G.: Bvns for the minimum sitting arrangement problem in a cycle. In: International Conference on Variable Neighborhood Search. pp. 82–96. Springer (2022)

[42] Rodríguez-García, M.Á., Sanchez-Oro, J., Rodriguez-Tello, E., Monfroy, E., Duarte, A.: Two-dimensional bandwidth minimization problem: Exact and heuristic approaches. Knowledge-Based Systems **214**, 106651 (2021)

[43] Rodriguez-Tello, E., Hao, J.K., Torres-Jimenez, J.: An effective two-stage simulated annealing algorithm for the minimum linear arrangement problem. Computers & Operations Research **35**(10), 3331–3346 (2008)

[44] Sánchez-Oro, J., Laguna, M., Duarte, A., Martí, R.: Scatter search for the profile minimization problem. Networks **65**(1), 10–21 (2015)

[45] Santos, V.G.M., de Carvalho, M.A.M.: Tailored heuristics in adaptive large neighborhood search applied to the cutwidth minimization problem. European Journal of Operational Research **289**(3), 1056–1066 (2021)

[46] Sinnl, M.: A note on computational approaches for the antibandwidth problem. Central European Journal of Operations Research **29**(3), 1057–1077 (2021)

[47] Soto, C., Del Ángel-Martínez, E., Fraire-Huacuja, H., Dorronsoro, B., Rangel, N., Cruz-Reyes, L.: Two novel branch and bound algorithms for the vertex bisection problem. Expert Systems with Applications **190**, 116169 (2022)

[48] Stützle, T., Ruiz, R.: Iterated greedy. Handbook of heuristics pp. 547–577 (2018)

[49] Sun, W., Hao, J.K., Li, W., Wu, Q.: An adaptive memetic algorithm for the bidirectional loop layout problem. Knowledge-Based Systems **258**, 110002 (2022)

[50] Zanakis, S.H., Evans, J.R., Vazacopoulos, A.A.: Heuristic methods and applications: a categorized survey. European Journal of Operational Research **43**(1), 88–110 (1989)

[51] Zhou, R., Hansen, E.A.: Breadth-first heuristic search. Artificial Intelligence **170**(4-5), 385–408 (2006)