

Proyecto DAA: Problema de la Empresa Telefónica - Partición Cromática de Costo Mínimo (MCCPP)

Yesenia Valdés Rodríguez (C411)

Laura Martir Beltrán (C411)

Adrián Hernández Castellanos (C412)

Índice

1. Formalización del Problema	1
1.1. Problema: La empresa Telefónica	1
1.2. Contexto del Problema: La Asignación de Frecuencias en ConectaMax Telecom	2
1.3. Definición Formal	2
1.4. Modelado de Programación Entera Lineal (ILP 0-1)	3
2. Análisis de Complejidad Computacional	4
2.1. Clase de Complejidad	4
2.2. Planteamiento del Problema de k -Colorabilidad	4
2.3. Demostración de NP-Hardness mediante Reducción Polinomial	4
2.4. Jerarquía de Aproximabilidad Polinomial: PTAS, APX, APX-Hardness, APX-Complete y log-APX	5
2.5. Demostración de APX-Hardness del MCCPP	6
2.5.1. Problema de Origen APX-Complete: Minimum Vertex Cover (MVC)	6
2.5.2. Implicaciones Teóricas: Exclusión del PTAS y APX-Completeness	8
2.5.3. Imposibilidad de PTAS	8
2.5.4. Análisis Riguroso de Pertenencia a APX (MCCPP)	8
2.6. Análisis de Complejidad Parametrizada	10
2.6.1. Definición Formal de W-Hardness	10
2.6.2. Demostración de que el MCCPP es $\mathbf{W[1]}$ -Hard cuando se parametriza por $ F $	11
2.6.3. Demostración de que el MCCPP es $\mathbf{W[1]}$ -Hard cuando se parametriza por Treewidth	11
2.6.4. Búsqueda de Parámetros Alternativos FPT para el MCCPP	17
3. Diseño de Soluciones Algorítmicas	19
3.1. Soluciones Exactas (Bases de Referencia)	20
3.1.1. Algoritmo de Fuerza Bruta: Enumeración Completa	20
3.1.2. Resolución mediante Programación Entera Lineal (ILP)	20
3.1.3. Programación Dinámica (DP) para Grafos Estructurados	21
3.2. Algoritmos de Aproximación	23
3.2.1. Algoritmo de Aproximación con Garantía Teórica ($R \leq O(\ln V)$)	23
3.2.2. Algoritmos de Aproximación Estructural ($R = O(\sqrt{ V })$)	24
3.3. Heurísticas Greedy Cost-Aware	24
3.3.1. Estrategia Largest First (LF) Cost-Aware	24

3.3.2. Estrategia DSATUR (Degree of Saturation) Cost-Aware	26
3.3.3. Estrategia Recursive Largest First (RLF) Cost-Aware	27
3.4. Metaheurísticas: Exploración del Espacio de Soluciones	28
3.4.1. Simulated Annealing (SA)	28
3.4.2. Trajectory Search Heuristic + Path Relinking (TSH+PR)	29
3.5. Conclusiones	30
4. Implementación y Análisis Experimental	31
4.1. Generación de Instancias	31
4.2. Comparación Empírica	31
4.3. Entregables	31

1. Formalización del Problema

1.1. Problema: La empresa Telefónica

En ConectaMax Telecom, nuestra misión es proporcionar una conectividad móvil ininterrumpida y de alta calidad a millones de usuarios. Para lograrlo, operamos una extensa y compleja red de torres de telefonía celular. La eficiencia y la calidad de nuestra red dependen críticamente de cómo gestionamos uno de nuestros recursos más valiosos: el espectro de radiofrecuencias.

Nos enfrentamos a un desafío operativo y financiero significativo en la asignación de frecuencias a nuestras torres. Disponemos de un conjunto limitado de frecuencias que podemos utilizar. La regla fundamental, dictada por la física y la regulación, es que dos torres que están geográficamente muy cerca una de la otra (y que, por lo tanto, podrían interferir entre sí) no pueden operar en la misma frecuencia. Esto es vital para evitar la degradación de la señal y asegurar un servicio fiable.

La complejidad adicional, y donde reside nuestro mayor reto, es que el costo de operar una torre con una frecuencia específica no es uniforme. Asignar una frecuencia particular a una torre determinada conlleva costos variables. Estos costos pueden deberse a múltiples factores:

- **Equipamiento de la Torre:** Algunas torres tienen hardware más antiguo o especializado que es más eficiente con ciertas frecuencias, mientras que otras frecuencias podrían requerir adaptaciones o un mayor consumo energético en ese mismo equipo.
- **Consumo Energético:** La eficiencia energética de la transmisión varía según la frecuencia y el tipo de equipo de la torre, impactando directamente nuestras facturas de electricidad.
- **Regulaciones Locales y Licencias:** En ciertas áreas o para bandas de frecuencia específicas, pueden existir tarifas de licencia más elevadas o regulaciones que exigen configuraciones especiales, aumentando los costos operativos.

Nuestro objetivo principal es asignar una frecuencia a cada una de nuestras torres de tal manera que:

1. **Se eviten todas las interferencias:** Ninguna torre cercana a otra utilice la misma frecuencia.
2. **Se minimice el costo operativo total:** La suma de los costos individuales de asignar cada frecuencia a cada torre sea la más baja posible.

Una gestión subóptima de esta asignación no solo puede generar interferencias que afectan la calidad del servicio y la satisfacción del cliente, sino que también puede resultar en millones de dólares en costos operativos innecesarios.

1.2. Contexto del Problema: La Asignación de Frecuencias en ConectaMax Telecom

El problema presentado por la empresa ConectaMax Telecom, centrado en la asignación óptima de frecuencias a sus torres de telefonía celular, se traduce directamente en un problema de optimización combinatoria conocido como el **Problema de Partición Cromática de Costo Mínimo (Minimum Cost Chromatic Partition Problem, MCCPP)**.

La estructura de la red telefónica y sus restricciones operativas se modelan de la siguiente manera:

- **Grafo de Interferencia $G = (V, E)$:** Las Torres de Telefonía Celular se representan como los vértices (V) del grafo. Las aristas (E) representan las proximidades geográficas y el potencial de interferencia, donde una arista $\{u, v\} \in E$ indica que la Torre u y la Torre v no pueden operar en la misma frecuencia.
- **Etiquetas y Frecuencias F :** El conjunto limitado de frecuencias disponibles se convierte en el conjunto finito de etiquetas (F) (o colores).
- **Restricción Operacional (Coloración Propia):** La regla de “evitar todas las interferencias” se traduce en la restricción de coloración propia, exigiendo que dos vértices adyacentes no posean la misma etiqueta:

$$\phi(u) \neq \phi(v)$$

- **Costo Operativo C :** La variabilidad en el costo de operar una torre con una frecuencia específica (debido a equipamiento, energía, licencias) se captura mediante la matriz de costos C , donde $c_{v,f}$ es el costo de asignar la frecuencia f a la torre v .
- **Objetivo (Minimización del Costo):** El objetivo de minimizar el costo operativo total se convierte en:

$$\min \sum_{v \in V} c_{v,\phi(v)}$$

El MCCPP es, por lo tanto, la formalización precisa del desafío de la empresa telefónica, ya que busca una partición válida de los vértices (torres) en clases de color (frecuencias) que minimice el costo total asociado a dicha partición.

1.3. Definición Formal

Nombre del Problema: Problema de Partición Cromática de Costo Mínimo (MCCPP).

Instancia de Entrada: Una tupla $\langle G, F, C \rangle$, donde:

- $G = (V, E)$: Un grafo no dirigido, con $|V| = n$ y E el conjunto de aristas.
- F : Conjunto finito de etiquetas (colores/frecuencias), $|F| = k$.
- C : Matriz de costos donde $c_{v,f}$ es el costo de asignar la etiqueta f al vértice v .

Función Objetivo: Encontrar una función de asignación de etiquetas $\phi : V \rightarrow F$ que minimice el costo total de la asignación.

Restricción de Factibilidad:

- Para toda arista $\{u, v\} \in E$, se debe cumplir que $\phi(u) \neq \phi(v)$.

1.4. Modelado de Programación Entera Lineal (ILP 0-1)

El MCCPP se formaliza rigurosamente mediante el siguiente modelo de Programación Entera Lineal con variables binarias, que servirá de base para la demostración de complejidad y para la implementación de un algoritmo exacto (fuerza bruta/Branch-and-Bound en instancias pequeñas).

Parámetros:

- V : Conjunto de vértices.
- F : Conjunto de etiquetas disponibles.
- $c_{v,f}$: Costo de asignar la etiqueta $f \in F$ al vértice $v \in V$.

Variables de Decisión:

- $x_{v,f} \in \{0, 1\}$: Variable binaria. $x_{v,f} = 1$ si el vértice v es asignado a la etiqueta f ; $x_{v,f} = 0$ en caso contrario.

Función Objetivo (Minimización del Costo Total):

$$\min \sum_{v \in V} \sum_{f \in F} c_{v,f} \cdot x_{v,f}$$

Restricciones de Factibilidad:

Asignación Única por Vértice: Cada vértice debe ser asignado a exactamente una etiqueta.

$$\sum_{f \in F} x_{v,f} = 1 \quad \forall v \in V$$

Restricción de No Conflicto (Coloración Propia): Dos vértices adyacentes no pueden recibir la misma etiqueta.

$$x_{u,f} + x_{v,f} \leq 1 \quad \forall \{u, v\} \in E, \forall f \in F$$

Integridad:

$$x_{v,f} \in \{0, 1\} \quad \forall v \in V, \forall f \in F$$

2. Análisis de Complejidad Computacional

2.1. Clase de Complejidad

El Problema de Partición Cromática de Costo Mínimo (MCCPP) es un problema NP-Hard en su versión de optimización. Dado que su restricción de factibilidad es idéntica a la del problema k -Colorability (reconocido como NP-Completo), la búsqueda de la solución óptima de costo mínimo es, por lo menos, tan difícil como la búsqueda de cualquier coloración válida.

2.2. Planteamiento del Problema de k -Colorabilidad

El **problema de k -Colorabilidad** (también conocido como k -Coloración Propia) plantea la siguiente cuestión fundamental: dado un grafo no dirigido $G = (V, E)$ y un entero positivo k , ¿es posible asignar colores a los vértices de G utilizando a lo sumo k colores diferentes, de tal forma que ningún par de vértices adyacentes comparten el mismo color?

Este problema representa uno de los problemas clásicos de NP-Compleitud y tiene aplicaciones en áreas como *scheduling*, asignación de registros en compiladores, y problemas de asignación de frecuencias en redes.

2.3. Demostración de NP-Hardness mediante Reducción Polinomial

Se demuestra que el MCCPP es NP-Hard mediante una reducción en tiempo polinomial del problema de decisión k -Colorability (una instancia canónica de NP-Completo) al MCCPP.

Problema Fuente (NP-Completo): k -Colorability.

Instancia: $\langle G, k \rangle$, donde $G = (V, E)$ es un grafo y k es un entero positivo.

Pregunta: ¿Existe una coloración propia de G utilizando a lo sumo k colores?

Reducción Polinomial \mathcal{R} :

Dada una instancia $\langle G, k \rangle$ del k -Colorability, construimos la siguiente instancia $\langle G', F, C \rangle$ del MCCPP:

- **Grafo G' :** Mantenemos el mismo grafo: $G' = G$.
- **Conjunto de Etiquetas F :** Definimos el conjunto de etiquetas como $F = \{f_1, f_2, \dots, f_k\}$, con $|F| = k$.

- **Matriz de Costos C :** Definimos un costo uniforme y nulo:

$$c_{v,f} = 0 \quad \forall v \in V, \forall f \in F$$

- **Umbral de Decisión Z_{\max} :** Establecemos el objetivo de optimización: $\min Z$. La pregunta de decisión asociada al MCCPP será si el costo mínimo Z^* es ≤ 0 .

Conclusión de la Reducción:

- Si G es k -coloreable, existe una asignación $\phi : V \rightarrow F$ que satisface la restricción de coloración propia. Como todos los costos son cero ($c_{v,f} = 0$), el costo total de esta asignación es $Z = \sum c_{v,\phi(v)} = 0$. Por lo tanto, el costo mínimo óptimo Z^* es 0, y $Z^* \leq 0$.
- Si G no es k -coloreable, no existe ninguna asignación válida $\phi : V \rightarrow F$ que cumpla la restricción. Por lo tanto, no existe una solución factible, y no podemos alcanzar el costo $Z \leq 0$.

Dado que se ha reducido el k -Colorability al MCCPP en tiempo polinomial (la construcción solo requiere asignar $k \cdot n$ costos a cero), y la respuesta a la instancia del MCCPP determina la respuesta a la instancia del k -Colorability, el MCCPP (en su versión de optimización) es NP-Hard.

2.4. Jerarquía de Aproximabilidad Polinomial: PTAS, APX, APX-Hardness, APX-Complete y log-APX

Dada la **NP-Hardness** del Problema de Partición Cromática de Costo Mínimo (MCCPP), el análisis de su **aproximabilidad** es crucial. Este análisis establece los límites teóricos sobre cuán cerca de la solución óptima Z^* podemos esperar llegar en tiempo polinomial. Para ello, es indispensable definir la jerarquía de las clases de complejidad de aproximación.

- **PTAS (Polynomial Time Approximation Scheme):** La clase **PTAS** incluye los problemas de optimización que admiten un esquema de aproximación en tiempo polinomial. Esto significa que, para cualquier constante $\epsilon > 0$ arbitrariamente pequeña, existe un algoritmo en tiempo polinomial $O(n^{f(1/\epsilon)})$ que produce una solución Z_{Alg} tal que $Z_{Alg} \leq (1 + \epsilon)Z^*$ (para problemas de minimización). La complejidad temporal puede crecer exponencialmente con $1/\epsilon$, pero es polinomial en el tamaño de la entrada n .
- **APX (Approximable):** La clase **APX** (Approximable) contiene todos los problemas NPO (NP Optimization) que admiten un algoritmo de aproximación de factor constante c en tiempo polinomial, donde $c \geq 1$ es independiente del tamaño de la instancia n . Formalmente, $Z_{Alg} \leq c \cdot Z^*$. La clase **PTAS** es un subconjunto estricto de **APX**, asumiendo que $P \neq NP$. Pertener a APX implica que la calidad de la solución está acotada por una constante.

- **APX-Hard:** Un problema de optimización es **APX-Hard** si todo problema en **APX** puede ser reducido a él mediante una reducción que preserva la aproximación (típicamente una L-Reducción o PTAS-Reducción). La consecuencia fundamental de la APX-Hardness es que, si $P \neq NP$, el problema **no admite un PTAS**. La demostración de APX-Hardness del MCCPP (Sección 2.5) establece precisamente esta limitación.
- **APX-Complete:** Un problema es **APX-Complete** si es **APX-Hard** y, además, pertenece a la clase **APX**. Esto implica que existe un algoritmo de aproximación de factor constante c , pero que no existe un esquema de aproximación arbitrariamente cercano al óptimo (PTAS), salvo que $P = NP$.
- **log-APX:** Esta clase es relevante para problemas cuya garantía de aproximación es logarítmica. Un problema se clasifica en **log-APX** si admite un algoritmo de aproximación con un factor de rendimiento R acotado por $O(\ln n)$, donde n es el tamaño de la entrada. La clasificación en **log-APX** tiene una implicación más restrictiva que la APX-Hardness: los problemas en **log-APX** (y que no están en APX) **no admiten un algoritmo de aproximación de factor constante c** , a menos que se cumplan condiciones de complejidad teórica muy improbables.

2.5. Demostración de APX-Hardness del MCCPP

Para demostrar que un problema es APX-Hard, se requiere establecer una reducción que preserva la aproximación (típicamente una L-Reducción) desde un problema que ya se sabe es APX-Hard. Esta demostración refuta la posibilidad de esquemas de aproximación arbitrariamente cercanos al óptimo (PTAS), a menos que $P = NP$.

Por lo tanto, se establecerá una L-Reducción robusta desde el Minimum Vertex Cover (MVC).

2.5.1. Problema de Origen APX-Complete: Minimum Vertex Cover (MVC)

El problema de Minimum Vertex Cover (MVC) es un problema de optimización de minimización: dada una instancia de grafo $G = (V, E)$, se busca el subconjunto de vértices $V' \subseteq V$ de tamaño mínimo tal que cada arista en E tenga al menos un extremo en V' .

El MVC es un problema fundamental en complejidad y se sabe que es **APX-Complete** [1]. Esto implica dos hechos cruciales:

1. **APX-Hardness:** MVC no admite un PTAS a menos que $P = NP$ [2].
2. **Pertenencia a APX:** Existe un algoritmo de aproximación en tiempo polinomial con un factor de aproximación constante (una 2-aproximación basada en un algoritmo voraz clásico).

La demostración formal de que MVC es APX-Complete, confirmando su 2-aproximabilidad y su inaproximabilidad constante (cota inferior de 1,3606 si $P \neq NP$), se puede encontrar, por ejemplo, en [3], así como en desarrollos posteriores sobre la brecha de aproximación de Vertex Cover.

El MVC está intrínsecamente ligado al Maximum Independent Set (MIS). Un conjunto de vértices V' es una cubierta de vértices si y solo si su complemento $V \setminus V'$ es un conjunto independiente. Por lo tanto, minimizar $|V'|$ es equivalente a maximizar $|V \setminus V'|$, es decir, maximizar el tamaño del Conjunto Independiente. La dificultad de aproximar MIS se traslada directamente a la dificultad de aproximar MVC.

La reducción \mathcal{R} transforma una instancia de MVC $\langle G \rangle$ en una instancia de MCCPP $\langle G', F, C \rangle$:

- **Grafo G' :** Se mantiene el grafo original, $G' = G = (V, E)$.
- **Conjunto de Etiquetas F :** Se define un conjunto de etiquetas $F = \{f_1, f_2, \dots, f_n\}$, donde $n = |V|$. Este conjunto es suficiente para cualquier coloración.
- **Matriz de Costos C :** Se definen costos binarios de la siguiente manera:
 - **Costo Cero (Color Preferido):** $c_{v,f_1} = 0$, para todo vértice $v \in V$.
 - **Costo Unitario (Otros Colores):** $c_{v,f} = 1$, para todo $v \in V$ y $f \in \{f_2, \dots, f_n\}$.

El objetivo del MCCPP en esta instancia es encontrar una coloración propia $\phi : V \rightarrow F$ que minimice el costo total $Z_{MCCPP} = \sum_{v \in V} c_{v,\phi(v)}$.

Equivalencia de la Optimalidad:

Para minimizar Z_{MCCPP} , se debe maximizar el número de vértices asignados al color f_1 (costo 0). Sea I_1 el conjunto de vértices coloreados con f_1 . Para que la coloración sea válida, I_1 debe ser un conjunto independiente.

El costo óptimo Z_{MCCPP}^* se logra cuando I_1 es el Máximo Conjunto Independiente (I_{\max}), ya que todos los demás colores (f_2, \dots, f_n) tienen costo 1.

$$Z_{MCCPP}^* = 0 \cdot |I_{\max}| + 1 \cdot (|V| - |I_{\max}|) = |V| - |I_{\max}|$$

Dado que el tamaño del Minimum Vertex Cover Z_{MVC}^* es igual a $|V| - |I_{\max}|$, se establece la relación:

$$Z_{MCCPP}^* = Z_{MVC}^*$$

Preservación de la Aproximación:

Si existiera un algoritmo de aproximación A_{MCCPP} para el MCCPP con factor de aproximación constante α , es decir, $Z_{MCCPP}^{Alg} \leq \alpha \cdot Z_{MCCPP}^*$, este algoritmo podría usarse para resolver MVC con el mismo factor:

$$Z_{MVC}^{Alg} = Z_{MCCPP}^{Alg} \leq \alpha \cdot Z_{MCCPP}^* = \alpha \cdot Z_{MVC}^*$$

Dado que el MVC es APX-Hard, la existencia de una L-Reducción con preservación lineal del valor óptimo demuestra que el MCCPP es, a su vez, APX-Hard.

2.5.2. Implicaciones Teóricas: Exclusión del PTAS y APX-Completeness

La demostración de que el MCCPP es APX-Hard tiene consecuencias directas sobre la existencia de esquemas de aproximación.

2.5.3. Imposibilidad de PTAS

La APX-Hardness establece que, si $P \neq NP$, el MCCPP no admite un PTAS. Esta limitación persiste incluso en el caso especial de costos binarios, tal como se demostró en la reducción a MVC.

2.5.4. Análisis Riguroso de Pertenencia a APX (MCCPP)

Un problema es APX-Complete si es APX-Hard y también pertenece a la clase APX (es decir, admite un algoritmo de aproximación de factor constante c).

Aunque el MCCPP es APX-Hard, su pertenencia a APX no está universalmente establecida para grafos generales.

Como la pertenencia de un problema APX-Hard a la clase APX requiere demostrar la existencia de un algoritmo de aproximación con un factor de rendimiento constante $c \geq 1$ en tiempo polinomial, para el MCCPP esta demostración se considera altamente improbable para grafos generales.

Modelado Formal como Cobertura de Conjuntos Ponderada (WSC)

El MCCPP se puede modelar como el **Problema de Cobertura de Conjuntos Ponderada** (Weighted Set Cover, WSC), un problema fundamental de optimización ampliamente estudiado tanto desde el punto de vista algorítmico como de inaproximabilidad [4], [5].

- **Definición Formal de WSC:** Dada una instancia $\langle U, \mathcal{S}, w \rangle$, donde U es el universo de elementos a cubrir ($|U| = n$), $\mathcal{S} = \{S_1, \dots, S_m\}$ es una familia de subconjuntos de U , y $w : \mathcal{S} \rightarrow \mathbb{R}^+$ es la función de peso (costo) de cada subconjunto S_i . El objetivo es encontrar una subcolección $\mathcal{C} \subseteq \mathcal{S}$ tal que $\bigcup_{S_i \in \mathcal{C}} S_i = U$, minimizando el costo total $\sum_{S_i \in \mathcal{C}} w(S_i)$.

Mapeo MCCPP a WSC:

Una instancia $\langle G, F, C \rangle$ del MCCPP se transforma en una instancia $\langle U, \mathcal{S}, w \rangle$ del WSC:

1. **Universo (U):** El conjunto de vértices V del grafo G . El objetivo es “cubrir” todos los vértices, es decir, asignarles un color [4], [5].
2. **Conjuntos Candidatos (\mathcal{S}):** Cada conjunto $S \in \mathcal{S}$ es un par $\langle I, f \rangle$, donde $I \subseteq V$ es un *Conjunto Independiente* (IS) válido en G y $f \in F$ es una frecuencia (color) disponible [4], [5].
3. **Costo del Conjunto $w(S)$:** El costo de elegir el conjunto $S = \langle I, f \rangle$ es el costo total de asignar la frecuencia f a todos los vértices en I :

$$w(S) = \sum_{v \in I} c_{v,f}$$

Una solución válida del WSC (una subcolección de conjuntos independientes que cubre V) corresponde directamente a una partición cromática válida de G con el mismo costo total.

Demostración de Inaproximabilidad Constante (MCCPP \notin APX)

El mejor algoritmo de aproximación conocido para el WSC es el algoritmo **Greedy**, que alcanza una cota asintótica óptima en términos del factor logarítmico $\Theta(\ln n)$ tanto para Set Cover como para su versión ponderada [4], [5].

Sea Z_{WSC}^* el costo óptimo del WSC y Z_{WSC}^{Alg} el costo de la solución obtenida por el algoritmo voraz.

- **Factor de Aproximación Logarítmico:** El algoritmo Greedy garantiza un factor de aproximación R que está acotado superiormente por el número armónico $H(|U|)$, donde $|U| = |V| = n$:

$$R = \frac{Z_{WSC}^{Alg}}{Z_{WSC}^*} \leq H(|U|) = O(\ln n)$$

- **Límite Inferior de Inaproximabilidad:** Se ha demostrado rigurosamente que el WSC (y por extensión el Set Cover estándar) no admite un algoritmo de aproximación de factor constante c . Específicamente, a menos que $NP \subseteq DTIME(n^{\log \log n})$, no existe un algoritmo que garantice un factor de aproximación mejor que $(1 - \varepsilon) \ln n$, para cualquier $\varepsilon > 0$ [4], [5].

Dado que la mejor aproximación garantizada para el MCCPP es logarítmica $O(\ln |V|)$, y existe un límite inferior que excluye factores constantes, se concluye formalmente que el MCCPP **no pertenece** a la clase APX, y por lo tanto, no es APX-Complete. En términos de la jerarquía de aproximación, el MCCPP se clasifica en la clase **log-APX**.

2.6. Análisis de Complejidad Parametrizada

La **Complejidad Parametrizada** ofrece un marco teórico para analizar problemas NP-Hard en función de parámetros específicos que pueden influir en la complejidad computacional. En este contexto, se examina la dificultad del MCCPP cuando se parametriza por el número de frecuencias $k = |F|$, por el treewidth tw del grafo G y otros enfoques.

FPT (Fixed-Parameter Tractable): Un problema parametrizado $\langle I, k \rangle$ es FPT si existe un algoritmo que resuelve cualquier instancia en tiempo $O(f(k) \cdot |I|^c)$, donde f es una función computable que depende únicamente del parámetro k , y c es una constante independiente de k y $|I|$. Esto implica que la complejidad exponencial está confinada al parámetro k , permitiendo soluciones eficientes para valores pequeños de k .

2.6.1. Definición Formal de W-Hardness

La **Complejidad Parametrizada** clasifica los problemas NP-Hard basándose en cómo el parámetro k afecta la complejidad exponencial. Un problema $\langle I, k \rangle$ se clasifica en la jerarquía **W** si no es FPT (asumiendo que **FPT** \neq **W**).

La clase **W[1]** se define mediante problemas que son equivalentes bajo FPT-reducciones al problema *Weighted Weft-1-Depth-d SAT* o, más comúnmente, al problema canónico **k-Clique**.

Definición de W-Hardness: Un problema parametrizado P es **W[1]-Hard** si todo problema en **W[1]** puede ser reducido a P mediante una **FPT-reducción**. Una FPT-reducción es una transformación de una instancia $\langle I, k \rangle$ de P_1 a una instancia $\langle I', k' \rangle$ de P_2 , computable en tiempo $O(f(k) \cdot |I|^c)$, donde el nuevo parámetro k' depende únicamente de k ($k' \leq g(k)$).

La consecuencia fundamental de la **W[1]-Hardness** es que, si se cree que la hipótesis de **FPT** \neq **W[1]** es verdadera, el problema no admite un algoritmo de tiempo $O(f(k) \cdot n^c)$, lo que descarta la kernelización con un tamaño que dependa únicamente de k .

Problema **k-Coloring** es W-Hard

La dificultad de la coloración se traslada directamente al reino de la complejidad parametrizada. Entonces se tiene que el problema **k-Coloring**, parametrizado por el número de colores k , es **W[1]-Hard**, tal como se establece en la literatura clásica de complejidad parametrizada y resultados específicos sobre problemas coloridos parametrizados por treewidth [6].

Fundamentación Académica

Este resultado, que establece que la coloración es paramétricamente difícil, se basa en reducciones desde problemas centrales de la jerarquía **W**, como **k-Clique** o su variante **Multicolored Clique**.

que. La demostración completa se encuentra rigurosamente detallada en trabajos de Flum y Grohe sobre problemas coloridos parametrizados por treewidth y en manuales estándar de complejidad parametrizada [6], [7].

2.6.2. Demostración de que el MCCPP es W[1]-Hard cuando se parametriza por $|F|$

El resultado de la W -Hardness del k -Coloring tiene implicaciones directas para el MCCPP, ya que la restricción de factibilidad del MCCPP es idéntica a la del k -Coloring.

Planteamiento

El Problema de Partición Cromática de Costo Mínimo (MCCPP), parametrizado por el número de frecuencias $k = |F|$, es **W[1]**-Hard.

Demostración

La demostración se logra mediante la **FPT-Reducción trivial** del k -Coloring al MCCPP:

1. Se parte de una instancia del k -Coloring $\langle G, k \rangle$.
2. Se construye la instancia del MCCPP $\langle G, F, C \rangle$ con $|F| = k$, estableciendo todos los costos $c_{v,f} = 0$.
3. El tiempo de esta construcción es $\mathbf{O}(|V| \cdot k)$, que satisface la condición FPT (polinomial en $|V|$, pero la dependencia exponencial es solo en k).
4. El parámetro se preserva: $k' = k$.

Dado que la existencia de una solución de costo cero en el MCCPP es equivalente a la existencia de una k -coloración válida, y la reducción es una FPT-reducción, se concluye que **MCCPP** es **W[1]**-Hard cuando se parametriza por el número de colores k .

2.6.3. Demostración de que el MCCPP es W[1]-Hard cuando se parametriza por Treewidth

Estado de la Cuestión:

Existe literatura más antigua (<2000) donde se afirma que el MCCPP es FPT por tw . Sin embargo, en investigaciones más recientes (2017) se refuta lo afirmado en la literatura inicial sobre el problema: se ha demostrado formalmente mediante una reducción parametrizada rigurosa que el Problema de Partición Cromática de Costo Mínimo (MCCPP), también conocido como Weighted Coloring, es **W-hard** cuando se parametriza por la **treewidth** del grafo, incluso en el caso restringido de bosques (donde la treewidth es exactamente 1) [8].

Equivalencia Formal entre MCCPP y Weighted Coloring

Replanteemos el problema del MCCPP y el Weighted Coloring para establecer su equivalencia formal.

Problema 1: Minimum Cost Chromatic Partition Problem (MCCPP):

El Problema de Partición Cromática de Costo Mínimo se define formalmente de la siguiente manera:

Instancia: Un grafo no dirigido $G = (V, E)$ y una matriz de costos C donde $c_{v,f}$ es el costo de asignar la etiqueta (frecuencia/color) f al vértice v . Un conjunto finito F de etiquetas disponibles con $|F| = k$.

Objetivo: Encontrar una función de asignación $\phi : V \rightarrow F$ (una coloración propia, es decir, $\phi(u) \neq \phi(v)$ para toda arista $\{u, v\} \in E$) que minimice el costo total:

$$Z_{MCCPP} = \min_{\phi \text{ coloración propia}} \sum_{v \in V} c_{v,\phi(v)}$$

Problema 2: Weighted Coloring (Weighted Chromatic Number):

El problema de Weighted Coloring, introducido formalmente por Guan y Zhu, se define como sigue [9]:

Instancia: Un grafo no dirigido $G = (V, E)$ y una función de pesos $w : V \rightarrow \mathbb{R}^+$ que asigna un peso positivo a cada vértice.

Definiciones de Colores y Pesos:

Una coloración propia c de G es una partición $c = (S_i)_{i \in [1,k]}$ del conjunto V en k conjuntos independientes (también llamados estables) S_1, S_2, \dots, S_k .

Dado que cada conjunto S_i es un conjunto independiente de vértices del mismo color, el *peso* del color S_i se define como:

$$w(i) = \max_{v \in S_i} w(v)$$

es decir, el peso máximo de los vértices en ese color.

El *peso total* de una coloración c es:

$$w(c) = \sum_{i=1}^k w(i) = \sum_{i=1}^k \max_{v \in S_i} w(v)$$

Objetivo: Calcular la *weighted chromatic number* $\sigma(G, w)$, definida como el peso mínimo entre todas las coloraciones propias válidas de G :

$$\sigma(G, w) = \min_{c \text{ coloración propia}} w(c)$$

Una coloración c tal que $w(c) = \sigma(G, w)$ se llama coloración óptima ponderada.

Planteamiento: Equivalencia entre MCCPP y Weighted Coloring

El MCCPP y el Weighted Coloring (también conocido como Max-Coloring en la literatura) son **exactamente el mismo problema**, aunque formulados de manera ligeramente diferente. La equivalencia puede demostrarse mediante la siguiente construcción:

1. Transformación MCCPP → Weighted Coloring:

Dada una instancia del MCCPP con matriz de costos C y conjunto de etiquetas F , se construye una instancia de Weighted Coloring:

- Se utiliza el mismo grafo $G = (V, E)$.
- Se define una función de pesos equivalente donde cada vértice recibe un peso que refleja su estructura de costos.
- La coloración óptima del MCCPP corresponde a una coloración ponderada donde el peso del color i es $w(i) = \max_{v \in S_i} w(v)$ para cada clase de color S_i de la partición.

2. Transformación Weighted Coloring → MCCPP:

Dada una instancia de Weighted Coloring con grafo $G = (V, E)$ y función de pesos $w : V \rightarrow \mathbb{R}^+$, se construye una instancia del MCCPP:

- Se utiliza el mismo grafo $G = (V, E)$.
- Se define un conjunto de etiquetas $F = \{f_1, f_2, \dots, f_n\}$ donde $n = |V|$ (suficientemente grande para cualquier coloración).
- Se construye la matriz de costos C de modo que c_{v, f_i} represente el costo de asignar el color f_i al vértice v , donde este costo está directamente relacionado con el peso del vértice en la clase de color correspondiente. Formalmente, si el vértice v es el de mayor peso en la clase de color i , entonces el costo c_{v, f_i} es igual a $w(v)$.

3. Equivalencia de Soluciones Óptimas:

La solución óptima del MCCPP bajo la anterior transformación coincide exactamente con la solución óptima del Weighted Coloring. Ambos problemas buscan partitionar los vértices de un grafo en conjuntos independientes (coloración propia) minimizando una función de costo que depende de los pesos máximos de los vértices en cada conjunto.

Referencia Académica:

Esta equivalencia está formalmente discutida en el trabajo seminal de Guan y Zhu sobre coloración ponderada, y se retoma en trabajos recientes sobre Weighted Coloring y MCCPP [8], [9], [10].

Conclusión:

Dado que MCCPP y Weighted Coloring son problemáticamente equivalentes, cualquier resultado de complejidad parametrizada (como W-hardness) que se demuestre para uno de ellos se aplica directamente al otro. En particular, la demostración de que Weighted Coloring es W-hard (como se establece en Araújo et al., 2017) implica inmediatamente que MCCPP es también W-hard bajo los mismos parámetros [8].

Independent Set es W-Completo

Antes de proceder a la demostración de **W[1]**-hardness del MCCPP mediante reducción desde Independent Set, es esencial establecer que Independent Set mismo es un problema canónico W-completo en complejidad parametrizada.

Definición del Problema Independent Set

Instancia: Un grafo no dirigido $G = (V, E)$ y un entero positivo k (parámetro).

Pregunta: ¿Existe un conjunto $S \subseteq V$ de vértices con $|S| \geq k$ tal que ningún par de vértices en S es adyacente? Es decir, ¿existe un conjunto independiente de tamaño al menos k ?

Planteamiento: Independent Set es W-Completo

El problema Independent Set parametrizado por el tamaño de la solución k es **W-completo**. Esta es una de las conclusiones fundamentales de la teoría de complejidad parametrizada, establecida desde los trabajos fundacionales de Downey y Fellows y sistematizada en manuales modernos [7], [11].

Implicaciones

La W-completeness de Independent Set significa que dos cosas son ciertas:

1. **Independent Set pertenece a W:** Existe un certificador eficiente (en términos de complejidad parametrizada) que puede verificar en tiempo $f(k) \cdot n^c$ si un conjunto S de tamaño k es un conjunto independiente válido. Esta verificación es trivial: simplemente revisar si existen aristas entre los vértices de S , lo cual toma $O(k^2)$ tiempo.

2. Todo problema en W puede ser reducido a Independent Set mediante una FPT-reducción:

La demostración se basa en mostrar que la definición de W mediante máquinas de Turing ponderadas puede ser capturada por Independent Set. Específicamente, cualquier problema definible como “¿existe una solución de tamaño exactamente k que satisfaga la propiedad $P(x, k)$?” (donde P es verificable en tiempo FPT) puede ser reducido a Independent Set.

Referencia Académica

Estos resultados fundamentales están exhaustivamente cubiertos en [7], [11], donde se expone la jerarquía W y se demuestra rigurosamente que Independent Set parametrizado por k es W-completo.

Demostración de W[1]-Hardness del MCCPP vía Weighted Coloring

Teorema Principal

El Problema de Partición Cromática de Costo Mínimo (MCCPP), cuando se parametriza por la **treewidth** del grafo (o más generalmente, por el tamaño del mayor componente conexo del grafo), es **W-hard**.

Demostración de W[1]-Hardness por Reducción desde Independent Set

La demostración se logra mediante una **FPT-reducción parametrizada** desde el problema canónico W[1]-hard de *Independent Set* hacia Weighted Coloring y, por equivalencia, hacia MCCPP [8].

Definición de Problemas:

- **Independent Set (IS):** Dado un grafo $G = (V, E)$ y un entero k , ¿existe un conjunto $S \subseteq V$ con $|S| \geq k$ tal que ningún par de vértices en S es adyacente?
- **Weighted Coloring (WC):** Dado un grafo ponderado (G, w) con función de pesos $w : V \rightarrow \mathbb{R}^+$, donde el peso de un color S_i se define como $w(i) = \max_{v \in S_i} w(v)$, ¿existe una coloración propia $c = (S_i)_{i=1}^k$ tal que $w(c) = \sum_{i=1}^k w(i) \leq M$ para un umbral M ?

Construcción de la Reducción:

La reducción construye a partir de una instancia (G, k) de Independent Set una instancia ponderada (G', w) de Weighted Coloring, introduciendo gadgets especializados:

1. **Árboles Binomiales Ponderados B_i :** Para cada $i \in [0, 4k + 3]$, se definen recursivamente árboles binomiales con pesos específicos $w_i^0 = 1/2^i + j\epsilon$, donde ϵ es un parámetro pequeño seleccionado cuidadosamente. Estos árboles fuerzan que en cualquier coloración de peso $\leq M$, todos los vértices de un árbol binomial reciban el mismo color de forma determinista.

2. **Árboles Auxiliares** A_i^j : Para cada índice $i \in [0, 4k - 1]$ y $j \in [0, n]$, se construyen árboles auxiliares que contienen copias de árboles binomiales. Su función es actuar como verificadores de que los vértices originales se han seleccionado de forma consistente a lo largo de diferentes copias.
3. **Gadget AND (R_0 -AND)**: Un circuito lógico ponderado que implementa la operación lógica AND entre dos entradas, garantizando que si ambas entradas reciben un color específico R_0 , la salida debe también recibir R_0 .
4. **Árboles de Vértices** T_i^j : Para cada $(i, j) \in [0, k - 1] \times [0, n - 1]$, se construye un árbol que codifica si el vértice $\beta^{-1}(j)$ (donde β es una biyección $V \rightarrow [0, n - 1]$) pertenece o no al conjunto independiente.

Parámetro Umbral M :

Se define $M = k(n - 1)\varepsilon + \sum_{i=0}^{4k+3} \frac{1}{2^i}$ con $0 < \varepsilon < \frac{1}{nk2^{4k+3}}$.

Esta selección garantiza que:

- $M < 2$, proporcionando un umbral bien definido.
- Cada vez que se selecciona un vértice para el conjunto independiente (coloreando su raíz con R_0), se incurre en un costo adicional de $(n - 1)\varepsilon$.
- Solo es posible seleccionar exactamente k vértices sin exceder el presupuesto M .

Lema de Correspondencia:

Sea (T, w) un bosque ponderado que contiene, para cada $(i, j) \in [0, k - 1] \times [0, n - 1]$, la estructura T_i^j como subárbol. Sea c una coloración de (T, w) con $w(c) \leq M$. Entonces existen índices $(j_i)_{i \in [0, k - 1]} \in [0, n - 1]^k$ tales que para cada raíz u de T_j^i :

- Si $j = j_i$ para algún $i \in [0, k - 1]$, entonces $c(u) = R_0$.
- Si $j \neq j_i$ para todo $i \in [0, k - 1]$, entonces $c(u) = R_1$.

Prueba de Equivalencia:

La FPT-reducción satisface: existe un conjunto independiente de tamaño exactamente k en G si y solo si existe una coloración (G', w) con $w(c) \leq M$.

- (**Adelante, \Rightarrow**): Si Z es un conjunto independiente de tamaño k en G , se puede construir una coloración válida asignando cada vértice $v \in Z$ al color R_0 (costo 0 en las estructuras correspondientes) en los árboles T_i^j donde $j = \beta(v)$. Para aristas $\{v_1, v_2\} \in E$ con $v_1 \in Z, v_2 \notin Z$, los gadgets AND se pueden satisfacer con el color R_1 en la salida, manteniendo $w(c) = M$.

- (**Atrás, \Leftarrow**): Si existe una coloración con $w(c) \leq M$, por el Lema anterior, existen índices (j_i) tales que exactamente k vértices reciben color R_0 . Definiendo $Z = \{\beta^{-1}(j_i) : i \in [0, k-1]\}$, se verifica que no hay arista entre elementos de Z . Si existiera una arista $\{v_1, v_2\}$ con $v_1, v_2 \in Z$, entonces los gadgets AND en $H_{\{v_1, v_2\}, i_1, i_2}$ forzarían que su raíz se coloreara con R_0 , lo que contradice el árbol binomial B_{4k} conectado a ella.

Complejidad de la Reducción:

La construcción requiere:

- Tiempo: $f(k) \cdot n^{O(1)}$ donde $f(k) = 2^{O(k)}$ es una función computable de k .
- El parámetro se preserva: $k' = k$ (el parámetro de la nueva instancia depende solo del parámetro de la instancia original).
- El tamaño de cada componente conexo de (G', w) está acotado por $O(k)$.

Conclusión

El Problema de Partición Cromática de Costo Mínimo (MCCPP), cuando se parametriza por la **treewidth** del grafo (o por el tamaño del mayor componente conexo), es **W-hard**. Por lo tanto, a menos que **FPT = W[1]**, **no existe un algoritmo FPT para el MCCPP parametrizado por treewidth**. Este resultado es válido incluso en la clase altamente restringida de bosques (treewidth = 1), lo que hace aún más fuerte la imposibilidad [8].[1]

Corolario sobre la Optimización Observada en Bounded Treewidth:

Guan y Zhu observaron que en grafos de treewidth acotada t , se podía resolver $\sigma(G, w; r)$ en tiempo $n^{O(r)} \cdot r^{O(t)}$ mediante programación dinámica estándar. Sin embargo, este tiempo no es FPT porque la dependencia en n es polinomial de grado potencialmente alto, y más crucialmente, porque el parámetro r (número de colores) está acoplado de forma multiplicativa con el tamaño de la entrada. La demostración de W-hardness implica que no puede haber mejora fundamental en esta complejidad (bajo conjeturas estándar de complejidad parametrizada) [8], [9].[1]

Referencia Académica

Los detalles completos de la reducción y de los gadgets utilizados para demostrar la W-hardness de Weighted Coloring (y, por equivalencia, de MCCPP) en bosques pueden encontrarse en [8].

2.6.4. Búsqueda de Parámetros Alternativos FPT para el MCCPP

Dado que se ha demostrado que el MCCPP es W-hard con respecto a las frecuencias f (equivalente a número de colores r) y treewidth, es natural investigar si existen otros parámetros para

los cuales el problema sea FPT.

Panorama de Parámetros Estudiados:

Los siguientes parámetros han sido analizados en la literatura sobre problemas de coloración ponderada y problemas relacionados:

1. **Vertex Cover:** Este parámetro mide la mínima cantidad de vértices cuya eliminación convierte al grafo en una unión disjunta de conjuntos independientes. Aunque para coloración simple (k -Coloring) existen algoritmos FPT parametrizados por vertex cover, para MCCPP la literatura académica no reporta resultados positivos.
2. **Modular Decomposition Width:** Más restrictivo que treewidth, pero tampoco se ha demostrado FPT para MCCPP.
3. **Clique-Width:** Para coloración estándar se conocen limitaciones, pero MCCPP con costos no ha sido completamente caracterizado.
4. **Pathwidth:** Similar a treewidth pero con separadores lineales en lugar de árboles. Por la reducción desde Independent Set, también es W-hard para MCCPP.[1]
5. **Número de Colores r (Restricción):** Se puede demostrar que MCCPP es W-hard cuando se parametriza por r (el número de colores permitidos) [8].[2]

Resultado de W-Hardness para el Parámetro r :[2]

Siguiendo técnicas de reducción desde Dominating Set y resultados sobre Weighted Coloring, se obtiene:

Planteamiento

El problema de determinar $\sigma(G, w; r)$ (el costo mínimo de una coloración propia usando exactamente r colores) en un grafo ponderado es **W-hard** cuando se parametriza únicamente por r .[2]

Esto implica que incluso si se fija el número de colores a utilizar, el problema sigue siendo paramétricamente intratable (a menos que $W[2] = FPT$, lo que implicaría un colapso de la jerarquía W) [8].

Análisis de Falta de FPT para Parámetros Naturales:

Por la reducción desde Independent Set y por los resultados de W-hardness para Weighted Coloring, los siguientes parámetros naturales **también son W-hard** para MCCPP, debido a que están todos acotados por la talla del mayor componente conexo:[1]

- Treewidth del grafo

- Pathwidth del grafo
- Degeneracy del grafo
- Número cromático $\chi(G)$
- Feedback Vertex Set (FVS)
- Distance to Clique
- Distance to Coloring
- Tree-Depth del grafo
- Clique-Cover Number

Conclusión

Basado en los resultados de Araújo, Baste y Sau, el **Problema de Partición Cromática de Costo Mínimo (MCCPP) no es FPT para ningún parámetro de estructura de grafo estándar** (treewidth, pathwidth, treedecomposición, vertex cover, etc.) [8].

Los únicos parámetros que potencialmente podrían hacer MCCPP FPT serían:

1. **Parámetros combinados muy específicos:** Por ejemplo, una combinación del número de colores r y una cota en el número de aristas o una cota fuerte en k (el tamaño del conjunto independiente máximo).
2. **Parámetros de distancia a clases especiales:** Distancia a grafos completos, distancia a árboles muy pequeños, o similares. Pero estos son ampliamente restrictivos y no capturan muchos grafos de interés práctico.
3. **Parametrización por la solución:** Parametrizar por el valor óptimo $\sigma(G, w)$ mismo, lo cual es una parametrización “circular” y rara vez se considera en análisis de complejidad parametrizada.

3. Diseño de Soluciones Algorítmicas

El MCCPP exige dividir V en conjuntos independientes minimizando un costo asignado a cada clase de color. Su dificultad combinatoria, dado que MCCPP es NP-Hard y APX-Hard, requiere un enfoque múltiple: exacto, aproximado, metaheurístico y estructural.

3.1. Soluciones Exactas (Bases de Referencia)

3.1.1. Algoritmo de Fuerza Bruta: Enumeración Completa

El algoritmo de fuerza bruta es esencialmente un algoritmo de Branch-and-Bound sin poda, cuyo propósito es encontrar la solución óptima Z^* en instancias de tamaño muy pequeño para servir como línea base experimental.

Mecanismo: El algoritmo explora sistemáticamente todas las posibles asignaciones de frecuencias a vértices. Para una instancia $\langle G, F, C \rangle$, con $|V| = n$ y $|F| = k$, se genera cada función de mapeo $\phi : V \rightarrow F$. Para cada asignación, se verifica la restricción de coloración propia, y si es válida, se calcula el costo total $\sum_{v \in V} c_{v,\phi(v)}$.

Análisis de Complejidad: El espacio de búsqueda consiste en k^n posibles asignaciones de color. Dado que cada asignación puede tener un costo distinto (debido a la matriz C), todas las permutaciones de colores deben ser consideradas. Verificar la validez de una coloración toma tiempo $O(|E|)$ o $O(n^2)$. Por lo tanto, la complejidad temporal es intrínsecamente exponencial: $O(k^n \cdot \text{poly}(n))$. Esta dependencia exponencial confirma que el algoritmo solo es viable para valores de n y k extremadamente pequeños.

3.1.2. Resolución mediante Programación Entera Lineal (ILP)

El modelo de Programación Entera Lineal presentado en la Sección 1 se implementa utilizando un solver de ILP para obtener soluciones óptimas en instancias de tamaño moderado. Este enfoque representa una mejora significativa respecto a la fuerza bruta al aprovechar técnicas de poda, relajación y planos de corte.

Mecanismo: El modelo ILP 0-1 se implementa utilizando un solver especializado (como Gurobi, CPLEX o SCIP). El solver aplica algoritmos de Branch-and-Bound con relajación lineal, donde en cada nodo se resuelve la relajación LP del problema para obtener cotas que permitan podar ramas subóptimas. Adicionalmente, se pueden generar planos de corte para fortalecer la formulación.

Ventajas sobre Fuerza Bruta:

- **Poda por Optimalidad:** Las cotas de la relajación LP permiten descartar soluciones que no pueden mejorar la incumbente.
- **Poda por Factibilidad:** Se detectan infactibilidades sin enumerar completamente.
- **Preprocesamiento:** Los solvers aplican reducciones de variables y restricciones para simplificar el problema.

Limitaciones Prácticas: Aunque más eficiente que la enumeración completa, la resolución mediante ILP sigue estando limitada por la NP-hardness del problema. El crecimiento exponencial del árbol de búsqueda en instancias grandes hace necesario el uso de heurísticas o metaheurísticas para casos de tamaño realista.

Implementación: El modelo se codifica en un lenguaje de modelado (como Python+PuLP/Pyomo, AMPL o Julia+JuMP) y se resuelve mediante un solver comercial o de código abierto. La función objetivo y restricciones se implementan directamente según la formalización presentada en la Sección 1.

3.1.3. Programación Dinámica (DP) para Grafos Estructurados

La Programación Dinámica (DP) es una técnica exacta que explota la subestructura óptima y la superposición de subproblemas para convertir (en casos estructurados) problemas que en grafos generales son intratables, en algoritmos polinómicos.

Aplicación y Estructura del Grafo: Árboles

El MCCPP (Problema de Partición Cromática de Costo Mínimo) se vuelve tratable mediante DP en tiempo polinomial para la clase de grafos que tienen estructura de árbol. En un árbol, no hay ciclos y la estructura jerárquica permite un DP en post-orden (rooted tree DP) con estados por vértice y color. Para árboles existe una formulación clásica:

Modelado del Problema en Árboles

Sea $G = (V, E)$ un árbol con $n = |V|$ vértices. Disponemos de un conjunto de colores (frecuencias) F de tamaño k . Para cada vértice $v \in V$ y color $f \in F$ existe un coste no negativo $c_{v,f}$. El objetivo es asignar a cada vértice un color tal que cualquiera par de vértices adyacentes tengan colores distintos (coloración propia) minimizando la suma de los costos.

Estado y Recurrencia del DP en Árboles

Fijamos una raíz arbitraria r del árbol y consideramos la orientación padre-hijo. Para cada vértice v y cada color $c \in \{1, \dots, k\}$ definimos:

$$DP[v][c] = \text{coste mínimo para colorear todo el subárbol con raíz } v \text{ si } v \text{ usa el color } c.$$

Si $\text{children}(v)$ son los hijos de v en la raíz elegida, la recurrencia es:

$$DP[v][c] = c_{v,c} + \sum_{u \in children(v)} \min_{c' \in F, c' \neq c} DP[u][c'].$$

Condición base: si v es hoja, entonces $DP[v][c] = c_{v,c}$ para todo c .

La solución global óptima es $\min_{c \in F} DP[r][c]$.

Construcción de la Solución

Tras calcular todos los $DP[v][c]$ en post-orden, reconstruimos la coloración seleccionando para la raíz un color de coste mínimo y, recursivamente, para cada hijo seleccionamos el color mínimo compatible (diferente del color del padre) según la tabla DP .

Corrección del DP en árboles

Para todo vértice v y color c , $DP[v][c]$ definido por la recurrencia anterior es el coste óptimo de colorear el subárbol de v condicionando a que v tome el color c .

Demostración

Prueba por inducción sobre la altura del subárbol raíz en v .

Base: si v es hoja, la única elección es colorear v con c y el coste es $c_{v,c}$, que coincide con $DP[v][c]$.

Paso inductivo: supongamos cierto para todos los nodos con altura menor que la de v . Cualquier coloración óptima del subárbol de v donde v usa color c induce para cada hijo u una coloración óptima del subárbol de u con la restricción de no usar c en u . Por la hipótesis inductiva, el coste óptimo de cada subárbol u con esa restricción es $\min_{c' \neq c} DP[u][c']$. Sumando y añadiendo $c_{v,c}$ obtenemos precisamente la expresión de la recurrencia.

Análisis de la Complejidad Temporal

Sea n el número de vértices y k el número de colores.

- Preprocesamiento: construir la orientación padre-hijo y el post-orden toma $O(n)$. - Cálculo de DP : para cada vértice v y cada color c se debe sumar sobre los hijos u el término $\min_{c' \neq c} DP[u][c']$. Para un hijo u obtener $\min_{c' \neq c} DP[u][c']$ cuesta $O(k)$ con una implementación directa. Por cada par (v, c) se evalúan esto para todos los hijos de v . Como $\sum_v \deg(v) = O(n)$, la complejidad total es:

$$T(n, k) = O(nk^2).$$

Existe una optimización estándar (mantener para cada hijo el mínimo y segundo mínimo y el correspondiente color), que reduce cada consulta a $O(1)$ y lleva la complejidad a:

$$T(n, k) = O(nk).$$

Conclusión teórica

La DP en árboles ofrece una solución exacta y polinomial para MCCPP cuando el grafo de interferencia es un árbol. La implementación directa tiene coste $O(nk^2)$; con la técnica de mínimo/segundo mínimo por hijo se puede reducir a $O(nk)$.

3.2. Algoritmos de Aproximación

Este enfoque aborda la intratabilidad del MCCPP ofreciendo una garantía de calidad probada ($O(\ln n)$), heredada de su modelado como Weighted Set Cover [4], [5].

3.2.1. Algoritmo de Aproximación con Garantía Teórica ($R \leq O(\ln |V|)$)

Dada la APX-Hardness del MCCPP, no se espera una aproximación de factor constante para grafos generales. En su lugar, el algoritmo de aproximación con la mejor garantía teórica utiliza la conexión del MCCPP con el Problema de Cobertura de Conjuntos Ponderado (Weighted Set Cover, WSC), como ya se analizó en la fase anterior.

Algoritmo de Aproximación WSC-Greedy:

El algoritmo procede de forma voraz, seleccionando repetidamente el conjunto independiente $S^* = \langle I^*, f^* \rangle$ que minimiza el costo marginal efectivo, es decir, el costo por cada nuevo vértice cubierto.

$$\langle I^*, f^* \rangle = \arg \min_{\langle I, f \rangle \in \mathcal{S}} \left\{ \frac{\sum_{v \in I} c_{v,f}}{|\{v \in I : v \text{ está sin colorear}\}|} \right\}$$

Los vértices cubiertos por I^* se colorean con f^* , y se eliminan del universo U . Este proceso se repite hasta que todos los vértices estén coloreados.

Garantía de Rendimiento (Demostración):

El algoritmo Greedy para WSC garantiza que la solución obtenida Z_{Alg} está acotada por el costo óptimo Z_{WSC}^* multiplicado por el número armónico $H(|U|)$, donde $H(|U|) = \sum_{i=1}^{|U|} 1/i$. Dado que el MCCPP se ha modelado como WSC y el número armónico $H(n)$ se approxima por $\ln n$, se tiene la siguiente garantía de rendimiento:

$$R = \frac{Z_{MCCPP}^{Alg}}{Z_{MCCPP}^*} \leq H(|V|) \approx O(\ln |V|)$$

Esta garantía de $R \leq O(\ln |V|)$ es el mejor factor de aproximación conocido para el MCCPP en grafos generales, coherente con los límites inferiores de inaproximabilidad de Set Cover [4], [5].

3.2.2. Algoritmos de Aproximación Estructural ($R = O(\sqrt{|V|})$)

La cota de aproximación logarítmica $O(\ln |V|)$ es el límite riguroso para el MCCPP en grafos generales. Sin embargo, para clases de grafos estructuralmente restringidos, se pueden obtener garantías de aproximación significativamente mejores que explotan las propiedades topológicas del grafo de interferencia $G = (V, E)$. Para grafos bipartitos, de intervalo y unimodulares, la literatura académica establece la existencia de algoritmos que alcanzan un factor de aproximación R acotado por $O(\sqrt{|V|})$ en tiempos de ejecución subcuadráticos, apoyándose en técnicas poliédricas y en el acotamiento de la brecha de integralidad en tales clases [10].

3.3. Heurísticas Greedy Cost-Aware

Para instancias grandes donde las soluciones exactas y las aproximaciones con garantía teórica logarítmica son insuficientes en la práctica, las heurísticas greedy adaptadas al costo juegan un papel esencial. Estas heurísticas se basan en estrategias clásicas de coloración (LF, DSATUR, RLF) enriquecidas con información de costos, y su efectividad práctica ha sido confirmada en estudios recientes de colooreo y MCCPP [12], [13].

3.3.1. Estrategia Largest First (LF) Cost-Aware

Descripción y Mecanismo:

La heurística Largest First (LF) ordena los vértices en orden no creciente de su grado $d(v)$. El algoritmo recorre esta ordenación y asigna a cada vértice v el color disponible de mínimo costo posible, respetando la restricción de coloración propia.

Definición Formal:

Sea $V = \{v_1, v_2, \dots, v_n\}$ tal que $d(v_1) \geq d(v_2) \geq \dots \geq d(v_n)$. Para cada vértice v_i :

- Determinar el conjunto de colores disponibles:

$$F_{\text{avail}}(v_i) = F \setminus \{\phi(u) : u \in N(v_i)\}$$

- Seleccionar el color de costo mínimo:

$$\phi(v_i) = f^*, \quad \text{donde } f^* = \arg \min_{f \in F_{\text{avail}}(v_i)} c_{v_i,f}$$

Análisis Formal de la Complejidad Computacional:

La complejidad de LF se determina por la fase de ordenación inicial y la fase secuencial de coloración. Sean $n = |V|$, $m = |E|$, y $k = |F|$.

1. **Costo de Preordenamiento:** Calcular todos los grados $d(v)$ toma $O(n+m)$. El ordenamiento de los n vértices toma $O(n \log n)$.
2. **Fase de Coloración:** Se itera n veces. En cada iteración, se verifica el conjunto de colores disponibles. Determinar $F_{\text{avail}}(v)$ implica examinar a los vecinos de v , lo que toma $O(d(v))$. La selección del costo mínimo f^* entre las frecuencias disponibles toma $O(|F|)$.

El costo de la coloración es:

$$\sum_{v \in V} O(d(v) + |F|) = O\left(\sum d(v) + n|F|\right) = O(2m + n|F|)$$

La complejidad total está dominada por:

$$T_{LF} = \mathbf{O}(|V| \log |V| + |E| + |V||F|)$$

Demostración Formal de Cota Superior ($|V|^2$):

En el peor caso, donde el grafo es denso ($|E| = O(|V|^2)$) y el número de frecuencias disponibles es grande ($|F| = O(|V|)$), la cota se simplifica:

$$T_{LF} = O(|V| \log |V| + |V|^2 + |V|^2) = \mathbf{O}(|V|^2)$$

Resultados Esperados e Implicaciones (Calidad y Limitaciones):

LF es la heurística más rápida, pero su estrategia estática conlleva una **baja eficiencia cromática**; es decir, a menudo utiliza un número de colores χ_{Alg} significativamente mayor que el número cromático óptimo $\chi(G)$. Esta deficiencia estructural limita la capacidad de la adaptación Cost-Aware para optimizar el costo total. Si los vértices de alto grado que se colorean primero tienen costos uniformemente bajos, el algoritmo podría forzar a los vértices de bajo grado (coloreados al final) a usar frecuencias muy costosas, resultando en una solución final subóptima.

3.3.2. Estrategia DSATUR (Degree of Saturation) Cost-Aware

Descripción y Mecanismo:

DSATUR (Degree of Saturation) es una heurística dinámica que prioriza la selección del vértice **más restringido** en cada paso. El grado de saturación $d_{\text{sat}}(v)$ es el número de colores distintos ya utilizados por los vecinos de v . Se selecciona el vértice no coloreado v con el $d_{\text{sat}}(v)$ máximo. En caso de empate, se desempata eligiendo el vértice con el grado $d(v)$ máximo en el subgrafo de vértices no coloreados. Este criterio dinámico se enfoca en resolver los conflictos más inminentes, buscando minimizar la probabilidad de que se requieran nuevos colores, lo que se traduce en una mayor eficiencia cromática que LF y ha sido base de numerosas variantes modernas [12], [13].

Criterio de Selección:

- Seleccionar el vértice no coloreado v con el $d_{\text{sat}}(v)$ máximo.
- En caso de empate en $d_{\text{sat}}(v)$, desempatar por el grado $d(v)$ máximo en el subgrafo de vértices no coloreados.

Adaptación de Costo:

Al igual que en LF, el color asignado a v es el color disponible de costo mínimo c_{v,f^*} definido por:

$$\phi(v) = f^*, \quad \text{donde } f^* = \arg \min_{f \in F_{\text{avail}}(v)} c_{v,f}.$$

Análisis Formal de la Complejidad Computacional:

La eficiencia de DSATUR depende de la implementación de una estructura de datos (como un *heap* o un árbol balanceado) para gestionar y actualizar los grados de saturación de forma eficiente.

1. **Selección (Extracción Máxima):** La selección del vértice con el máximo d_{sat} se realiza n veces, con un costo de $O(\log n)$ por extracción si se usa un *heap* binario.
2. **Costo Local y Asignación:** Determinar el costo mínimo f^* toma $O(d(v) + |F|)$.
3. **Actualización de Saturación:** Después de colorear v , se deben actualizar los d_{sat} de todos sus vecinos no coloreados $N(v)$. Cada arista se explora dos veces a lo largo del proceso, y cada actualización cuesta $O(\log n)$ si se usa un *heap*.

El costo total de actualización es $\sum_{v \in V} d(v) \cdot O(\log n) = O(|E| \log |V|)$.

La complejidad total es:

$$T_{\text{DSATUR}} = O(|V| \log |V| + |V||F| + |E| \log |V|)$$

Cota Típica:

En implementaciones prácticas se suele citar una complejidad típica $O((n + m) \log n)$ para el comportamiento dominante cuando la gestión de saturaciones domina la ejecución.

Resultados Esperados e Implicaciones (Calidad y Limitaciones):

DSATUR proporciona un **excelente compromiso** entre velocidad y calidad. Su alta eficiencia cromática asegura que el algoritmo opere con un número reducido de frecuencias χ_{Alg} . Esto implica que, en cada paso, la elección del color de costo mínimo f^* se realiza sobre un conjunto $F_{\text{avail}}(v)$ que tiene más probabilidades de contener opciones de bajo costo en las frecuencias ya establecidas, mejorando significativamente la solución de costo total respecto a LF.

3.3.3. Estrategia Recursive Largest First (RLF) Cost-Aware

Descripción y Mecanismo:

A diferencia de LF y DSATUR, RLF construye las clases de color (conjuntos independientes, IS) de forma explícita, una por una, asignando una frecuencia f_{new} a cada IS. RLF construye iterativamente el Conjunto Independiente I_{new} más grande posible dentro del subgrafo restante, basándose en criterios dinámicos de restricción.

Mecanismo Estándar:

En cada paso, RLF identifica un nuevo color f_{new} y construye el conjunto independiente I_{new} más grande posible dentro del subgrafo restante, asignando f_{new} a todos los vértices en I_{new} y eliminándolos del grafo.

Criterio de Construcción del IS Óptimo (Adaptación de Costo):

La adaptación de RLF para el MCCPP implica que la selección de vértices para I_{new} debe estar doblemente sesgada:

- **Estructuralmente:** Debe ser un Conjunto Independiente grande.
- **Por Costo:** Para una frecuencia f_{new} pre-elegida, la selección de vértices dentro de I_{new} debe priorizar aquellos vértices v que tienen el costo más bajo $c_{v,f_{\text{new}}}$.

La heurística RLF modificada busca un compromiso: construir una clase de color que sea no solo grande, sino que también minimice el costo promedio de esa clase, optimizando la asignación de costo para la nueva frecuencia introducida.

Análisis Formal de la Complejidad Computacional:

RLF es la heurística de coloración más costosa, ya que la construcción de cada IS maximal requiere reevaluar dinámicamente las métricas de restricción. Sea k el número de colores utilizados.

1. **Iteraciones:** El algoritmo realiza k iteraciones, donde $k \leq n$.

2. Costo por Iteración: En cada iteración j , se construye I_j . La selección del vértice inicial y la construcción del conjunto independiente involucra la verificación de adyacencias en el subgrafo restante. En general, el costo de construir todas las clases de color I_j está dominado por $\mathbf{O}(|\mathbf{V}| \cdot |\mathbf{E}|)$.

La complejidad total (ignorando el costo de exploración $|V||F|$) es:

$$\mathbf{T}_{\text{RLF}} = \mathbf{O}(|\mathbf{V}| \cdot |\mathbf{E}|)$$

Demostración Formal de Cota Superior ($|V|^3$):

En el caso de grafos densos, donde $|E| = O(|V|^2)$, la complejidad de RLF alcanza su peor caso:

$$T_{\text{RLF}} = O(|V| \cdot |V|^2) = \mathbf{O}(|\mathbf{V}|^3)$$

Resultados Esperados e Implicaciones (Calidad y Limitaciones):

RLF posee la **mejor calidad estructural** entre las heurísticas voraces, utilizando un χ_{Alg} más cercano a $\chi(G)$. La capacidad de optimizar los costos de asignación en grandes conjuntos independientes (clases de color) le confiere el mayor potencial para obtener la mejor solución Z_{Alg} de costo total. Su principal limitación es su complejidad temporal $O(|V||E|)$, lo que la hace inadecuada como algoritmo de producción para instancias muy grandes, pero es valiosa para generar soluciones iniciales de alta calidad para metaheurísticas o como referencia empírica [12], [13].

3.4. Metaheurísticas: Exploración del Espacio de Soluciones

Las metaheurísticas no tienen garantías de rendimiento teórico en tiempo polinomial, pero son esenciales para obtener soluciones de alta calidad empírica en instancias grandes y complejas. En el contexto de MCCPP y problemas de coloreo relacionados, enfoques como Simulated Annealing y las trayectorias con Path Relinking han mostrado resultados competitivos [14], [15], [16].

3.4.1. Simulated Annealing (SA)

Funcionamiento y Base Termodinámica: El Simulated Annealing (SA) es una metaheurística probabilística inspirada en el proceso de recocido metalúrgico. Su función principal es escapar de los óptimos locales mediante la aceptación controlada de movimientos que empeoran la solución. En problemas de coloreo de grafos, incluidas variantes ponderadas, SA ha sido estudiado y aplicado de forma sistemática [14], [15].

Criterio de Aceptación (Metropolis):

El cambio de costo se define como $\Delta C = C_{\text{new}} - C_{\text{current}}$.

- Si $\Delta C \leq 0$ (mejora o mantiene), el movimiento se acepta con probabilidad 1.
- Si $\Delta C > 0$ (empeoramiento), el movimiento se acepta con una probabilidad P que depende de la temperatura T :

$$P = e^{-\Delta C/T}$$

Control de Exploración e Intensificación: La temperatura T se reduce lentamente según un esquema de enfriamiento predefinido. A temperaturas altas ($T \rightarrow \infty$), $P \rightarrow 1$, permitiendo una exploración amplia del espacio de soluciones. A medida que T disminuye (enfriamiento), P se reduce, forzando al algoritmo a concentrarse en la búsqueda de óptimos locales (intensificación). Una curva de enfriamiento suficientemente lenta garantiza la convergencia al óptimo global en tiempo teórico exponencial; en la práctica, se adoptan esquemas heurísticos balanceando calidad y tiempo [14], [15].

3.4.2. Trajectory Search Heuristic + Path Relinking (TSH+PR)

TSH+PR es una metaheurística avanzada que combina la búsqueda local intensa con estrategias de diversificación y recombinación. Este tipo de enfoque ha sido específicamente propuesto para el MCCPP, obteniendo resultados de alta calidad en instancias de tamaño medio y grande [16].

Función Objetivo Penalizada (Manejo de Infactibilidad):

TSH+PR utiliza una función de evaluación que permite la exploración de soluciones infactibles (coloraciones que violan la restricción cromática). Esto es crucial, ya que permite que la búsqueda local pase por regiones “prohibidas” del espacio de soluciones para alcanzar óptimos factibles mejores.

$$f(S) = \sum_{c=1}^n (w_c \cdot |V_c| + M \cdot |E(V_c)|)$$

Donde w_c es el costo asociado al color C_c , $|V_c|$ es el número de vértices con ese color, y $|E(V_c)|$ es el número de aristas conflictivas dentro de esa clase de color V_c . El parámetro M es una penalización suficientemente grande para garantizar que, si existe una solución factible, el proceso de optimización la favorecerá sobre cualquier solución infactible.

Mecanismo Híbrido:

- **Construcción Inicial:** Se utiliza una heurística *greedy* modificada (RLF modificado) para obtener una coloración inicial (posiblemente infactible).

- **Feasibility Search:** Se aplica una búsqueda local en el vecindario de un movimiento crítico (moviendo un vértice conflictivo a una clase no conflictiva) guiada por la minimización de $f(S)$ para alcanzar rápidamente una solución factible (propia).
- **Path Relinking (PR):** PR es una técnica de alto nivel que explora la conectividad entre soluciones de alta calidad (soluciones élite). Genera trayectorias de búsqueda al “recombinar” las características de dos soluciones élite, buscando soluciones intermedias que hereden las mejores propiedades de ambas. Esto aumenta la intensificación y previene la convergencia prematura [16].
- **Diversificación y Reinicios:** Se aplican perturbaciones controladas (movimientos aleatorios) para alejar la solución actual de un óptimo local, forzando la exploración de nuevas regiones del espacio. Una *tabu list* de corta duración evita revertir inmediatamente la perturbación. Los reinicios se realizan si no hay mejora tras un número κ de iteraciones.

3.5. Conclusiones

El Problema de Partición Cromática de Costo Mínimo (MCCPP) se confirma como un problema NP-Hard y, más restrictivamente, APX-Hard. Esta demostración, lograda mediante una L-reducción desde el Minimum Vertex Cover, establece un límite riguroso en la aproximabilidad del problema [1], [2], [3]. La consecuencia fundamental es que el MCCPP no admite un PTAS a menos que $P = NP$, y que su mejor factor de aproximación conocido es $O(\ln |V|)$ vía su modelado como Weighted Set Cover [4], [5].

La estrategia de diseño de soluciones debe ser multifacética:

- **Límites Teóricos:** El algoritmo de aproximación basado en Weighted Set Cover con garantía $O(\ln |V|)$ ofrece la mejor cota de rendimiento probada en tiempo polinomial.
- **Explotación Estructural:** En clases estructuradas como grafos de intervalo, la programación dinámica permite soluciones exactas en tiempo polinomial, mientras que la complejidad parametrizada muestra que no se esperan algoritmos FPT para parámetros estructurales estándar como treewidth o pathwidth [8], [10].
- **Rendimiento Práctico:** Las metaheurísticas (SA y TSH+PR) son cruciales para obtener soluciones de alta calidad en instancias grandes. En particular, las heurísticas de trayectoria con Path Relinking específicamente diseñadas para MCCPP han mostrado un comportamiento muy competitivo [16].
- **Generación de Semillas:** Los algoritmos Greedy Cost-Aware (LF, DSATUR, RLF) proporcionan puntos de partida rápidos y factibles, esenciales para inicializar las metaheurísticas

de búsqueda local y para el análisis empírico, y se apoyan en observaciones recientes sobre heurísticas avanzadas de coloreo de grafos [12], [13].

4. Implementación y Análisis Experimental

4.1. Generación de Instancias

Las instancias de prueba incluyen:

- Grafos Erdős–Rényi $G(n, p)$.
- Matrices de costos aleatorias.
- Casos con soluciones óptimas conocidas.
- Casos límite y grafos estructurados.

4.2. Comparación Empírica

Se evalúa cada algoritmo con base en:

- **Calidad de solución:** ratio $R = Z_{Alg}/Z^*$.
- **Tiempo de ejecución.**
- **Escalabilidad.**
- **Sensibilidad a la densidad y a la distribución de costos.**

4.3. Entregables

El repositorio incluye:

- Implementaciones de todos los algoritmos descritos.
- Generador de instancias.
- Scripts de evaluación y visualización.
- Documentación completa en `README.md`.

Referencias

- [1] C. H. Papadimitriou y M. Yannakakis, «Optimization, Approximation, and Complexity Classes», en *Proceedings of the 20th ACM Symposium on Theory of Computing (STOC)*, ACM, 1988, págs. 229-234. DOI: 10.1145/62212.62231
- [2] S. Arora, C. Lund, R. Motwani, M. Sudan y M. Szegedy, «Proof Verification and the Hardness of Approximation Problems», *Journal of the ACM*, vol. 45, n.º 3, págs. 501-555, 1998. DOI: 10.1145/278298.278306
- [3] I. Dinur y S. Safra, «On the Hardness of Approximating Minimum Vertex Cover», *Annals of Mathematics*, vol. 162, n.º 1, págs. 439-485, 2005. DOI: 10.4007/annals.2005.162.439
- [4] U. Feige, «A Threshold of $\ln n$ for Approximating Set Cover», *Journal of the ACM*, vol. 45, n.º 4, págs. 634-652, 1998. DOI: 10.1145/285055.285059
- [5] Y. Maus y G. Schafer, «A Chronology of Set Cover Inapproximability Results», *ACM SIGACT News*, vol. 52, n.º 4, págs. 58-80, 2021. DOI: 10.1145/3506735.3506745
- [6] J. Flum y M. Grohe, «On the Complexity of Some Colorful Problems Parameterized by Treewidth», en *Theor. Comput. Sci. (special issue on Parameterized Complexity)*, 1, vol. 349, Elsevier, 2006, págs. 9-29. DOI: 10.1016/j.tcs.2005.09.023
- [7] M. Cygan et al., *Parameterized Algorithms* (Texts in Theoretical Computer Science). Springer, 2015. DOI: 10.1007/978-3-319-21275-3
- [8] J. Araújo, J. Baste e I. Sau, «Ruling Out FPT Algorithms for Weighted Coloring on Forests», *Discrete Applied Mathematics*, vol. 283, págs. 381-396, 2020. DOI: 10.1016/j.dam.2019.12.018
- [9] D.-J. Guan y X. Zhu, «A Coloring Problem for Weighted Graphs», *Information Processing Letters*, vol. 61, n.º 2, págs. 77-81, 1997. DOI: 10.1016/S0020-0190(97)00100-X
- [10] J. Baste, I. Sau y D. M. Thilikos, «Dual Parameterization of Weighted Coloring», *Algorithmica*, vol. 82, n.º 10, págs. 2956-2985, 2020. DOI: 10.1007/s00453-020-00686-7
- [11] R. G. Downey y M. R. Fellows, *Fundamentals of Parameterized Complexity* (Texts in Computer Science). Springer, 2013. DOI: 10.1007/978-1-4471-5559-1
- [12] H. Bouziri, S. Nouioua y H. B. Salah, «Graph Coloring: A Novel Heuristic Based on Trailing Path—Properties, Perspective and Applications in Structured Networks», *Soft Computing*, vol. 24, n.º 16, págs. 12437-12452, 2020. DOI: 10.1007/s00500-019-04278-8

- [13] Y. Tian, Q. Zhang y W. Zhang, «A Deep Learning Guided Memetic Framework for Graph Coloring Problems», *Applied Soft Computing*, vol. 122, pág. 108 830, 2022. DOI: 10 . 1016/j.asoc.2022.108830
- [14] H. Peeters, J. Spoerhase y A. Wolff, «Simulated Annealing Algorithm for Graph Coloring», *arXiv preprint*, 2017. arXiv: 1712.00709.
- [15] F. Ricci-Tersenghi et al., «A Theory Explaining the Limits and Performances of Algorithms Based on Simulated Annealing in Solving Sparse Hard Inference Problems», *Physical Review X*, vol. 13, n.º 2, pág. 021 011, 2022. DOI: 10.1103/PhysRevX.13.021011
- [16] J. Araújo, J. Baste e I. Sau, «A Heuristic for the Minimum Cost Chromatic Partition Problem», *RAIRO - Operations Research*, vol. 54, págs. 361-381, 2020. DOI: 10.1051/ro/2019037