Conectando la Universidad de La Habana: Diseño y Análisis de una Red de Fibra Óptica con Restricciones de Grado

Nombre del Estudiante

1. Introducción

La modernización de la infraestructura de red es un componente esencial para el desarrollo académico y científico de las universidades modernas. En este contexto, la Universidad de La Habana (UH), con el apoyo técnico de ETECSA, se propone interconectar sus edificios principales mediante una red de fibra óptica de alta velocidad.

El presente trabajo aborda este problema desde una perspectiva de **Diseño y Análisis de Algoritmos**, modelándolo como un problema de optimización combinatoria sobre grafos, sujeto a restricciones técnicas reales. El objetivo es diseñar una red de costo mínimo que conecte todos los edificios sin ciclos innecesarios y respetando la capacidad limitada de los equipos de red instalados en cada edificio.

2. Descripción Informal del Problema

Se dispone de un conjunto de edificios que deben ser interconectados mediante enlaces de fibra óptica. Cada posible enlace tiene un costo asociado y cada edificio cuenta con un equipo de red con un número máximo de puertos, lo que limita la cantidad de conexiones directas que puede soportar.

El objetivo es seleccionar un conjunto de enlaces que:

- Conecte todos los edificios.
- No contenga ciclos.
- Respete las restricciones de grado impuestas por los equipos de red.
- Minimice el costo total de instalación.

3. Formalización Matemática

3.1. Modelo de Grafo

El problema se modela como un grafo no dirigido y ponderado:

$$G = (V, E, w)$$

donde:

- V es el conjunto de vértices que representan los edificios.
- E es el conjunto de aristas que representan las posibles conexiones.
- $w : E \to \mathbb{R}^+$ es una función de costos.

A cada vértice $v \in V$ se le asocia una cota de grado:

$$b(v) \in \mathbb{N}$$

3.2. Definición del Problema

Encontrar un subconjunto de aristas $T \subseteq E$ tal que:

1. (V, T) sea un árbol generador.
2. $\deg_T(v) \leq b(v)$ para todo $v \in V$.
3. Se minimice $\sum_{e \in T} w(e)$.

Este problema se conoce como **Degree-Constrained Minimum Spanning Tree (DC-MST)**.

4. Análisis de Complejidad Computacional

En esta sección se estudia rigurosamente la complejidad computacional del problema DC-MST, con el objetivo de fundamentar teóricamente las decisiones algorítmicas adoptadas en el resto del trabajo.

4.1. Marco teórico

En esta sección se analiza rigurosamente la complejidad computacional del problema *Degree-Constrained Minimum Spanning Tree* (DC-MST). El objetivo es justificar, desde la teoría de la complejidad computacional, por qué no es razonable esperar algoritmos exactos eficientes para su resolución en el caso general y fundamentar el uso de heurísticas y métodos aproximados.

4.2. Pertenencia del DC-MST a NP

Un problema de decisión pertenece a la clase **P** si existe un algoritmo determinista que lo resuelve en tiempo polinomial con respecto al tamaño de la entrada. Un problema pertenece a la clase **NP** si, dada una solución candidata, esta puede verificarse en tiempo polinomial.

Un problema es **NP-duro** si todo problema de **NP** puede reducirse a él mediante una transformación computable en tiempo polinomial. Si, además, el problema pertenece a **NP**, se dice que es **NP-completo**.

El problema clásico del Árbol Generador Mínimo (MST) pertenece a la clase **P**. Sin embargo, como se mostrará a continuación, la introducción de restricciones sobre el grado máximo de los vértices altera radicalmente su complejidad computacional.

4.3. Problema canónico elegido: Hamiltonian Path

Para demostrar la NP-dureza del problema DC-MST se emplea una reducción polinomial desde el problema canónico **Hamiltonian Path (HP)**.

Definición (Hamiltonian Path). Dado un grafo no dirigido $G = (V, E)$, determinar si existe un camino simple que visite todos los vértices exactamente una vez.

Este problema es conocido por ser **NP-completo**.

La elección de Hamiltonian Path es natural por las siguientes razones estructurales:

- Un camino es un grafo conexo y acíclico, es decir, un árbol.
- En un camino, los vértices internos tienen grado exactamente 2.
- Los vértices extremos del camino tienen grado exactamente 1.

Estas propiedades encajan directamente con la noción de árbol generador con restricciones de grado, particularmente cuando la cota de grado es igual a 2.

4.4. Construcción de la reducción

Sea $G = (V, E)$ una instancia arbitraria del problema Hamiltonian Path. A partir de ella se construye una instancia del problema DC-MST de la siguiente forma:

- Se define $V' = V$.
- Se define $E' = E$.
- Para toda arista $e \in E'$, se asigna un peso unitario: $w(e) = 1$.
- Se define la cota de grado para todos los vértices como:

$$b(v) = 2 \quad \forall v \in V'.$$

La instancia resultante está dada por el grafo ponderado $G' = (V', E', w)$ junto con la función de cotas de grado b.

4.5. Correctitud de la reducción

Se demuestra a continuación que la instancia construida de DC-MST tiene solución si y solo si la instancia original de Hamiltonian Path también la tiene.

Implicación directa ($\Rightarrow$)

Supóngase que existe un camino hamiltoniano en el grafo original G. Por definición, dicho camino:

- Visita todos los vértices exactamente una vez.
- Es conexo y no contiene ciclos.
- Contiene exactamente $|V| - 1$ aristas.

Además, en un camino hamiltoniano cada vértice tiene grado a lo sumo 2. Por tanto, este camino constituye un árbol generador válido para la instancia de DC-MST, respeta las restricciones de grado impuestas y su costo total es:

$$(|V| - 1) \cdot 1 = |V| - 1.$$

En consecuencia, existe una solución válida para la instancia de DC-MST

—

Implicación inversa ($\Leftarrow$)

Supóngase ahora que existe una solución válida para la instancia de DC-MST construida. Dicha solución es un árbol generador T que conecta todos los vértices y satisface:

$$\deg_T(v) \leq 2 \quad \forall v \in V'.$$

Un árbol en el que todos los vértices tienen grado a lo sumo 2 solo puede tener estructura de camino. La aciclicidad impide configuraciones alternativas y la restricción de grado excluye cualquier ramificación. Por tanto, el árbol T define un camino simple que visita todos los vértices exactamente una vez.

En consecuencia, existe un camino hamiltoniano en el grafo original G.

4.6. Complejidad de la transformación

La transformación descrita no añade vértices ni aristas al grafo original. La asignación de pesos y cotas de grado se realiza recorriendo los conjuntos V y E, por lo que su complejidad temporal es:

$$O(|V| + |E|).$$

Por tanto, la reducción es computable en tiempo polinomial.

4.7. Conclusión

Dado que Hamiltonian Path es un problema NP-completo y que existe una reducción polinomial $HP \leq_p DC\text{-}MST$, se concluye que el problema DC-MST es **NP-duro**. Este resultado fundamenta teóricamente el enfoque adoptado en el resto del trabajo, basado en el uso de heurísticas y métodos aproximados.

5. Diseño de Algoritmos

En esta sección se presentan los algoritmos diseñados para resolver el problema *Degree-Constrained Minimum Spanning Tree* (DC-MST). Dado que en la Sección 4 se demostró que el problema es NP-duro, no es razonable esperar la existencia de algoritmos exactos eficientes para el caso general. Por ello, se adopta un enfoque escalonado que combina métodos exactos, heurísticos y de mejora iterativa.

Este enfoque es coherente con los principios del Diseño y Análisis de Algoritmos para problemas de optimización combinatoria complejos.

5.1. Consideraciones generales de diseño

El problema DC-MST generaliza el problema clásico del Árbol Generador Mínimo (MST) mediante la introducción de restricciones de grado en los vértices. Mientras que el MST puede resolverse en tiempo polinomial, la restricción adicional impide el uso directo de algoritmos clásicos como Kruskal o Prim.

Dado este contexto, el diseño algorítmico sigue tres objetivos fundamentales:

- Garantizar soluciones óptimas en instancias pequeñas (algoritmo exacto).
- Obtener soluciones factibles de manera eficiente (heurística voraz).
- Mejorar la calidad de las soluciones sin explorar todo el espacio de búsqueda (búsqueda local).

—

5.2. Algoritmo exacto por enumeración exhaustiva

5.2.1. Idea general

El algoritmo exacto explora exhaustivamente el espacio de soluciones posibles, enumerando todos los árboles generadores del grafo y seleccionando aquel que cumpla las restricciones de grado y tenga el menor costo total. Este enfoque garantiza la optimalidad global de la solución obtenida.

5.2.2. Fundamento teórico

Un árbol generador sobre un conjunto de n vértices contiene exactamente $n-1$ aristas. El número de árboles generadores posibles en un grafo puede crecer de forma exponencial con respecto a n, lo que implica que este enfoque no es viable para instancias grandes. No obstante, resulta útil como referencia teórica y para validar heurísticas.

5.2.3. Pseudocódigo

Algorithm 1 DC-MST Exacto por Fuerza Bruta

Require: Grafo $G = (V, E, w)$, cotas de grado $b(v)$
Ensure: Árbol generador óptimo T^*

$T^* \leftarrow \emptyset$
$costo\_min \leftarrow +\infty$
for all árboles generadores $T \subseteq E$ **do**
    if $\forall v \in V : \deg_T(v) \leq b(v)$ **then**
        $costo \leftarrow \sum_{e \in T} w(e)$
        if $costo < costo\_min$ **then**
            $costo\_min \leftarrow costo$
            $T^* \leftarrow T$
        end if
    end if
end forreturn T^*

—

5.3. Heurística voraz con restricción de grado

5.3.1. Motivación

El algoritmo de Kruskal resuelve eficientemente el problema MST seleccionando aristas de menor costo y evitando ciclos. Sin embargo, este algoritmo no considera límites en el grado de los vértices. La heurística propuesta adapta esta estrategia incorporando la verificación explícita de las restricciones de grado en cada paso.

5.3.2. Estrategia voraz

La heurística selecciona iterativamente la arista de menor peso disponible, siempre que su inclusión no genere un ciclo ni viole las cotas de grado. Aunque esta estrategia no garantiza optimalidad, suele producir soluciones de buena calidad en tiempos razonables.

5.3.3. Pseudocódigo

Algorithm 2 Heurística Voraz para DC-MST

Require: Grafo $G = (V, E, w)$, cotas de grado $b(v)$
Ensure: Árbol generador factible T

```
Ordenar E en orden creciente de w
Inicializar estructura UNION-FIND sobre V
T ← ∅
deg(v) ← 0  ∀v ∈ V
for all e = (u, v) ∈ E en orden creciente do
    if Find(u) ≠ Find(v) then
        if deg(u) < b(u) y deg(v) < b(v) then
            T ← T ∪ {e}
            Union(u, v)
            deg(u) ← deg(u) + 1
            deg(v) ← deg(v) + 1
        end if
    end if
end forreturn T
```

—

5.4. Mejora mediante búsqueda local

5.4.1. Justificación

Las heurísticas voraces pueden quedar atrapadas en soluciones subóptimas debido a decisiones locales tempranas. La búsqueda local permite explorar el entorno de una solución factible, buscando mejoras incrementales sin necesidad de una exploración exhaustiva.

5.4.2. Definición del vecindario

El vecindario de una solución se define mediante intercambios de aristas (*edge swaps*), en los que se elimina una arista del árbol actual y se añade otra externa, preservando la conectividad y las restricciones de grado.

5.4.3. Pseudocódigo

—

Algorithm 3 Búsqueda Local para DC-MST

Require: Solución inicial T, grafo $G = (V, E, w)$, cotas $b(v)$
Ensure: Solución mejorada T

```
mejora ← true
while mejora do
    mejora ← false
    for all e ∈ T do
        for all e' ∈ E \ T do
            T' ← T − {e} + {e'}
            if T' es árbol generador then
                if ∀v : deg_{T'}(v) ≤ b(v) then
                    if costo(T') < costo(T) then
                        T ← T'
                        mejora ← true
                        break
                    end if
                end if
            end if
        end for
        if mejora then
            break
        end if
    end for
end whilereturn T
```

5.5. Metaheurísticas para el problema DC-MST

Dado que el problema Degree-Constrained Minimum Spanning Tree es NP-duro, incluso las heurísticas clásicas pueden quedar atrapadas en óptimos locales cuando se enfrentan a instancias de mayor tamaño o estructuras desfavorables del grafo. Por esta razón, se considera la incorporación de metaheurísticas, las cuales permiten una exploración más profunda y flexible del espacio de soluciones.

Las metaheurísticas se caracterizan por introducir mecanismos de exploración global, como aceptación controlada de soluciones subóptimas o el uso de memoria, manteniendo siempre la factibilidad de las soluciones generadas. En este trabajo se implementaron dos metaheurísticas especialmente adecuadas para problemas de optimización combinatoria sobre grafos: Simulated Annealing y Tabu Search.

5.5.1. Simulated Annealing

Simulated Annealing es una metaheurística inspirada en el proceso físico de recocido térmico. Su principio fundamental consiste en permitir, con cierta probabilidad, la aceptación de soluciones de mayor costo, con el objetivo de escapar de óptimos locales en etapas tempranas de la búsqueda.

El algoritmo mantiene una solución actual y genera soluciones vecinas mediante in-

tercambios de aristas (edge swaps). Si la solución vecina mejora el costo, se acepta directamente. En caso contrario, se acepta con una probabilidad que depende del incremento de costo y de un parámetro denominado temperatura, el cual decrece gradualmente a lo largo de la ejecución.

Esta estrategia permite una exploración amplia del espacio de soluciones al inicio, favoreciendo posteriormente la explotación local conforme disminuye la temperatura.

Algorithm 4 Simulated Annealing para DC-MST

Require: Solución inicial T, grafo $G = (V, E, w)$, cotas $b(v)$
Ensure: Solución mejor encontrada T^*

$T^* \leftarrow T$
Inicializar temperatura T_{emp}
while $T_{emp} > T_{min}$ **do**
    Generar solución vecina T'
    if T' es factible **then**
        if $costo(T') < costo(T)$ **then**
            $T \leftarrow T'$
        else
            Aceptar T' con probabilidad $e^{-\Delta/T_{emp}}$
        end if
    end if
    if $costo(T) < costo(T^*)$ **then**
        $T^* \leftarrow T$
    end if
    Actualizar temperatura
end whilereturn T^*

5.5.2. Tabu Search

Tabu Search es una metaheurística basada en búsqueda local que incorpora memoria explícita para evitar ciclos y exploraciones redundantes. A diferencia de la búsqueda local clásica, Tabu Search permite aceptar soluciones no mejorantes, siempre que el movimiento realizado no esté prohibido por la lista tabu.

En el contexto del DC-MST, los movimientos consisten en intercambios de aristas. Cada movimiento reciente se almacena temporalmente en una lista tabu, impidiendo su reversión inmediata. Este mecanismo fuerza al algoritmo a explorar nuevas regiones del espacio de soluciones, mejorando su capacidad de exploración global.

5.6. Síntesis del enfoque algorítmico

El diseño algorítmico propuesto combina exactitud, eficiencia y calidad de solución. El algoritmo exacto proporciona una referencia teórica, la heurística voraz garantiza escalabilidad y la búsqueda local mejora la calidad de las soluciones obtenidas, logrando un

Algorithm 5 Tabu Search para DC-MST

Require: Solución inicial T, grafo $G = (V, E, w)$, cotas $b(v)$
Ensure: Mejor solución encontrada T^*

```
T* ← T
Inicializar lista tabu
for iteraciones máximas do
    Generar vecindario factible de T
    Seleccionar mejor T' no tabu
    T ← T'
    Actualizar lista tabu
    if costo(T) < costo(T*) then
        T* ← T
    end if
end forreturn T*
```

equilibrio adecuado entre rigor teórico y aplicabilidad práctica.

6. Análisis Experimental

En esta sección se presenta el estudio experimental realizado sobre los algoritmos diseñados para el problema *Degree-Constrained Minimum Spanning Tree* (DC-MST). El objetivo principal es evaluar empíricamente el comportamiento de los algoritmos, contrastando los resultados prácticos con las conclusiones teóricas obtenidas en las secciones anteriores.

Dado que el problema es NP-duro, el análisis experimental resulta fundamental para comprender el compromiso entre optimalidad, eficiencia y escalabilidad de las soluciones propuestas.

—

6.1. Metodología Experimental

Los experimentos se realizaron sobre instancias generadas aleatoriamente mediante un generador parametrizable. Cada instancia se define por los siguientes parámetros:

- Número de vértices $|V|$.
- Probabilidad de existencia de arista (modelo de Erdős–Rényi).
- Rango de pesos de las aristas.
- Cota máxima de grado uniforme $b(v)$.

Para cada configuración experimental se ejecutaron los siguientes algoritmos:

- Algoritmo exacto por fuerza bruta (solo para instancias pequeñas).
- Heurística voraz con restricción de grado.
- Búsqueda local aplicada sobre la solución voraz.
- Simulated Annealing,utilizando como solución inicial el resultado de la heurística voraz
- Tabu Search,aplicada sobre la misma solución inicial, con una lista tabu de tamaño fijo

Cada ejecución mide el tiempo de cómputo y el costo total de la solución obtenida. Los resultados se almacenaron en archivos CSV con el objetivo de facilitar su posterior análisis y reproducibilidad.

—

6.2. Métricas de Evaluación

Las métricas empleadas en el análisis experimental son las siguientes:

- **Costo total**: suma de los pesos de las aristas del árbol generado.
- **Tiempo de ejecución**: tiempo de cómputo medido en segundos.
- **Ratio de aproximación**: cociente entre el costo de una solución heurística y el costo óptimo, cuando este último está disponible.

Estas métricas permiten evaluar tanto la calidad de las soluciones como la eficiencia computacional de cada enfoque.

—

6.3. Resultados Experimentales

Los resultados obtenidos confirman las predicciones teóricas realizadas en el análisis de complejidad:

- El algoritmo exacto presenta un crecimiento exponencial del tiempo de ejecución, lo que restringe su uso a instancias con pocos vértices.
- La heurística voraz escala eficientemente incluso para instancias más grandes, produciendo soluciones factibles en tiempos muy reducidos.
- La búsqueda local mejora sistemáticamente el costo de la solución voraz, acercándose al óptimo cuando este es computable.

- Simulated Annealing logró escapar de óptimos locales generados por la búsqueda local clásica.
- Tabu Search presentó una exploración más estable del espacio de soluciones gracias al uso de memoria.

En instancias donde fue posible calcular la solución óptima, se observó que el ratio de aproximación de la heurística voraz es aceptable, y que la búsqueda local reduce significativamente dicha brecha.Las metaheurísticas introducidas mostraron un comportamiento superior en términos de calidad de solución.Ambos enfoques obtuvieron soluciones de menor costo que las heurísticas tradicionales, a costa de un mayor tiempo de ejecución, aunque este incremento se mantuvo dentro de límites razonables para instancias de tamaño medio.

—

6.4. Discusión

El análisis experimental pone de manifiesto el clásico compromiso entre optimalidad y eficiencia en problemas NP-duros. Mientras que el algoritmo exacto garantiza la mejor solución posible, su costo computacional lo hace impracticable en escenarios reales.

Por otro lado, las heurísticas permiten obtener soluciones de alta calidad en tiempos razonables, lo que las convierte en herramientas adecuadas para aplicaciones prácticas. La búsqueda local demuestra ser una estrategia efectiva para refinar soluciones iniciales sin incurrir en una exploración exhaustiva del espacio de búsqueda.

Estos resultados validan empíricamente la estrategia algorítmica adoptada en este trabajo.

—

6.5. Amenazas a la Validez

Como en todo estudio experimental, existen limitaciones que deben ser consideradas:

- El uso de instancias aleatorias puede no reflejar todos los escenarios reales posibles.
- Las restricciones de grado se consideraron uniformes, lo que simplifica ciertos casos prácticos.
- El análisis se basa en un número finito de instancias y parámetros.

A pesar de estas limitaciones, el diseño experimental es suficiente para extraer conclusiones válidas sobre el comportamiento relativo de los algoritmos evaluados.

—

6.6. Conclusión Experimental

Los experimentos realizados confirman que el problema DC-MST presenta un alto nivel de complejidad computacional, coherente con su clasificación como NP-duro. Al mismo tiempo, demuestran que la combinación de heurísticas voraces y búsqueda local permite obtener soluciones eficientes y de buena calidad en la práctica.

Estos resultados respaldan el enfoque adoptado y justifican el uso de técnicas aproximadas para la resolución de problemas complejos de diseño de redes bajo restricciones reales.

7. Conclusiones

El trabajo evidencia la brecha entre optimalidad teórica y eficiencia práctica, y muestra la utilidad de las heurísticas para problemas NP-completos de relevancia real.