

# Diseño Robusto de Redes Eléctricas con Restricción de Presupuesto ante Fallas de una Sola Planta

Luis Alejandro Arteaga Morales y Mauricio Sunde Jiménez

University of Havana

Havana, Cuba

Emails: luisalejandroarteagamorales@gmail.com, mauriciosundejimenez@gmail.com

**Resumen**—Este artículo formaliza un problema de diseño de redes de transmisión con restricción de presupuesto, motivado por la planificación de distribución eléctrica. Dadas líneas de transmisión candidatas con costos de construcción y capacidades de transporte, sitios de generación con producción acotada y nodos de demanda con consumo requerido, el objetivo es seleccionar un subconjunto de líneas dentro del presupuesto que minimice, en el peor caso, la demanda total no satisfecha ante la falla de exactamente un generador (robustez de contingencia simple).

Palabras clave: diseño de redes, optimización robusta, flujo con capacidad, análisis de contingencias, planificación de redes eléctricas

## I. INTRODUCCIÓN

Las infraestructuras de distribución eléctrica suelen planificarse bajo presupuestos de inversión estrictos y se espera que permanezcan operativas ante contingencias creíbles. El presente trabajo aborda el diseño de una red de transmisión que sea resiliente ante la falla de cualquier planta generadora individual. Los objetivos son: (i) formalizar el problema como un diseño de red con restricción presupuestaria y flujo con capacidad, (ii) analizar la complejidad computacional, (iii) implementar una línea base exacta adecuada para instancias pequeñas, (iv) desarrollar enfoques heurísticos y metaheurísticos escalables, y (v) comparar empíricamente calidad de solución y tiempo de ejecución sobre instancias generadas.

## II. DEFINICIÓN DEL PROBLEMA

Una red eléctrica se representa como un grafo cuyos nodos incluyen sitios de generación y centros de consumo. Se proporciona un conjunto de líneas de transmisión candidatas, cada una con un costo de construcción y una capacidad de transporte. Un diseño consiste en seleccionar qué líneas construir bajo un presupuesto global. Para cualquier diseño, la potencia puede enrutarse a lo largo de las líneas construidas, sujeta a las capacidades de línea y a los límites de producción de los generadores. El criterio de desempeño es la robustez bajo falla de un solo generador: para cada escenario en el que falla exactamente un generador, se evalúa la demanda no satisfecha resultante; el objetivo es minimizar el peor valor de demanda total no satisfecha entre todos esos escenarios.

### A. Instancia del Problema (Entrada del Algoritmo)

Una instancia del problema se define mediante la tupla

$$(G = (V, E), P, C, \{\kappa_e, u_e\}_{e \in E}, \{g_p\}_{p \in P}, \{d_c\}_{c \in C}, B),$$

donde todos los símbolos y parámetros en esta tupla de instancia se definen en la Sección III.

### B. Especificación de la Solución (Salida del Algoritmo)

Se espera que un algoritmo produzca un *diseño* y un *valor de robustez* asociado. El diseño puede representarse como (i) un subconjunto de líneas construidas  $E' \subseteq E$ , o equivalentemente (ii) un vector binario  $x \in \{0, 1\}^{|E|}$  que indica qué líneas se construyen. El valor de robustez es la peor demanda total no satisfecha ante la falla de exactamente una planta.

## III. FORMULACIÓN FORMAL DEL PROBLEMA

### A. Conjuntos y Parámetros

Sea  $G = (V, E)$  un grafo dirigido de líneas de transmisión candidatas.<sup>1</sup> Se definen los siguientes conjuntos:

- $P \subseteq V$ : conjunto de plantas generadoras.
- $C \subseteq V$ : conjunto de centros de consumo, con  $P \cap C = \emptyset$ .
- $S := P$ : conjunto de escenarios de contingencia simple, donde el escenario  $s \in S$  denota la falla de la planta  $s$ .

Para cada planta  $p \in P$ ,  $g_p \geq 0$  denota su capacidad máxima de generación. Para cada consumidor  $c \in C$ ,  $d_c \geq 0$  denota su demanda. Para cada línea  $e \in E$ ,  $\kappa_e \geq 0$  denota su costo de construcción y  $u_e \geq 0$  denota su capacidad de transporte. El presupuesto total disponible es  $B \geq 0$ .

Para un nodo  $v \in V$ , sea  $\delta^+(v)$  el conjunto de arcos salientes y  $\delta^-(v)$  el conjunto de arcos entrantes.

### B. Variables de Decisión

Las decisiones de construcción de líneas son independientes del escenario:

$$x_e \in \{0, 1\} \quad \forall e \in E,$$

donde  $x_e = 1$  indica que la línea  $e$  se construye.

Para cada escenario de contingencia  $s \in S$ , se definen variables operacionales:

$$f_e^{(s)} \geq 0 \quad \forall e \in E \quad r_p^{(s)} \geq 0 \quad \forall p \in P \quad y_c^{(s)} \geq 0 \quad \forall c \in C,$$

donde  $f_e^{(s)}$  es la potencia transmitida en la línea  $e$ ,  $r_p^{(s)}$  es la generación producida por la planta  $p$ , y  $y_c^{(s)}$  es la demanda no satisfecha en el consumidor  $c$  en el escenario  $s$ .

<sup>1</sup>Si se proporciona una red no dirigida, cada línea no dirigida puede modelarse como dos arcos dirigidos opuestos con el mismo costo de construcción y capacidad.

Para linearizar el objetivo de peor caso, se introduce una variable auxiliar  $z \geq 0$ .

### C. Restricciones

#### 1) Presupuesto::

$$\sum_{e \in E} \kappa_e x_e \leq B. \quad (1)$$

#### 2) Capacidad y activación de línea:: Para cada escenario $s \in S$ ,

$$0 \leq f_e^{(s)} \leq u_e x_e \quad \forall e \in E. \quad (2)$$

#### 3) Límites de generación bajo falla de una sola planta:: Para cada escenario $s \in S$ ,

$$r_s^{(s)} = 0, \quad (3)$$

y para todo  $p \in P \setminus \{s\}$ ,

$$0 \leq r_p^{(s)} \leq g_p. \quad (4)$$

#### 4) Conservación de flujo:: Para cada escenario $s \in S$ , se imponen las siguientes restricciones de balance:

- *Plantas* ( $p \in P$ ): el flujo neto de salida es igual a la potencia generada,

$$\sum_{e \in \delta^+(p)} f_e^{(s)} - \sum_{e \in \delta^-(p)} f_e^{(s)} = r_p^{(s)}. \quad (5)$$

- *Consumidores* ( $c \in C$ ): el flujo neto de entrada más la demanda no satisfecha es igual a la demanda,

$$\sum_{e \in \delta^-(c)} f_e^{(s)} - \sum_{e \in \delta^+(c)} f_e^{(s)} + y_c^{(s)} = d_c. \quad (6)$$

- *Nodos de tránsito* ( $v \in V \setminus (P \cup C)$ ): el flujo se conserva,

$$\sum_{e \in \delta^-(v)} f_e^{(s)} - \sum_{e \in \delta^+(v)} f_e^{(s)} = 0. \quad (7)$$

### D. Función Objetivo

El criterio de robustez minimiza la peor demanda total no satisfecha a través de todos los escenarios de falla de una sola planta:

$$\min z \quad (8)$$

sujeto a

$$z \geq \sum_{c \in C} y_c^{(s)} \quad \forall s \in S. \quad (9)$$

## IV. COMPLEJIDAD COMPUTACIONAL

**Definición 1** (Versión de Decisión). *Dada una instancia como en la Sección III y un umbral  $Z \geq 0$ , el problema de decisión pregunta si existe un diseño  $x \in \{0,1\}^{|E|}$  que satisfaga la restricción de presupuesto tal que, para todo  $s \in S$ , existan variables operacionales  $(f^{(s)}, r^{(s)}, y^{(s)})$  que satisfagan las restricciones y  $\max_{s \in S} \sum_{c \in C} y_c^{(s)} \leq Z$ .*

**Teorema 1.** *La versión de decisión del problema de diseño robusto de redes eléctricas con restricción de presupuesto es NP-completa. En consecuencia, el problema de optimización es NP-difícil.*

*Proof.* La pertenencia a NP se cumple porque un certificado consistente en  $x$  y, para cada  $s \in S$ , una tupla  $(f^{(s)}, r^{(s)}, y^{(s)})$  puede verificarse contra las restricciones en tiempo polinómico en  $|V| + |E| + |P| + |C|$ .

La NP-dificultad se demuestra mediante una reducción en tiempo polinómico desde SET COVER, que es NP-completo [1], [2]. Sea  $(\mathcal{U}, \mathcal{S}, k)$  una instancia de SET COVER con  $\mathcal{U} = \{u_1, \dots, u_n\}$  y  $\mathcal{S} = \{S_1, \dots, S_m\} \subseteq 2^{\mathcal{U}}$ . Se construye una instancia de decisión con umbral  $Z = 0$  como sigue. Se crean dos plantas  $P = \{A, B\}$ , un consumidor  $c_i$  para cada elemento  $u_i$ , y un nodo de tránsito  $v_j$  para cada conjunto  $S_j$ . Se fijan las demandas  $d_{c_i} = 1$  para todo  $i$ , se fijan las capacidades de generación  $g_A = g_B = n$ , y se fija  $M = n$ . Se añaden los siguientes arcos:

- para cada  $j$ :  $(A, v_j)$  con  $\kappa = 1$  y  $u = M$ ;
- para cada  $j$ :  $(B, v_j)$  con  $\kappa = 0$  y  $u = M$ ;
- para cada par  $(i, j)$  con  $u_i \in S_j$ :  $(v_j, c_i)$  con  $\kappa = 0$  y  $u = 1$ .

Se fija el presupuesto en  $B = k$ . La construcción es polinómica en  $n + m + \sum_j |S_j|$ .

(Si) Si  $(\mathcal{U}, \mathcal{S}, k)$  es una instancia YES, sea  $J \subseteq \{1, \dots, m\}$  tal que  $|J| \leq k$  y  $\bigcup_{j \in J} S_j = \mathcal{U}$ . Se elige  $x_{(A, v_j)} = 1$  si y solo si  $j \in J$ , y se fija  $x_e = 1$  para todo arco  $e$  con  $\kappa_e = 0$ . Entonces se cumple la restricción de presupuesto. Bajo la falla de  $B$ , para cada  $i$  se selecciona  $j \in J$  con  $u_i \in S_j$  y se envía una unidad de flujo a lo largo de  $A \rightarrow v_j \rightarrow c_i$ . Bajo la falla de  $A$ , para cada  $i$  se selecciona cualquier  $j$  con  $u_i \in S_j$  y se envía una unidad de flujo a lo largo de  $B \rightarrow v_j \rightarrow c_i$ . En ambos escenarios, todas las demandas se satisfacen, por lo que  $\sum_{c \in C} y_c^{(s)} = 0$ .

(Solo si) Si existe un diseño con demanda no satisfecha en el peor caso igual a 0, considérese el escenario donde falla  $B$ . Entonces  $r_B^{(B)} = 0$  y  $y_{c_i}^{(B)} = 0$  para todo  $i$ . Para cada  $i$ , el balance en  $c_i$  implica que entra una unidad de flujo a  $c_i$  por algún arco  $(v_j, c_i)$ ; por tanto,  $u_i \in S_j$ . Como la única planta capaz de suministrar flujo es  $A$ , la restricción de activación fuerza  $x_{(A, v_j)} = 1$ . Por tanto,  $J = \{j : x_{(A, v_j)} = 1\}$  satisface  $\bigcup_{j \in J} S_j = \mathcal{U}$ . Además, la restricción de presupuesto implica  $|J| = \sum_{j=1}^m x_{(A, v_j)} \leq B = k$ , por lo que  $(\mathcal{U}, \mathcal{S}, k)$  es una instancia YES.

Por tanto, el problema de decisión es NP-completo, y se sigue la NP-dificultad del problema de optimización.  $\square$

## V. ALGORITMOS EVALUADOS

Todos los enfoques evaluados producen un conjunto de líneas de transmisión construidas y su valor de robustez. Para un diseño fijo  $E' \subseteq E$ , el desempeño se evalúa de forma independiente para cada escenario de contingencia  $s \in P$  resolviendo una instancia de *flujo máximo*. Una súper-fuente se conecta a cada planta  $p$  con capacidad  $g_p$ , excepto que a la planta fallida  $s$  se le asigna capacidad 0; cada consumidor  $c$  se conecta a una súper-sumidero con capacidad

$d_c$ . Solo se habilitan las líneas en  $E'$ , con capacidades  $u_e$ . Sea  $F^{(s)}(E')$  el valor de flujo máximo bajo el escenario  $s$ , y sea  $U^{(s)}(E') := \sum_{c \in C} d_c - F^{(s)}(E')$  la demanda no satisfecha resultante. El valor de robustez reportado por un algoritmo es  $\max_{s \in P} U^{(s)}(E')$ . Todas las implementaciones de referencia utilizadas en este estudio están disponibles en <https://github.com/Sekai02/daa-project-2025>.

#### A. Línea Base de Fuerza Bruta (Exacto)

La línea base de fuerza bruta enumera cada subconjunto  $E' \subseteq E$  que satisface la restricción de presupuesto  $\sum_{e \in E'} \kappa_e \leq B$ . Para cada subconjunto factible, el valor de robustez se evalúa y se retorna el mejor subconjunto. Este método es exacto pero requiere  $\Theta(2^{|E|})$  evaluaciones de subconjuntos en el peor caso y, por tanto, solo es práctico en instancias pequeñas. Se utiliza como oráculo de corrección para comparar enfoques alternativos.

#### B. Construcciones Voraces

Se implementaron dos heurísticas voraces deterministas. El método *voraz puro* parte del diseño vacío y agrega repetidamente una sola línea que produce la mayor mejora inmediata en la demanda no satisfecha en el peor caso, sujeto al presupuesto; termina cuando ninguna línea restante mejora el objetivo. El método *voraz sensible al costo* sigue la misma idea de construcción incremental, pero prioriza las adiciones utilizando una puntuación mejora-por-costo y aplica un paso de posprocesamiento que elimina líneas cuya eliminación preserva el valor de robustez alcanzado, mejorando la eficiencia presupuestaria.

#### C. GRASP con Búsqueda Local Basada en Intercambios

Un procedimiento GRASP construye múltiples soluciones voraces aleatorizadas utilizando una lista restringida de candidatos prometedores de adiciones de líneas, ordenada por mejora-por-costo. Cada solución construida se refina mediante una búsqueda local que explora movimientos de intercambio (*swap*): eliminar una línea construida y agregar una línea no construida, manteniendo la factibilidad presupuestaria. Se retorna la mejor solución entre todas las iteraciones de GRASP.

#### D. Recocido Simulado

El recocido simulado mantiene un único diseño incumbente y explora su vecindad mediante movimientos de inserción, eliminación e intercambio. Los movimientos que mejoran se aceptan, mientras que los movimientos que empeoran se aceptan con una probabilidad que decrece de acuerdo con un programa de temperatura. La función de energía es lexicográfica: minimización primaria del valor de robustez y, como criterio secundario, minimización del costo total de construcción.

#### E. Búsqueda Tabú

La búsqueda tabú realiza mejora iterativa utilizando movimientos de inserción, eliminación e intercambio. Los movimientos aplicados recientemente se almacenan en una lista tabú durante una tenencia fija para desalentar ciclos; un

criterio de aspiración permite movimientos tabú si mejoran la mejor solución conocida. El método termina tras un número fijo de iteraciones y retorna el mejor diseño encontrado.

#### F. Búsqueda Binaria con Oráculo de Factibilidad Branch-and-Bound (Exacto)

El enfoque exacto seleccionado reformula el objetivo de optimización como un problema de decisión. Para un umbral entero  $Z$ , se define  $\text{FEASIBLE}(Z)$  como verdadero si existe un diseño  $E' \subseteq E$  con  $\sum_{e \in E'} \kappa_e \leq B$  tal que  $\max_{s \in P} U^{(s)}(E') \leq Z$ . El predicado es monótono.

**Lema 1 (Monotonía).** Si  $\text{FEASIBLE}(Z)$  es verdadero, entonces  $\text{FEASIBLE}(Z')$  es verdadero para todo  $Z' \geq Z$ .

*Proof.* Si un diseño alcanza  $\max_{s \in P} U^{(s)}(E') \leq Z$ , entonces el mismo diseño alcanza  $\max_{s \in P} U^{(s)}(E') \leq Z'$  para cualquier  $Z' \geq Z$  por definición.  $\square$

Por monotonía, el valor óptimo de robustez  $z^*$  puede encontrarse mediante búsqueda binaria sobre  $Z \in [0, \sum_{c \in C} d_c]$ . Cada consulta de factibilidad se responde mediante *branch-and-bound*: una búsqueda en profundidad explora decisiones de incluir/excluir para cada línea candidata, poda ramas que exceden el presupuesto y utiliza una cota optimista obtenida al evaluar el diseño en el que se asume que todas las líneas candidatas restantes están disponibles. Si incluso esta finalización optimista no puede lograr demanda no satisfecha  $\leq Z$ , entonces no existe una finalización factible por debajo de  $Z$  y la rama puede podarse.

## VI. METODOLOGÍA EXPERIMENTAL

Los experimentos se diseñaron para evaluar tanto la calidad de la solución como la escalabilidad bajo un presupuesto fijo de tiempo por instancia. Se utilizaron dos suites de instancias: (i) *small-quality*, diseñada para permitir validación contra el oráculo de fuerza bruta, y (ii) *medium-limits*, diseñada para estresar la escalabilidad bajo el mismo límite de tiempo. Cada suite contiene 15 instancias generadas que abarcan múltiples regímenes estructurales (construcciones aleatorias, de cuello de botella y por capas). Cada algoritmo se ejecutó una vez por instancia bajo un *timeout* de 60 segundos. Las métricas registradas fueron el tiempo de ejecución (segundos) y el valor de robustez (demanda no satisfecha en el peor caso). Para las metaheurísticas estocásticas, las implementaciones utilizan semillas pseudorandom fijas y conteos de iteración acotados para asegurar reproducibilidad y para evitar tiempos de ejecución no acotados. Siempre que fuerza bruta completó, su valor objetivo se trató como el óptimo y se utilizó para computar brechas de optimalidad para los demás métodos.

## VII. RESULTADOS Y DISCUSIÓN

La Tabla I reporta la fracción de instancias resueltas dentro del límite de 60 segundos. La línea base de fuerza bruta exhibe *timeouts* frecuentes en familias aleatorias, confirmando que la enumeración exhaustiva no es viable más allá de entradas pequeñas y estructuradas. El método exacto de búsqueda binaria resuelve todas las instancias de *small-quality* y la mayoría

de las de medium-limits; los demás métodos resuelven todas las instancias debido a cronogramas de iteración acotados. Debe señalarse que las tasas de “resuelto” del 100% de las heurísticas reflejan terminación garantizada bajo presupuestos fijos de iteraciones, mientras que los métodos exactos pueden exceder el límite de tiempo porque buscan certificados de optimalidad.

TABLE I  
PORCENTAJE DE INSTANCIAS RESUELTAS DENTRO DEL LÍMITE DE 60 SEGUNDOS (15 INSTANCIAS POR SUITE).

Algoritmo	Small-quality	Medium-limits
Fuerza bruta (exacto)	73.3%	33.3%
Búsqueda binaria + B&B (exacto)	100.0%	80.0%
Recocido simulado	100.0%	100.0%
Búsqueda tabú	100.0%	100.0%
Voraz (sensible al costo)	100.0%	100.0%
Voraz puro	100.0%	100.0%
GRASP + búsqueda local con intercambio	100.0%	100.0%

La calidad de solución se resume en la Tabla II, restringida a instancias donde el óptimo de fuerza bruta está disponible (11 instancias en small-quality y 5 instancias en medium-limits). El método de búsqueda binaria iguala el óptimo de fuerza bruta en todos los casos comparables. La búsqueda tabú y el recocido simulado son competitivos pero exhiben brechas no nulas. Los enfoques voraces y basados en GRASP muestran brechas consistentemente grandes, lo que indica que las heurísticas puramente constructivas y la búsqueda voraz aleatorizada localmente no son suficientes para robustez bajo contingencias en los regímenes evaluados.

TABLE II  
CALIDAD RELATIVA A FUERZA BRUTA EN INSTANCIAS DONDE EL ÓPTIMO DE FUERZA BRUTA ESTÁ DISPONIBLE. LA BRECHA MEDIA SE MIDE EN UNIDADES DE DEMANDA NO SATISFECHA.

Algoritmo	Small-quality		Medium-limits	
	Óptimo	Brecha media	Óptimo	Brecha media
Búsqueda binaria + B&B (exacto)	100.0%	0	100.0%	0
Búsqueda tabú	90.9%	1.45	80.0%	5
Recocido simulado	54.5%	4.45	80.0%	4
Voraz (sensible al costo)	0.0%	12.82	0.0%	23.40
Voraz puro	0.0%	12.82	0.0%	23.40
GRASP + búsqueda local con intercambio	0.0%	12.82	0.0%	23.40

Las Figuras 1, 2 y 3 proporcionan un resumen visual de la tasa de éxito, la brecha de optimalidad y el tiempo de ejecución a través de las familias de instancias.

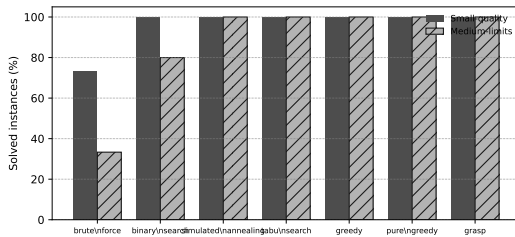


Fig. 1. Tasa de instancias resueltas bajo el límite de 60 segundos.

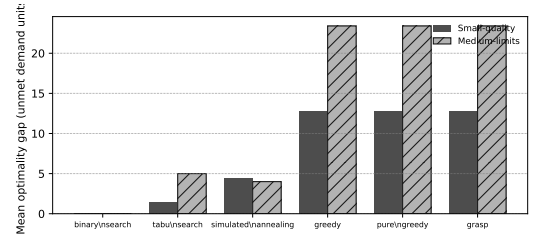


Fig. 2. Brecha media de optimalidad en instancias donde fuerza bruta completó.

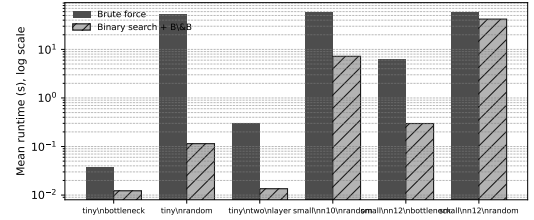


Fig. 3. Tiempo de ejecución medio por familia de instancias (timeouts contados a 60 segundos; escala logarítmica).

#### A. Mejor Algoritmo y Justificación

El método de búsqueda binaria con un oráculo de factibilidad *branch-and-bound* es el mejor algoritmo desarrollado por el equipo de investigación para este problema. Es exacto (cuando termina), iguala el óptimo de fuerza bruta en todos los casos comparables y mejora sustancialmente el tiempo de ejecución respecto a fuerza bruta en las familias estructuradas. Por tanto, el algoritmo de fuerza bruta se conserva únicamente como oráculo de corrección y para instancias muy pequeñas.

Las construcciones voraces y GRASP se descartan como enfoques principales debido a brechas de optimalidad consistentemente grandes en instancias validadas. El recocido simulado y la búsqueda tabú son heurísticas viables y pueden servir como alternativas bajo presupuestos de tiempo estrictos; sin embargo, no proporcionan garantía de optimalidad y exhiben brechas no nulas en instancias validadas.

#### B. Límites Empíricos de Tamaño de Instancia

Bajo el límite de 60 segundos por instancia, fuerza bruta resolvió 11/15 instancias en la suite small-quality y 5/15 instancias en la suite medium-limits. En la suite medium-limits, resolvió todas las instancias de cuello de botella etiquetadas  $n = 12$  pero excedió el límite en todas las instancias aleatorias probadas (etiquetadas  $n = 10$  y  $n = 12$ ). El método de búsqueda binaria resolvió 15/15 instancias en small-quality y 12/15 instancias en medium-limits; sus únicas fallas ocurrieron en instancias aleatorias densas etiquetadas  $n = 12$ . Estos resultados indican que la solvibilidad exacta práctica depende fuertemente de la instancia: las instancias estructuradas permanecen tratables para métodos exactos durante sustancialmente más tiempo que las instancias aleatorias densas.

### *C. Cuándo el Método de Búsqueda Binaria se Desempeña Bien o Mal*

El oráculo *branch-and-bound* se apoya en una cota optimista de finalización que asume que todas las líneas candidatas restantes pueden construirse. Esta cota es informativa para familias estructuradas (instancias de cuello de botella y por capas), donde solo un subconjunto pequeño de líneas puede reducir materialmente la demanda no satisfecha en el peor caso, lo que conduce a una poda fuerte y a una convergencia rápida. En familias aleatorias densas, muchas soluciones parciales parecen prometedoras bajo la relajación optimista, debilitando la poda y causando una explosión exponencial en las ramas exploradas; este comportamiento explica los *timeouts* observados en la familia aleatoria  $n = 12$  de medium-limits.

## VIII. CONCLUSIÓN

Se formalizó un problema de diseño robusto de redes de transmisión con restricción de presupuesto bajo fallas de un solo generador y se demostró NP-difícil. Se implementó un algoritmo exacto de fuerza bruta como línea base (oráculo) y se evaluaron varios algoritmos heurísticos y metaheurísticos. Los resultados experimentales identifican la búsqueda binaria con un oráculo de factibilidad *branch-and-bound* como el enfoque más efectivo: iguala el óptimo de fuerza bruta siempre que el óptimo está disponible y escala sustancialmente mejor en instancias estructuradas.

## REFERENCES

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. Cambridge, MA, USA: MIT Press, 2022.
- [2] J. Kleinberg and É. Tardos, *Algorithm Design*. Boston, MA, USA: Addison-Wesley, 2005.