Robust Budget-Constrained Power Grid Design Under Single-Plant Failures

Luis Alejandro Arteaga Morales and Mauricio Sunde Jiménez
University of Havana
Havana, Cuba
Emails: luisalejandroarteagamorales@gmail.com, mauriciosundejimenez@gmail.com

***Abstract*—This paper formalizes a budget-constrained transmission network design problem motivated by power distribution planning. Given candidate transmission lines with construction costs and transport capacities, generation sites with bounded production, and demand nodes with required consumption, the aim is to select a subset of lines within budget that minimizes the worst-case total unmet demand under the failure of exactly one generator (single-contingency robustness).**

***Index Terms*—network design, robust optimization, capacitated flow, contingency analysis, power grid planning**

I. Introduction

Power distribution infrastructures are commonly planned under stringent investment budgets and are expected to remain operational under credible contingencies. The present work, addresses the design of a transmission network that is resilient to the failure of any single generation plant. The objectives are to (i) formalize the problem as a budget-constrained network design with capacitated flow, (ii) analyze computational complexity, (iii) implement an exact baseline suitable for small instances, (iv) develop scalable heuristic and metaheuristic approaches, and (v) empirically compare solution quality and runtime on generated instances.

II. Problem Definition

A power network is represented as a graph whose nodes include generation sites and consumption centers. A set of candidate transmission lines is given, each with a construction cost and a transport capacity. A design consists of selecting which lines to build under a global budget. For any design, power can be routed along built lines subject to line capacities and generator production limits. The performance criterion is robustness under single-generator failure: for each scenario in which exactly one generator fails, the resulting unmet demand is assessed; the objective is to minimize the worst-case total unmet demand across all such scenarios.

A. Problem Instance (Algorithm Input)

A problem instance is defined by the tuple

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

All symbols and parameters in this instance tuple are defined in Section III.

B. Solution Specification (Algorithm Output)

An algorithm is expected to output a *design* and an associated *robustness value*. The design can be represented as either (i) a subset of built lines $E' \subseteq E$, or equivalently (ii) a binary vector $x \in \{0,1\}^{|E|}$ indicating which lines are constructed. The robustness value is the worst-case total unmet demand under the failure of exactly one plant.

III. Formal Problem Formulation

A. Sets and Parameters

Let $G=(V,E)$ be a directed graph of candidate transmission lines.[1] The following sets are defined:

- $P \subseteq V$: set of generation plants.
- $C \subseteq V$: set of consumption centers, with $P \cap C = \varnothing$.
- $S := P$: set of single-contingency scenarios, where scenario $s \in S$ denotes the failure of plant s.

For each plant $p \in P$, $g_p \geq 0$ denotes its maximum generation capacity. For each consumer $c \in C$, $d_c \geq 0$ denotes its demand. For each line $e \in E$, $\kappa_e \geq 0$ denotes its construction cost and $u_e \geq 0$ denotes its transport capacity. The total available budget is $B \geq 0$.

For a node $v \in V$, let $\delta^+(v)$ be the set of outgoing arcs and $\delta^-(v)$ the set of incoming arcs.

B. Decision Variables

Line construction decisions are scenario-independent:

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

where $x_e = 1$ indicates that line e is built.

For each contingency scenario $s \in S$, operational variables are defined:

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

where $f_e^{(s)}$ is the transmitted power on line e, $r_p^{(s)}$ is the generation output of plant p, and $y_c^{(s)}$ is the unmet demand at consumer c in scenario s.

To linearize the worst-case objective, an auxiliary variable $z \geq 0$ is introduced.

[1] If an undirected network is provided, each undirected line can be modeled as two opposite directed arcs with the same construction cost and capacity.

C. Constraints

1) Budget::

$$\sum_{e\in E} \kappa_e x_e \le B. \tag{1}$$

2) Line capacity and activation:: For each scenario $s \in S$,

$$0 \le f_e^{(s)} \le u_e x_e \qquad \forall e \in E. \tag{2}$$

3) Generation limits under single-plant failure:: For each scenario $s \in S$,

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

and for all $p \in P \setminus \{s\}$,

$$0 \le r_p^{(s)} \le g_p. \tag{4}$$

4) Flow conservation:: For each scenario $s \in S$, the following balance constraints are imposed:

- *Plants* ($p \in P$): net outflow equals generated power,

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

- *Consumers* ($c \in C$): net inflow plus unmet demand equals demand,

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

- *Transit nodes* ($v \in V \setminus (P \cup C)$): flow is conserved,

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

D. Objective Function

The robustness criterion minimizes the worst-case total unmet demand across all single-plant failure scenarios:

$$\min \ z \tag{8}$$

subject to

$$z \ge \sum_{c\in C} y_c^{(s)} \qquad \forall s \in S. \tag{9}$$

IV. Computational Complexity

Definition 1 (Decision Version)**.** *Given an instance as in Section III and a threshold $Z \ge 0$, the decision problem asks whether there exists a design $x \in \{0,1\}^{|E|}$ satisfying the budget constraint such that, for every $s \in S$, there exist operational variables $(f^{(s)}, r^{(s)}, y^{(s)})$ satisfying the constraints and $\max_{s\in S} \sum_{c\in C} y_c^{(s)} \le Z$.*

Theorem 1. *The decision version of the robust budget-constrained power grid design problem is NP-complete. Consequently, the optimization problem is NP-hard.*

Proof. Membership in NP holds because a certificate consisting of x and, for each $s \in S$, a tuple $(f^{(s)}, r^{(s)}, y^{(s)})$ can be verified against the constraints in time polynomial in $|V| + |E| + |P| + |C|$.

NP-hardness is proved by a polynomial-time reduction from Set Cover, which is NP-complete [1], [2]. Let $(\mathcal{U}, \mathcal{S}, k)$ be a Set Cover instance with $\mathcal{U} = \{u_1, \dots, u_n\}$ and $\mathcal{S} = \{S_1, \dots, S_m\} \subseteq 2^{\mathcal{U}}$. Construct a decision instance with threshold $Z = 0$ as follows. Create two plants $P = \{A, B\}$, a consumer c_i for each element u_i, and a transit node v_j for each set S_j. Set demands $d_{c_i} = 1$ for all i, set generation capacities $g_A = g_B = n$, and set $M = n$. Add the following arcs:

- for each j: (A, v_j) with $\kappa = 1$ and $u = M$;
- for each j: (B, v_j) with $\kappa = 0$ and $u = M$;
- for each pair (i, j) with $u_i \in S_j$: (v_j, c_i) with $\kappa = 0$ and $u = 1$.

Set the budget to $B = k$. The construction is polynomial in $n + m + \sum_j |S_j|$.

(*If*) If $(\mathcal{U}, \mathcal{S}, k)$ is a YES-instance, let $J \subseteq \{1, \dots, m\}$ satisfy $|J| \le k$ and $\bigcup_{j\in J} S_j = \mathcal{U}$. Choose $x_{(A,v_j)} = 1$ iff $j \in J$, and set $x_e = 1$ for every arc e with $\kappa_e = 0$. Then the budget constraint holds. Under failure of B, for each i select $j \in J$ with $u_i \in S_j$ and send one unit of flow along $A \to v_j \to c_i$. Under failure of A, for each i select any j with $u_i \in S_j$ and send one unit of flow along $B \to v_j \to c_i$. In both scenarios all demands are satisfied, so $\sum_{c\in C} y_c^{(s)} = 0$.

(*Only if*) If a design exists with worst-case unmet demand 0, consider the scenario where B fails. Then $r_B^{(B)} = 0$ and $y_{c_i}^{(B)} = 0$ for all i. For each i, the balance at c_i implies that one unit of flow enters c_i on some arc (v_j, c_i), hence $u_i \in S_j$. Since the only plant able to supply flow is A, the activation constraint forces $x_{(A,v_j)} = 1$. Therefore, $J = \{j : x_{(A,v_j)} = 1\}$ satisfies $\bigcup_{j\in J} S_j = \mathcal{U}$. Moreover, the budget constraint yields $|J| = \sum_{j=1}^{m} x_{(A,v_j)} \le B = k$, so $(\mathcal{U}, \mathcal{S}, k)$ is a YES-instance.

Therefore, the decision problem is NP-complete, and NP-hardness of the optimization problem follows. ☐

V. Algorithms Evaluated

All evaluated approaches output a set of built transmission lines and its robustness value. For a fixed design $E' \subseteq E$, performance is evaluated independently for each contingency scenario $s \in P$ by solving a max-flow instance. A super-source is connected to each plant p with capacity g_p, except that the failed plant s is assigned capacity 0; each consumer c is connected to a super-sink with capacity d_c. Only lines in E' are enabled, with capacities u_e. Let $F^{(s)}(E')$ denote the maximum flow value under scenario s, and let $U^{(s)}(E') := \sum_{c\in C} d_c - F^{(s)}(E')$ be the resulting unmet demand. The robustness value reported by an algorithm is $\max_{s\in P} U^{(s)}(E')$. All reference implementations used in this study are available at https://github.com/Sekai02/daa-project-2025.

A. Brute Force Baseline (Exact)

The brute force baseline enumerates every subset $E' \subseteq E$ satisfying the budget constraint $\sum_{e\in E'} \kappa_e \le B$. For each feasible subset, the robustness value is evaluated and the best subset is returned. This method is exact but requires $\Theta(2^{|E|})$ subset evaluations in the worst case and is therefore only practical on small instances. It is used as a correctness oracle for comparing alternative approaches.

B. Greedy Constructions

Two deterministic greedy heuristics were implemented. The *pure greedy* method starts from the empty design and repeatedly adds a single line that yields the largest immediate improvement in worst-case unmet demand, subject to the budget; it terminates when no remaining line improves the objective. The *cost-aware greedy* method follows the same incremental construction idea but prioritizes additions using an improvement-to-cost score and applies a post-processing step that removes lines whose deletion preserves the achieved robustness value, improving budget efficiency.

C. GRASP with Swap-Based Local Search

A GRASP procedure constructs multiple randomized greedy solutions using a restricted candidate list of promising line additions ranked by improvement-to-cost. Each constructed solution is then refined by a local search that explores swap moves (remove one built line and add one unbuilt line) while maintaining budget feasibility. The best solution over all GRASP iterations is returned.

D. Simulated Annealing

Simulated annealing maintains a single incumbent design and explores its neighborhood using insertion, deletion, and swap moves. Improving moves are accepted, while worsening moves are accepted with a probability that decreases according to a temperature schedule. The energy function is lexicographic: primary minimization of the robustness value and, as a secondary criterion, minimization of total construction cost.

E. Tabu Search

Tabu search performs iterative improvement using insertion, deletion, and swap moves. Recently applied moves are stored in a tabu list for a fixed tenure to discourage cycling; an aspiration criterion allows tabu moves if they improve upon the best-known solution. The method terminates after a fixed number of iterations and returns the best design encountered.

F. Binary Search with Branch-and-Bound Feasibility Oracle (Exact)

The selected exact approach reformulates the optimization objective as a decision problem. For an integer threshold Z, define $\textsc{Feasible}(Z)$ to be true if there exists a design $E' \subseteq E$ with $\sum_{e \in E'} \kappa_e \leq B$ such that $\max_{s \in P} U^{(s)}(E') \leq Z$. The predicate is monotone.

Lemma 1 (Monotonicity)**.** *If* $\textsc{Feasible}(Z)$ *is true, then* $\textsc{Feasible}(Z')$ *is true for every* $Z' \geq Z$.

Proof. If a design attains $\max_{s \in P} U^{(s)}(E') \leq Z$, then the same design attains $\max_{s \in P} U^{(s)}(E') \leq Z'$ for any $Z' \geq Z$ by definition. □

By monotonicity, the optimal robustness value z^* can be found via binary search over $Z \in [0, \sum_{c \in C} d_c]$. Each feasibility query is answered via branch-and-bound: a depth-first search explores include/exclude decisions for each candidate line, prunes branches that exceed the budget, and uses an optimistic bound obtained by evaluating the design in which all remaining candidate lines are assumed available. If even this optimistic completion cannot achieve unmet demand $\leq Z$, then no feasible completion exists below Z and the branch can be pruned.

VI. Experimental Methodology

Experiments were designed to evaluate both solution quality and scalability under a fixed per-instance time budget. Two instance suites were used: (i) *small-quality*, designed to enable validation against the brute force oracle, and (ii) *medium-limits*, designed to stress scalability under the same time cap. Each suite contains 15 generated instances spanning multiple structural regimes (random, bottleneck, and layered constructions). Each algorithm was executed once per instance under a 60-second timeout. The recorded metrics were runtime (seconds) and robustness value (worst-case unmet demand). For stochastic metaheuristics, the implementations use fixed pseudorandom seeds and bounded iteration counts to ensure reproducibility and to prevent unbounded runtimes. Whenever brute force completed, its objective value was treated as the optimum and used to compute optimality gaps for the other methods.

VII. Results and Discussion

Table I reports the fraction of instances solved within the 60-second limit. The brute force baseline exhibits frequent timeouts on random families, confirming that exhaustive enumeration is not viable beyond small and structured inputs. The binary-search exact method solves all small-quality instances and most medium-limits instances; the remaining methods solve all instances due to bounded iteration schedules. It should be noted that the 100% "solved" rates of the heuristics reflect guaranteed termination under fixed iteration budgets, whereas the exact methods may time out because they search for certificates of optimality.

TABLE I
Percentage of instances solved within the 60-second limit (15 instances per suite).

Algorithm	Small-quality	Medium-limits
Brute force (exact)	73.3%	33.3%
Binary search + B&B (exact)	100.0%	80.0%
Simulated annealing	100.0%	100.0%
Tabu search	100.0%	100.0%
Greedy (cost-aware)	100.0%	100.0%
Pure greedy	100.0%	100.0%
GRASP + swap local search	100.0%	100.0%

Solution quality is summarized in Table II, restricted to instances where the brute force optimum is available (11 instances in small-quality and 5 instances in medium-limits). The

TABLE II
QUALITY RELATIVE TO BRUTE FORCE ON INSTANCES WHERE THE BRUTE FORCE OPTIMUM IS AVAILABLE. MEAN GAP IS MEASURED IN UNITS OF UNMET DEMAND.

Algorithm	**Small-quality**		**Medium-limits**	
	Optimal	**Mean gap**	**Optimal**	**Mean gap**
Binary search + B&B (exact)	100.0%	0	100.0%	0
Tabu search	90.9%	1.45	80.0%	5
Simulated annealing	54.5%	4.45	80.0%	4
Greedy (cost-aware)	0.0%	12.82	0.0%	23.40
Pure greedy	0.0%	12.82	0.0%	23.40
GRASP + swap local search	0.0%	12.82	0.0%	23.40

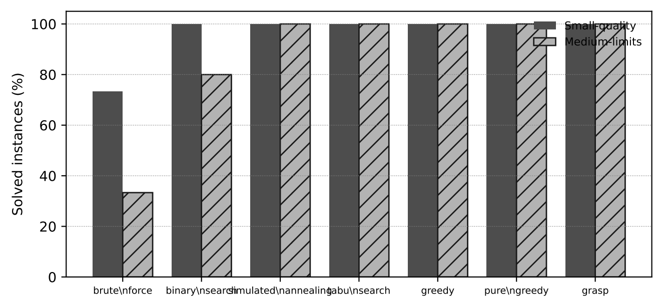

Fig. 1. Solved-instance rate under the 60-second cap.

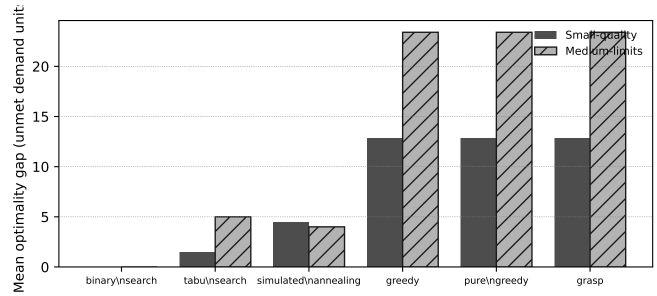

Fig. 2. Mean optimality gap on instances where brute force completed.

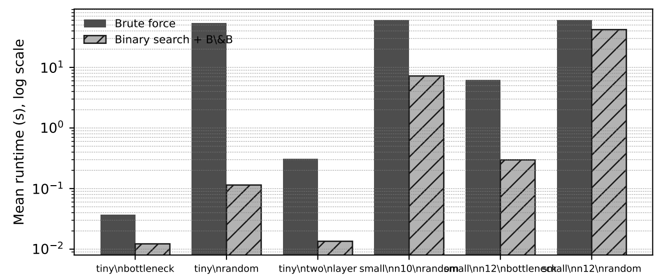

Fig. 3. Mean runtime by instance family (timeouts counted at 60 seconds; log scale).

binary-search method matches the brute force optimum in every comparable case. Tabu search and simulated annealing are competitive but exhibit nonzero gaps. The greedy and GRASP-based approaches show consistently large gaps, indicating that purely constructive heuristics and locally randomized greedy search are insufficient for robustness under contingencies in the tested regimes.

Figures 1, 2, and 3 provide a visual summary of success rate, optimality gap, and runtime across the instance families.

A. Best Algorithm and Rationale

The binary-search method with a branch-and-bound feasibility oracle is the best algorithm developed by the research team for this problem. It is exact (when it terminates), it matches the brute force optimum in all comparable cases, and it substantially improves runtime over brute force on the structured families. The brute force algorithm is therefore retained only as a correctness oracle and for very small instances.

Greedy constructions and GRASP are discarded as primary approaches due to consistently large optimality gaps on validated instances. Simulated annealing and tabu search are viable heuristics and can serve as fallbacks under strict time budgets; however, they provide no optimality guarantee and exhibit nonzero gaps on validated instances.

B. Empirical Instance Limits

Under the 60-second per-instance limit, brute force solved 11/15 instances in the small-quality suite and 5/15 instances in the medium-limits suite. In the medium-limits suite, it solved all bottleneck instances labeled $n = 12$ but timed out on all tested random instances (labeled $n = 10$ and $n = 12$). The binary-search method solved 15/15 instances in small-quality and 12/15 instances in medium-limits; its only failures occurred in dense random instances labeled $n = 12$. These results indicate that practical exact solvability is strongly instance-dependent: structured instances remain tractable for exact methods substantially longer than dense random instances.

C. When the Binary-Search Method Performs Well or Poorly

The branch-and-bound oracle relies on an optimistic completion bound that assumes all remaining candidate lines can be built. This bound is informative for structured families (bottleneck and layered instances) where only a small subset of lines can materially reduce worst-case unmet demand, leading to strong pruning and fast convergence. In dense random families, many partial solutions appear promising under the optimistic relaxation, weakening pruning and causing an exponential blowup in explored branches; this behavior explains the observed timeouts in the medium-limits random $n = 12$ family.

VIII. CONCLUSION

A robust budget-constrained transmission network design problem under single-generator failures was formalized and shown NP-hard. An exact brute force algorithm was implemented as a baseline oracle and several heuristic and metaheuristic algorithms were evaluated. The experimental results identify binary search with a branch-and-bound feasibility oracle as the most effective approach: it matches the brute force optimum whenever the optimum is available and scales substantially better on structured instances.

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.