logistics

MDPI

Article

Optimal-Repair Large Neighborhood Search for the Planning of Extensive Fault-Tolerant Distribution Networks

Renato Bruni [1,*], Alberto Geri [2], Marco Maccioni [2] and Ludovico Nati [2]

1 Dipartimento di Ingegneria Informatica, Automatica e Gestionale, Università di Roma "Sapienza", 00185 Roma, Italy
2 Dipartimento di Ingegneria Astronautica, Elettrica ed Energetica, Università di Roma "Sapienza", 00185 Roma, Italy; alberto.geri@uniroma1.it (A.G.); marco.maccioni@uniroma1.it (M.M.); ludovico.nati@uniroma1.it (L.N.)
* Correspondence: bruni@diag.uniroma1.it

Abstract

Background: Planning power distribution networks is crucial in contemporary infrastructure development. Current distribution paradigms require not only cost minimization but also reliability and fault tolerance. However, designing meshed network topologies is a computationally demanding combinatorial optimization problem, especially for large instances. *Methods*: We reframe this problem as a multi-depot vehicle routing problem in which electrical substations act as depots and power lines represent routes. We develop a four-phase Large Neighborhood Search (LNS) that combines geographically-based destroy operators with a topology-specific MILP repair operator. Each repair subproblem is solved to optimality under the adopted topological, flow conservation, and line capacity constraints. *Results*: Experiments on realistic medium-voltage distribution network instances with up to 1150 nodes show that the proposed method handles cases that are beyond the reach of exact global optimization. Compared with a greedy constructive heuristic, the best LNS solution achieves an average cost reduction of 28.5%. Ablation and sensitivity analyses support the algorithmic design and show stable behavior under reasonable parameter variations. AC power flow analyses on the largest instance confirm electrical consistency under the tested single-branch outage scenarios, with a maximum voltage deviation of 5.1%. *Conclusions*: The proposed optimal-repair LNS provides a scalable approach for planning large fault-tolerant distribution networks under topology and line capacity constraints.

Keywords: network design; vehicle routing; search heuristic; matheuristic

Academic Editor: Robert Handfield

Received: 23 June 2026
Revised: 25 August 2026
Accepted: 1 September 2026
Published: 4 September 2026

1. Introduction

The contemporary energy landscape is undergoing a radical transformation, driven by the increasing integration of renewable energy sources and the digitalization of power systems. In this context, the planning of power distribution networks has evolved from a purely economic exercise into a multi-faceted challenge where infrastructure resilience and reliability are paramount [1]. Historically, Distribution System Planning (DSP) focused primarily on minimizing capital expenditures (CAPEX) and operational costs (OPEX) under static load conditions. However, the contemporary emphasis on "smart grids" and the increasing frequency of extreme weather events have shifted the focus toward ensuring service continuity under contingency [2]. To address these requirements, network architectures have moved away from traditional radial structures towards more robust configurations. Among these, meshed network topologies have gained significant attention for their ability to provide redundant supply paths and guarantee N-1 security, significantly reducing the System Average Interruption Duration Index (SAIDI) [3,4]. When built, these networks are operated radially by keeping appropriate tie-lines open, while the meshed structure provides redundancy that can be exploited through post-fault reconfiguration.

Designing such meshed networks is a large-scale combinatorial optimization problem characterized by nonlinear constraints and a very wide search space. Exact Mixed-Integer Linear Programming (MILP) formulations such as [5] can solve instances up to medium size, but their computational burden becomes prohibitive for large grids. This limitation has motivated the adoption of advanced metaheuristics capable of efficiently exploring large neighborhoods while preserving feasibility with respect to electrical and topological constraints. Among these approaches, Large Neighborhood Search (LNS) has emerged as a powerful framework for solving complex routing and network design problems. Originally introduced by Shaw [6] for vehicle routing problems, LNS iteratively destroys a portion of the current solution and reconstructs it through a repair procedure, allowing for the exploration of distant regions of the solution space with limited algorithmic complexity. Subsequent works have demonstrated the versatility and effectiveness of LNS across a wide range of applications; see e.g., [7–10]. In recent years, several logistics-oriented studies have demonstrated the effectiveness of LNS in complex multi-depot and infrastructure-constrained settings. For instance, ref. [11] applied an adaptive LNS framework to a capacitated vehicle routing problem with parcel lockers, showing that destroy-and-repair strategies are well suited to handling large structured distribution networks. Similarly, ref. [12] studied a capacitated vehicle routing model for distribution and repair activities, highlighting the benefits of combining heuristic exploration with structured subproblem resolution.

In this work, we build upon these methodological ideas and extend them to the domain of power distribution network planning. We reformulate the planning of meshed distribution networks as a Multi-Depot Vehicle Routing Problem (MDVRP) in which substations act as depots, feeders correspond to routes, and demand nodes represent customers. This analogy enables the use of LNS to efficiently explore alternative network topologies, thereby extending destroy-and-repair methodologies to a new class of infrastructure design problems. Unlike adaptive variants of LNS that rely on multiple destroy and repair operators with dynamic selection mechanisms, our method employs a single problem-specific destroy operator and a single optimal-repair operator. The destroy operator removes geographically coherent portions of the network, while the repair operator reconstructs them by solving an exact MILP model on a restricted subgraph. This hybrid design following the matheuristics paradigm [13] makes the repair locally optimal while preserving the meshed structure, aiming at a favorable trade-off between scalability and solution quality. Recent studies have also emphasized the importance of reproducible and multi-metric benchmarking protocols for the fair assessment of metaheuristic algorithms in constrained engineering optimization. In particular, ref. [14] proposed a reproducible multi-metric framework for comparing metaheuristics, while [15] showed that algorithmic rankings may be sensitive to the adopted evaluation protocol. These considerations motivate our study's use of the two complementary dimensions of solution quality and computation time.

The proposed approach can solve very large distribution planning instances that are beyond the reach of exact global optimization methods while enforcing the requirements of the adopted topology–capacity planning model. In particular, the repair model enforces topology, demand coverage, flow conservation, and line capacity constraints, while more detailed operational aspects can be assessed a posteriori. The two solutions obtained for the largest instance are also checked through AC power flow analyses under the considered single-branch outage scenarios, providing additional evidence of their electrical consistency.

https://doi.org/10.3390/logistics10090207

The main contributions of this work can be summarized as follows: (1) An efficient approach for solving large-scale distribution network planning, capable of handling realistic networks that are computationally intractable for exact global optimization methods. (2) A novel methodological reframing of fault-tolerant power distribution network planning as a multi-depot vehicle routing problem, enabling the possible transfer of other advanced routing heuristics to infrastructure network design. (3) A multi-phase LNS heuristic specifically tailored to the design of meshed distribution networks, combining a geographically based phase-dependent destroy operator with a locally exact MILP-based repair operator. (4) The integration of an exact optimization model as a repair mechanism within a heuristic framework, producing a computationally affordable matheuristic in which each restricted repair subproblem is solved to proven optimality.

The rest of this work is organized as follows: Section 2 introduces the problem setting and describes the proposed large neighborhood search framework, including the initialization procedure and the destroy-and-repair strategies; Section 3 presents the computational experiments, including comparison with a greedy heuristic, AC power flow validation, an ablation study, and a sensitivity analysis; Section 4 discusses the main findings and limitations; finally, Section 5 concludes the work and outlines directions for future research.

2. Materials and Methods

2.1. Problem Setting and Mathematical Notation

We consider a distribution network represented by a directed graph $G = (N, A)$, where the set of nodes $N = \{v_1, \ldots, v_{|N|}\}$ is partitioned into a set of primary substations, or *roots* , which inject electrical power into the system:

$$R = \{r_1, \ldots, r_{|R|}\}$$

and a set of *demand nodes*, representing consumers or prosumers (e.g., photovoltaic-equipped buildings) that require electricity and may occasionally generate surplus energy:

$$D = \{d_1, \ldots, d_{|D|}\}.$$

Each arc $(i, j) \in A$ models a candidate electric connection with activation cost c_{ij}. Since distribution networks typically employ a limited set of standardized cable types, we assume, without loss of generality, a single power capacity $\bar{P}$ for all candidate arcs. This assumption is made for notational simplicity only. The use of multiple cable sections can be readily incorporated by associating each arc with alternative capacities and costs, without modifying the structure of the model or the proposed heuristic framework.

A *power line* is a path

$$L = (v_0, v_1, \ldots, v_k),$$

where $v_0, v_k \in R$ and $v_1, \ldots, v_{k-1} \in D$ are distinct demand nodes. Note that the roots v_0 and v_k are not necessarily distinct. Thus, L is a simple path if $v_0 \neq v_k$ or a cycle if $v_0 = v_k$. The set of all power lines in a solution is denoted by

$$\mathcal{L} = \{L_1, \ldots, L_{|\mathcal{L}|}\}.$$

Each demand node $v_i \in D$ must belong to exactly one power line. The presence of a demand node v_i in power line L means that v_i will be supplied by the roots v_0, v_k of L. Note that even if the planning algorithm constructs power lines, at operation time each power line will be split into two radial *feeders* by opening one appropriate arc, ensuring full compliance with the radiality requirements of a distribution system [16,17]. This operational reconfiguration is performed after the planning stage, and as such is beyond the scope of this work. Each power line must satisfy the capacity limits and topological requirements described below. Voltage constraints are not included in the repair MILP, and can be assessed a posteriori through AC power flow validation.

2.2. Target Topologies and Their Vehicle Routing Reframing

Two admissible meshed topologies are considered, called *Loop-Feeder* and *Open-Loop* . In loop-feeder topology, a power line L may contain, at the extremes, either two different roots or the same root:

$$L = (r_a, \ldots, r_b), \quad r_a \neq r_b$$

or

$$L = (r_a, \ldots, r_a).$$

In open-loop topology, each power line must connect two *distinct* roots [18]; hence, each of its demand nodes can be supplied by each of the two roots:

$$L = (r_a, \ldots, r_b), \qquad r_a \neq r_b.$$

In both topologies, every demand node is reachable through two alternative paths, so that if one path fails, supply can be restored through the other. However, only the open-loop topology guarantees that these two paths will originate from two distinct roots:

$$\forall d \in D,\ \exists r_a, r_b \in R,\ r_a \neq r_b :\ d \text{ is reachable from both.} \tag{1}$$

As a consequence, open-loop networks ensure supply continuity not only under line outages but also in the event of a root (substation) failure, whereas loop-feeder networks provide redundancy against arc failures only.

This structure naturally aligns with a Multi-Depot Vehicle Routing Problem (MDVRP). Each root $r \in R$ plays the role of a depot, each demand node $d \in D$ the role of a customer, and each power line $L \in \mathcal{L}$ corresponds to a route that serves a subset of customers in D. The specific MDVRP variant depends on the target network topology. In the open-loop topology, each route must start at a root $r_a \in R$ and terminate at a different root $r_b \in R$, with $r_a \neq r_b$. This induces a routing structure with distinct origin and destination depots, which is a multi-depot extension of the open vehicle routing problem, where routes are not required to return to their origin depot; see also [19,20]. In the loop-feeder topology, a route may either start and end at the same root or connect two distinct roots. The former situation corresponds to the classical MDVRP, where routes originate from a given depot and return to it after serving their assigned customers. The latter case corresponds to a routing structure with distinct origin and destination depots, as already explained above. In both cases, each demand node must be visited by exactly one route. To give an example, Figure 1 shows two small illustrative networks with 50 nodes, one for loop-feeder topology (left) and one for open-loop topology (right). Red squares are roots, while blue dots are demand nodes. As can be seen, under the loop-feeder topology a power line may either connect two distinct roots or return to the same root (e.g., (1,19,38,44,42,48,36,40,39,1)), whereas under the open-loop topology a power line can only connect two distinct roots.

Within this routing interpretation, some electrical feasibility constraints naturally translate into capacity constraints on vehicle routes. In particular, the thermal rating of distribution lines induces a maximum transferable power, which directly corresponds to the vehicle capacity in the MDVRP. Formally, each customer (demand node) $d \in D$ is associated with an electrical demand p_d, and the cumulative demand served along a route must not exceed the power capacity of the corresponding feeder. Additional operational constraints such as detailed voltage limits or protection requirements could further restrict the admissible length and composition of routes. These aspects are not included as constraints in the repair MILP considered here, where the optimization model focuses on topology, demand coverage, flow conservation, and line capacity limits.

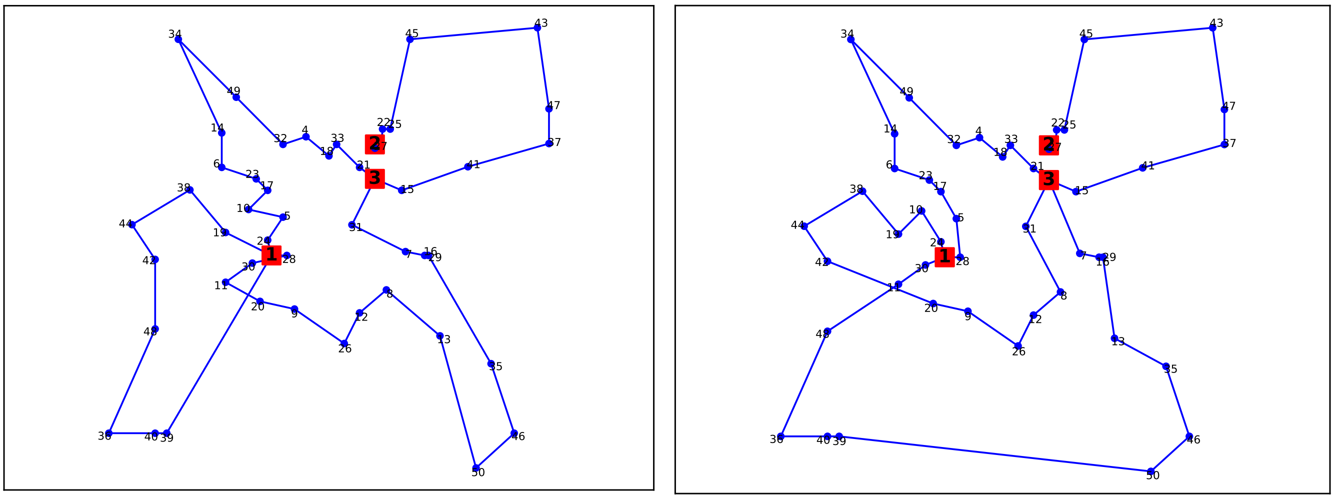

Figure 1. Sample network configurations of loop-feeder (**left**) and open-loop (**right**) topologies for a small instance with 50 nodes. Red squares are roots, while blue dots are demand nodes.

Casting the problem in this form enables the application of Large Neighborhood Search (LNS), where destroy operations modify subsets of routes and repair operations reconstruct feasible routes consistent with the prescribed depot–route incidence structure. We stress that the MDVRP reformulation is used here to guide the construction of effective destroy-and-repair neighborhoods. It does not replace the electrical feasibility model. Electrical feasibility is enforced during the repair step by the topology-specific MILP, which is a restricted version of the planning formulations introduced in [5] and includes flow conservation and capacity constraints on the induced subgraph associated with the destroyed portion of the current solution. Thus, the proposed technique does not optimize a purely topological surrogate of the planning problem; the routing interpretation guides the search, while the repair MILP enforces the relevant topological and capacity constraints.

2.3. Overview of the Proposed Algorithm

The proposed solution approach is based on an LNS framework tailored to the planning of fault-tolerant distribution networks. In line with classical LNS schemes for vehicle routing problems, the algorithm iteratively improves a current solution by alternating *destroy steps* and *repair steps*. At each iteration, a solution is represented by a set of routes $\mathcal{L} = \{L_1, \ldots, L_{|\mathcal{L}|}\}$, where each route corresponds to a power line connecting one or two roots and serving a subset of demand nodes. The destroy step removes a portion of the current routes according to an adaptive multi-phase strategy. The repair step reconstructs feasible routes by solving a restricted MILP optimization problem.

The search is organized into four phases. In **Phase 1**, the algorithm builds an initial feasible solution by assigning one power line to each demand node. Thus, the initial solution is highly fragmented, with each power line serving exactly one customer.

In **Phase 2**, the algorithm aims at reducing this fragmentation by grouping geographically close power lines. The clustering algorithm DBSCAN [21] is applied to the centroids of the current power lines, yielding a family of clusters. A complete iteration of this phase consists of processing all clusters produced by DBSCAN, one cluster at a time, through destroy-and-repair moves. If the average saturation level is still below the threshold τ_1 after one complete iteration, DBSCAN is applied again to the updated solution and a new complete iteration of Phase 2 is performed. This phase promotes the creation of power lines serving multiple customers.

Once the average saturation level of the power lines exceeds a prescribed threshold τ_1, the algorithm enters **Phase 3**. In this phase, a matching based on the centroids of all current power lines is computed, yielding pairs of power lines to be destroyed and reconstructed together. A complete iteration of this phase consists of computing the matching and then processing all resulting pairs, one pair at a time.

When the improvement delivered by a complete Phase 3 iteration falls below a threshold τ_2, the algorithm switches to **Phase 4**. In this final phase, since pairwise optimization appears no longer able to obtain meaningful improvements, the search explores larger neighborhoods. The current power lines are partitioned into groups of average cardinality three or four through a k-means clustering procedure [22]. A complete iteration of this phase consists of computing such a partition and then destroying and reconstructing all clusters one at a time. Overall, the four-phase structure guides the search from a highly fragmented initial solution towards progressively more structured and larger-scale local re-optimization moves. The complete LNS multi-phase matheuristic will be described in more detail in the next sections, and is also summarized in Algorithm 1.

2.4. Phase 1: Initialization

The algorithm starts from an initial greedy solution. For each demand node $d \in D$, we construct a trivial power line

$$L_d = \big(r_1(d), d, r_2(d)\big), \tag{2}$$

where $r_1(d), r_2(d) \in R$ are two distinct roots associated with demand node d, with $r_1(d) \neq r_2(d)$. In the simplest implementation, they can be chosen as the two roots that are nearest to d in electrical distance. Thus, each initial power line contains exactly one demand node in the middle and connects two distinct roots. The initial solution is

$$\mathcal{L}^{(0)} = \{L_d : d \in D\}. \tag{3}$$

This initialization produces a highly fragmented solution which is nevertheless already feasible for both the loop-feeder and open-loop topologies. In the vehicle routing interpretation, it corresponds to assigning one route to each customer. The purpose of the subsequent phases is to progressively reduce this fragmentation and improve the objective value by repeatedly destroying and reconstructing suitable subsets of power lines.

2.5. Destroy Operator

The destroy operator aims to remove a structured subset of routes from the current solution in order to define a subproblem that is suitable for effective re-optimization. As commonly done in LNS heuristics for vehicle routing problems, destruction is performed at the route level rather than on individual arcs. In analogy with LNS heuristics for vehicle routing problems, the destruction strategy evolves during the search according to the structure of the incumbent solution. Since the initial solution is highly fragmented, we first pursue the concatenation of geometrically close demand nodes. When the power lines become sufficiently saturated, the search moves to a pairwise re-optimization phase. Finally, when pairwise optimization ceases to provide meaningful improvements, the search enlarges the neighborhood by jointly re-optimizing larger groups of power lines.

A key principle of this operator is to remove groups of routes that are likely to interact strongly so that their joint re-optimization may yield significant improvements. In particular, for each route $L \in \mathcal{L}$, a geometric centroid $c(L)$ is computed from the coordinates of its customers. Routes with nearby centroids are assumed to serve the same area, making them good candidates for joint re-optimization. The distance-based criterion is not intended to approximate electrical coupling, impedance relations, or flow interactions; it is only a technique to identify spatially close power lines that may share low-cost alternative connections, and as such can be profitably re-optimized together. After the destroy set is selected, the topology, load allocation, flows, and capacity feasibility are determined by the topology-specific MILP repair model.

2.5.1. Phase 2: Concatenation of Geometrically Close Power Lines

At the early stage of the search, the solution is typically characterized by a large number of short and weakly loaded power lines. This situation corresponds to a highly fragmented routing solution in which routes are far from being saturated with respect to their capacity. Let

$$\rho(L) = \frac{\sum\limits_{d \in L} p_d}{\bar{P}} \tag{4}$$

denote the load ratio of a power line L, where p_d is the electrical demand of customer d and $\bar{P}$ is the capacity of the feeder. Phase 2 is applied as long as the average saturation level

$$\bar{\rho} = \frac{1}{|\mathcal{L}|} \sum_{L \in \mathcal{L}} \rho(L) \tag{5}$$

remains below the predefined threshold τ_1. The goal here is to reduce the number of power lines by merging those that are spatially close and electrically compatible, allowing the subsequent repair step to construct longer and better loaded power lines. Let $c(L)$ denote the geometric centroid of power line L. We apply DBSCAN to the set $\{c(L) : L \in \mathcal{L}\}$, obtaining clusters

$$\mathcal{C} = \{C_1, \ldots, C_m\}. \tag{6}$$

A complete iteration of Phase 2 consists of processing all these clusters, one at a time; at the end of the complete iteration, the average saturation level is computed. If it is below τ_1, DBSCAN is applied again and another complete Phase 2 iteration is performed. Let C_k be a generic cluster and let μ be a user-defined upper bound on the number of routes destroyed at once. We define the set $\mathcal{S}_k \subseteq C_k$ of power lines to be destroyed as

$$\mathcal{S}_k = \begin{cases} C_k & \text{if } |C_k| \le \mu, \\ \{L^\star\} \cup \mathcal{N}_{\mu-1}(L^\star) & \text{otherwise,} \end{cases} \tag{7}$$

where

$$L^\star \in \arg\min_{L \in C_k} \rho(L) \tag{8}$$

is the least saturated power line in the cluster and $\mathcal{N}_{\mu-1}(L^\star)$ denotes the set of the $\mu - 1$ power lines in $C_k \setminus \{L^\star\}$ with smallest centroid distance to $L^\star$. The destroy step removes all arcs belonging to the selected power lines in $\mathcal{S}_k$. The set of demand nodes freed by this removal is

$$D_k^{\text{free}} = \bigcup_{L \in \mathcal{S}_k} \big(V(L) \cap D\big), \tag{9}$$

where $V(L)$ denotes the set of nodes visited by power line L. In the subsequent repair step, these are merged (concatenation) by solving a topology-specific MILP on the subgraph induced by the freed demand nodes and all the roots:

$$G_k = (N_k, A_k), \qquad N_k = R \cup D_k^{\text{free}}, \qquad A_k = \{(i,j) \in A : \ i \in N_k, \ j \in N_k\}. \tag{10}$$

Since the repair MILP is formulated according to the target topology, the repair step reconstructs the partially destroyed solution so as to ensure topological feasibility and preserve the required loop-feeder or open-loop structure throughout the search. The initial solution is already compatible with both topologies, since each initial power line connects two distinct roots. During the subsequent destroy-and-repair phases, topological feasibility is preserved by the corresponding topology-specific repair MILP.

2.5.2. Phase 3: Destruction of Power Line Pairs via Matching

As the power lines become more saturated, further improvements are less likely to come from simple spatial aggregation. When the average saturation level exceeds the threshold τ_1, the algorithm moves to Phase 3. During this phase, routes are paired by minimizing the distance between their centroids, with one pair at a time selected for destruction. Removing pairs of routes limits the size of the induced subproblem while still allowing nontrivial rearrangements of customer assignments. Let $\mathcal{L}$ be the current set of power lines. We compute a matching M by pairing power lines with minimal centroid distance:

$$M = \{(L_a^1, L_b^1), \dots, (L_a^q, L_b^q)\}. \tag{11}$$

A complete iteration of Phase 3 consists of computing the matching M and then processing all its pairs, one pair at a time. For each pair (L_a, L_b), the destroy step removes all arcs in $L_a \cup L_b$, and the corresponding freed demand nodes are

$$D_{ab}^{\text{free}} = \big(V(L_a) \cap D\big) \cup \big(V(L_b) \cap D\big). \tag{12}$$

The subsequent repair step solves a topology-specific MILP on the following induced subgraph G_{ab}, producing feasible power lines that reconnect the freed demand nodes according to the target topology:

$$G_{ab} = (N_{ab}, A_{ab}), \qquad N_{ab} = R \cup D_{ab}^{\text{free}}, \qquad A_{ab} = \{(i,j) \in A : \; i \in N_{ab}, \; j \in N_{ab}\}. \tag{13}$$

At the end of each complete iteration, the objective improvement produced by the whole sequence of pairwise destroy-and-repair moves is evaluated. If this improvement is still significant, a new matching is computed on the updated solution and another complete Phase 3 iteration is performed; otherwise, if the improvement falls below a threshold τ_2, the algorithm switches to Phase 4.

2.5.3. Phase 4: Destruction of Larger Groups via K-Means Clustering.

When pairwise re-optimization is no longer sufficient to obtain meaningful improvements, the algorithm explores larger neighborhoods by jointly destroying and reconstructing larger groups of power lines. In this phase, the current power lines are partitioned by means of a K-means clustering procedure. Let

$$K \in \left\{ \left\lceil \frac{|\mathcal{L}|}{3} \right\rceil, \left\lceil \frac{|\mathcal{L}|}{4} \right\rceil \right\} \tag{14}$$

denote the number of clusters used in the partition, selected so as to obtain groups of average cardinality close to three or four. The resulting partition is denoted by

$$\mathcal{P} = \{G_1, \dots, G_s\}, \tag{15}$$

where each $G_t \subseteq \mathcal{L}$ is a cluster of power lines. A complete iteration of Phase 4 consists of computing such a partition and then processing all clusters in $\mathcal{P}$, one cluster at a time. For a generic group G_t, the destroy step removes all arcs belonging to the power lines in G_t. The corresponding freed demand nodes are

$$D_t^{\text{free}} = \bigcup_{L \in G_t} \big(V(L) \cap D \big). \tag{16}$$

The repair step then solves the topology-specific MILP on the induced subgraph

$$G_t = (N_t, A_t), \qquad N_t = R \cup D_t^{\text{free}}, \qquad A_t = \{(i,j) \in A : i \in N_t,\ j \in N_t\}. \tag{17}$$

By jointly re-optimizing larger groups of power lines, Phase 4 explores neighborhoods that cannot be reached through pairwise moves only, thereby providing a stronger form of intensification in the final stage of the search; however, this comes at the price of solving larger repair subproblems.

2.6. Repair Step: Exact MILP Solved on the Induced Subgraph

After each destroy step, a *repair step* reconstructs the removed part of the solution. This is achieved by solving the following mixed-integer linear programming model, proposed in [5], on the generic induced subgraph $G' = (N', A')$ associated with the current destroy move, where G' denotes G_k, G_{ab}, or G_t depending on the active search phase, as defined above. For notational simplicity, N and D in the repair model denote N' and $D \cap N'$, respectively. Therefore, the same repair mechanism is used throughout all search phases, whereas the destroy operator determines the size and structure of the currently explored neighborhood. MILP-based repair constitutes a matheuristic mechanism often used in routing contexts [13,23]. In other words, we locally rebuild a feasible set of power lines while enforcing the electrical and topological constraints required by loop-feeder or open-loop topologies.

Binary variables x_{ij} indicate whether a candidate arc $(i,j) \in A'$ is activated. Continuous variables f_{ij} represent the electrical power flow on arc (i,j), where a positive (resp. negative) value denotes flow from i to j (resp. from j to i). Each demand node $d \in D$ is associated with a fixed electrical demand p_d. Here, the set of arcs incident to node d within G' is denoted as $\delta(d)$, its outgoing and incoming arcs in A' are denoted by $\delta^+(d)$ and $\delta^-(d)$, respectively, and $A(S)$ is the set of arcs with both endpoints in S. The objective function minimizes the total installation cost:

$$\min \sum_{(i,j) \in A'} c_{ij}\, x_{ij}. \tag{18}$$

Topological constraints require that each demand node must be incident to exactly two active arcs, enforcing a meshed (loop-feeder or open-loop) structure:

$$\sum_{(i,j) \in \delta(d)} x_{ij} = 2 \qquad \forall d \in D. \tag{19}$$

Hereafter, for notational simplicity, D and N will denote $D \cap N'$ and N', meaning that the constraints apply only to the portion of the graph involved in the current destroy-and-repair move. To prevent the formation of isolated subtours, subtour elimination constraints are imposed. In the loop-feeder case, these constraints are enforced on subsets of demand nodes:

$$\sum_{(i,j) \in A(S)} x_{ij} \le |S| - 1 \qquad \forall S \subseteq D,\ |S| \ge 2. \tag{20}$$

In the open-loop case, stronger subtour elimination constraints are required. For every subset of nodes containing at most one root, the number of active arcs with both endpoints in the subset must be strictly less than the size of the subset:

$$\sum_{(i,j)\in A(S)} x_{ij} \le |S| - 1 \qquad \forall S \subseteq N,\ |S| \ge 2,\ |S \cap R| \le 1. \tag{21}$$

These constraints ensure that no power line can close on the same root and that each demand node is connected to two distinct roots through two independent paths.

Electrical flow conservation constraints require that at each demand node, the total incoming power must satisfy the local demand:

$$\sum_{(i,j)\in\delta^-(d)} f_{ij} = p_d + \sum_{(i,j)\in\delta^+(d)} f_{ij} \qquad \forall d \in D. \tag{22}$$

Power limits of the distribution lines are modeled by bounding the magnitude of the flow on each active arc:

$$-\frac{p_{ij}}{2} x_{ij} \le f_{ij} \le \frac{p_{ij}}{2} x_{ij} \qquad \forall (i,j) \in A' \tag{23}$$

where, under the standardized-cable assumption, we set $p_{ij} = \bar{P}$ for all candidate arcs $(i,j) \in A'$. The factor $1/2$ in constraint (23) is used as a conservative reliability margin, consistent with the fault-tolerant planning model introduced in [5]. Since each power line is operated by splitting it into two radial feeders, in the worst case a single-line fault may force one side of the power line to supply the entire load associated with that power line. Limiting the nominal flow on each active arc to one half of its thermal rating reserves capacity for this worst-case post-fault load transfer without explicitly enumerating all contingency scenarios. Thus, constraint (23) should be interpreted as a tractable design rule for incorporating the $N-1$ requirement within the planning model, and is consistent with standard planning rules for loop-feeder or open-loop systems [24]. It does not replace detailed operational analyses such as AC power-flow validation, protection coordination, or dynamic post-fault switching studies.

Finally, arc-activation variables are binary:

$$x_{ij} \in \{0,1\} \qquad \forall (i,j) \in A'. \tag{24}$$

The above formulations correspond to a restricted version of the loop-feeder and open-loop planning models introduced in [5], adapted here for local re-optimization. The distinction between loop-feeder and open-loop topologies is handled by choosing between (20) and (21). In practice, constraints (20) and (21) are not explicitly added to the model from the beginning, since their number is exponential in the number of nodes. Following the implementation strategy adopted in [5], they are enforced dynamically through a lazy constraint generation scheme within the branch-and-cut process. The repair MILP is initially solved without the full family of subtour elimination constraints. Whenever an integer incumbent solution is found, the active subgraph induced by the selected arcs is inspected to detect violated subtour constraints. If a violated subset is identified, the corresponding inequality is added as a lazy constraint and the optimization is resumed. This separation procedure is applied according to the target topology, using constraints (20) for the loop-feeder case and constraints (21) for the open-loop case. Since the repair problems are defined on restricted subgraphs generated by the destroy operator, this dynamic enforcement remains computationally efficient in the proposed LNS framework.

Since the size of G' is limited by the destroy step, the resulting MILP instances are small and can be solved to proven optimality within manageable computational time, allowing the repair step to act as a locally exact operator within the large neighborhood search framework.

2.7. Acceptance and Stopping Criteria and Pseudocode

Let $\mathcal{L}$ denote the current solution before a destroy-and-repair move and let $\tilde{\mathcal{L}}$ denote the corresponding candidate solution obtained after reconstructing the selected subset of power lines. We adopt an improving-only acceptance rule at the move level, with the candidate solution accepted only if it strictly improves the objective value:

$$\text{if} \quad z(\tilde{\mathcal{L}}) < z(\mathcal{L}) \quad \text{then} \quad \mathcal{L} \leftarrow \tilde{\mathcal{L}}, \qquad \text{otherwise } \mathcal{L} \text{ is retained.} \tag{25}$$

This acceptance rule applies to every elementary destroy-and-repair move throughout the search. The alternation of multiple destruction phases with an exact repair step provides a balance between intensification and diversification, analogous to state-of-the-art LNS heuristics for multi-depot vehicle routing problems.

For the transition from Phase 3 to Phase 4, we compare the objective value at the beginning of a complete Phase 3 iteration with the objective value obtained after processing all pairs generated by the matching. Denoting these two values by z_{in} and z_{out}, respectively, if the relative improvement is smaller than a threshold τ_2, then the algorithm switches to Phase 4:

$$\frac{z_{\text{in}} - z_{\text{out}}}{z_{\text{in}}} < \tau_2. \tag{26}$$

Otherwise, a new complete Phase 3 iteration is performed by recomputing the centroid-based matching on the updated solution.

The stopping criterion is evaluated only in the final search phase. In the complete version of the algorithm, it is checked only during Phase 4. In alternative configurations where Phase 4 is disabled, it is checked during Phase 3. More precisely, let $z^{(k)}$ denote the objective value of the current solution at the end of the k-th complete iteration of the last active phase. The search is terminated whenever no significant objective improvement is observed over the last h complete iterations of that phase, that is,

$$\frac{z^{(k-h)} - z^{(k)}}{z^{(k-h)}} < \varepsilon, \tag{27}$$

where $\varepsilon > 0$ is a user-defined stopping threshold. In particular, if $h = 1$, the criterion simply compares the objective value of the current complete iteration with that of the immediately preceding one.

As a safeguard, the algorithm is also terminated if a maximum number $K_{\max}$ of iterations is reached, where the iteration counter includes all iterations performed throughout the search. Algorithm 1 summarizes the complete proposed LNS multi-phase matheuristic and highlights how the destroy operator evolves from the initial aggregation phase to progressively larger neighborhood moves.

Algorithm 1 Large Neighborhood Search for Meshed Distribution Network Planning

Require: Graph $G = (N, A)$, roots R, demand nodes D, costs c_{ij}, demands p_d, standard capacity $\bar{P}$, target topology (Loop-Feeder or Open-Loop), parameters $\tau_1, \tau_2, \mu, K_{\max}, h, \varepsilon$.
Ensure: A feasible set of power lines iteratively reducing total installation cost.

Construct the initial solution $\mathcal{L}^{(0)}$ by assigning each $d \in D$ to $L_d = (r_1(d), d, r_2(d))$ where $r_1(d), r_2(d) \in R$ and $r_1(d) \neq r_2(d)$. ▷ Phase 1: initialization
$\mathcal{L} \leftarrow \mathcal{L}^{(0)}$, $k \leftarrow 0$, $k_4 \leftarrow 0$, $\mathsf{phase} \leftarrow 2$.
while $k < K_{\max}$ **do**
    if $\mathsf{phase} = 2$ **then** ▷ Phase 2: DBSCAN-based aggregation
        $k \leftarrow k + 1$
        Compute the average saturation $\bar{\rho}$ of the current solution.
        Compute route centroids, DBSCAN on them to obtain clusters $\mathcal{C} = \{C_1, \dots, C_m\}$.
        for each cluster $C \in \mathcal{C}$ **do**
            if $|C| \leq \mu$ **then**
                $\mathcal{S} \leftarrow C$
            else
                $L^\star \in \arg\min_{L \in C} \rho(L)$
                $\mathcal{S} \leftarrow \{L^\star\} \cup \mathcal{N}_{\mu-1}(L^\star)$
            end if
            Apply the destroy-and-repair move to $\mathcal{S}$.
        end for
        Recompute the average saturation $\bar{\rho}$ of the current solution.
        if $\bar{\rho} \geq \tau_1$ **then**
            $\mathsf{phase} \leftarrow 3$
        end if
    else if $\mathsf{phase} = 3$ **then** ▷ Phase 3: pairwise matching
        $k \leftarrow k + 1$
        $z_{\text{in}} \leftarrow z(\mathcal{L})$
        Compute route centroids and their matching $M = \{(L_a^1, L_b^1), \dots, (L_a^q, L_b^q)\}$.
        for each pair $(L_a, L_b) \in M$ **do**
            $\mathcal{S} \leftarrow \{L_a, L_b\}$
            Apply the destroy-and-repair move to $\mathcal{S}$.
        end for
        $z_{\text{out}} \leftarrow z(\mathcal{L})$
        if $\dfrac{z_{\text{in}} - z_{\text{out}}}{z_{\text{in}}} < \tau_2$ **then**
            $\mathsf{phase} \leftarrow 4$
        end if
    else ▷ Phase 4: K-means clustering
        $k \leftarrow k + 1$
        Compute a K-means partition $\mathcal{P} = \{G_1, \dots, G_s\}$ of the current power lines.
        for each cluster $G \in \mathcal{P}$ **do**
            $\mathcal{S} \leftarrow G$
            Apply the destroy-and-repair move to $\mathcal{S}$.
        end for
        $k_4 \leftarrow k_4 + 1$
        Store $z(\mathcal{L})$ as $z_4^{(k_4)}$.
        if $k_4 \geq h$ **and** $\dfrac{z_4^{(k_4-h)} - z_4^{(k_4)}}{z_4^{(k_4-h)}} < \varepsilon$ **then**
            break
        end if
    end if
end while
return $\mathcal{L}$

3. Experiments

This section reports the computational assessment of the proposed method on realistic test instances of increasing size, up to 1150 nodes. Both loop-feeder and open-loop topologies were considered as targets. The full procedure was implemented in Python 3.13. DBSCAN, matching, and K-means were handled through scikit-learn [25], while all repair MILP subproblems were solved to optimality with the exact solver Gurobi Optimizer 11.0 [26].

Three variants of the proposed method were considered:

- `LNS-Pairs`, which stops after the pairwise phase.
- `LNS-Pairs+Triples`, which extends the pairwise phase with a final large-neighborhood phase based on triples of power lines.
- `LNS-Pairs+Quadruples`, which extends the pairwise phase with a final large-neighborhood phase based on quadruples of power lines.

The three variants were executed independently. `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples` correspond to two alternative configurations of the final Phase 4, where the current power lines are clustered into groups of average cardinality three or four, respectively.

For each instance, we also computed a lower bound using Gurobi on the complete optimization model. For the largest instances, when Gurobi was unable to provide a lower bound on the full MILP, we used instead the value of the linear relaxation of the model, still computed with Gurobi. The percentage gap was computed as $100\frac{UB-LB}{UB}$, where UB is the objective value returned by our method and LB is the available lower bound.

3.1. Test Instances

The test bed consists of realistic instances of increasing size, designed to reflect large-scale fault-tolerant distribution network planning problems.

Each instance represents a potential medium-sized voltage distribution network of a municipality, with 40% of the demand nodes located in the central area and the remaining 60% in the suburbs. The active power demand of each node is generated as the absolute value of a Gaussian random variable. Two different distributions are used: one for demand nodes in the central area, with mean 0.4 MW and standard deviation 0.2 MW, and one for demand nodes in the suburbs, with mean 0.2 MW and standard deviation 0.6 MW Any possible connection between two nodes is treated as a candidate arc; a 95 mm^2 copper cable is considered for all candidate arcs, with the power capacity limit set to 6 MW. A constant installation cost per kilometer is assumed; consequently, the objective function minimizes the total length of the network to be built. The largest instance contains 1150 nodes and about 660,000 candidate arcs. Note that exact methods already become computationally prohibitive on instances with only a few hundred nodes. Table 1 summarizes the instances considered in the experiments. The main data for all test instances are available in [27].

Table 1. Test instances used in the computational study.

Instance	$\lvert N \rvert$	$\lvert R \rvert$	$\lvert D \rvert$	$\lvert A \rvert$
Net_54	54	4	50	1425
Net_78	78	3	75	3000
Net_104	104	4	100	5350
Net_262	262	12	250	34,125
Net_525	525	25	500	137,250
Net_630	630	30	600	197,700
Net_735	735	35	700	269,150

Table 1. *Cont.*

Instance	$\|N\|$	$\|R\|$	$\|D\|$	$\|A\|$
Net_840	840	40	800	351,600
Net_945	945	45	900	445,050
Net_1050	1050	50	1000	549,500
Net_1150	1150	50	1100	659,450

3.2. Computational Results

Table 2 reports the objective values obtained by the three variants of the proposed method together with the available lower bounds for the loop-feeder topology. Table 3 report the corresponding computation times. Tables 4 and 5 report the corresponding information for the open-loop topology. For each instance, lower-bound information was computed with Gurobi in two ways. The MILP LB column reports the lower bound obtained while solving the complete mixed-integer formulation, whereas the LP LB column reports the value of the linear programming relaxation of the same model obtained by relaxing the integrality of the binary arc-activation variables. When the MILP lower bound is available, the relative optimality gaps reported in parentheses is computed with respect to it. For the largest instances, where the complete MILP did not provide a useful lower bound within the computational limit, the MILP LB entry is marked as unavailable and the LP relaxation bound is used instead. Reporting both values whenever available shows how far away the LP relaxation can be from the stronger MILP lower bound. This is helpful when interpreting the gaps for the largest instances, where only the LP bound is available and the resulting gaps may overestimate the actual distance from the optimum.

Table 2. Objective values for the loop-feeder topology. For each result, the relative optimality gap $100(\mathrm{UB} - \mathrm{LB})/\mathrm{UB}$ is reported in parentheses. Winners are marked in bold.

Instance	`LNS-Pairs`	`LNS-Pairs+Trip`	`LNS-Pairs+Quad`	MILP LB	LP LB
Net_54	44.6275 (5.0%)	**42.3889** (0.0%)	**42.3889** (0.0%)	42.3889	39.1753
Net_78	35.5217 (14.3%)	31.6174 (3.7%)	**30.4462** (0.0%)	30.4462	27.6490
Net_104	84.9246 (19.0%)	**73.7874** (6.8%)	74.3180 (7.5%)	68.7475	61.8177
Net_262	246.7676 (20.1%)	219.7740 (10.3%)	**216.5328** (9.0%)	197.1342	189.2183
Net_525	544.0627 (19.9%)	491.7712 (11.4%)	**491.3454** (11.3%)	435.9326	422.6057
Net_630	564.4586 (18.1%)	514.9479 (10.2%)	**506.9663** (8.8%)	462.4127	456.2031
Net_735	667.7905 (22.8%)	610.4919 (15.6%)	**599.2704** (14.0%)	515.3567	515.3567
Net_840	755.4908 (19.5%)	693.8032 (12.4%)	**683.5816** (11.1%)	607.8700	600.4822
Net_945	846.8055 (21.7%)	759.7139 (12.8%)	**755.5838** (12.3%)	–	662.8131
Net_1050	1052.6328 (24.9%)	930.6233 (15.1%)	**923.8944** (14.4%)	–	790.4480
Net_1150	1109.6548 (22.3%)	995.8333 (13.4%)	**990.8286** (12.9%)	–	862.5968

Table 3. Computation times for the loop-feeder topology (seconds). Winners are marked in bold.

Instance	`LNS-Pairs`	`LNS-Pairs+Trip`	`LNS-Pairs+Quad`
Net_54	**2.54**	24.42	27.35
Net_78	**5.06**	87.85	132.33
Net_104	**8.07**	263.01	259.68
Net_262	**68.61**	243.88	580.11
Net_525	**283.23**	1478.53	1625.33
Net_630	**381.52**	868.36	1922.64
Net_735	**541.19**	1302.26	2951.61
Net_840	**278.14**	1666.62	2867.19
Net_945	**394.45**	2182.19	3464.42
Net_1050	**630.62**	5173.41	3197.85
Net_1150	**637.51**	3382.09	4747.04

Table 4. Objective values for the open-loop topology. For each result, the relative optimality gap $100(\mathrm{UB} - \mathrm{LB})/\mathrm{UB}$ is reported in parentheses. Winners are marked in bold.

Instance	`LNS-Pairs`	`LNS-Pairs+Trip`	`LNS-Pairs+Quad`	**MILP LB**	**LP LB**
Net_54	45.2881 (6.4%)	**42.3889** (0.0%)	**42.3889** (0.0%)	42.3889	39.1753
Net_78	41.3029 (26.4%)	34.3780 (11.6%)	**32.2183** (5.7%)	30.3972	27.6490
Net_104	85.8340 (20.6%)	**76.0442** (10.4%)	76.0681 (10.5%)	68.1106	61.8177
Net_262	276.5604 (28.2%)	225.0161 (11.7%)	**224.0615** (11.4%)	198.6081	189.2183
Net_525	579.4190 (24.7%)	517.7115 (15.7%)	**502.3424** (13.1%)	436.5089	422.6057
Net_630	627.0558 (26.0%)	528.7934 (12.3%)	**522.3168** (11.2%)	463.7881	456.2031
Net_735	706.7650 (24.9%)	**618.3980** (14.1%)	626.1763 (15.2%)	531.0983	515.3567
Net_840	803.2614 (24.8%)	696.8543 (13.3%)	**695.3928** (13.1%)	604.2728	600.4822
Net_945	891.2955 (25.6%)	789.4810 (16.0%)	**771.4591** (14.1%)	–	662.8131
Net_1050	1074.5050 (26.4%)	**940.9400** (16.0%)	946.1962 (16.5%)	–	790.4480
Net_1150	1163.5771 (25.9%)	1018.8045 (15.3%)	**1011.3016** (14.7%)	–	862.5968

Table 5. Computation times for the open-loop topology (seconds). Winners are marked in bold.

Instance	`LNS-Pairs`	`LNS-Pairs+Trip`	`LNS-Pairs+Quad`
Net_54	**2.29**	20.10	27.83
Net_78	**6.55**	317.73	200.31
Net_104	**12.11**	456.97	254.00
Net_262	**58.66**	1262.46	1180.14
Net_525	**552.92**	1731.48	2353.97
Net_630	**682.93**	2471.33	3062.03
Net_735	**1559.13**	5609.03	4095.28
Net_840	**1143.00**	5502.20	6058.44
Net_945	**796.48**	4208.45	5073.51
Net_1050	**715.31**	5284.51	4636.74
Net_1150	**893.18**	5457.91	6307.26

Figures 2–5 report some representative network configurations generated by the proposed method, specifically by variant `LNS-Pairs+Quad` for instances Net_262 and Net_525, under both the loop-feeder and open-loop topologies. Red squares are roots, while blue dots are demand nodes.

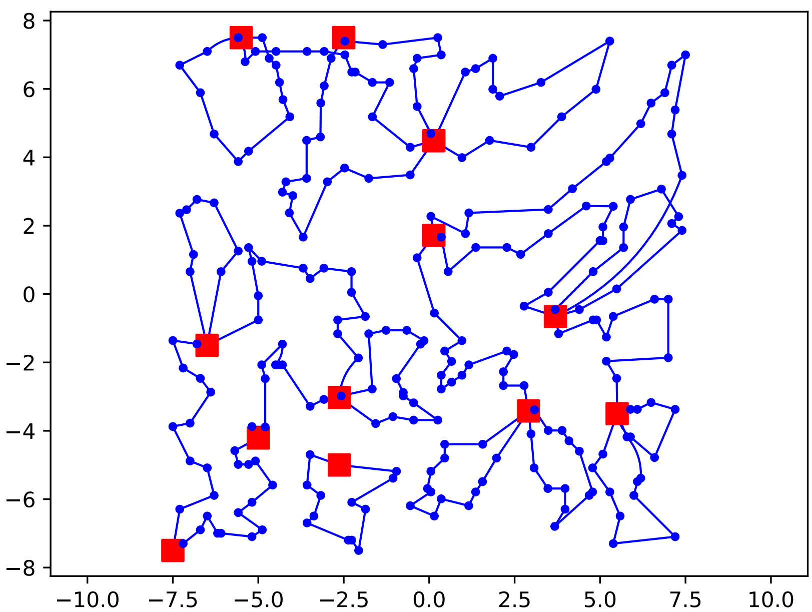

Figure 2. Net_262 solution for loop-feeder topology.

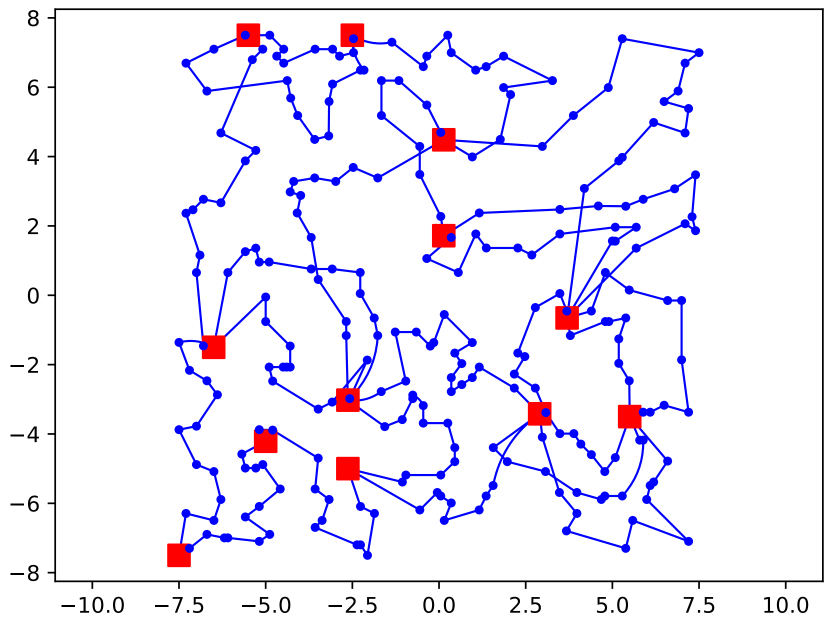

Figure 3. Net_262 solution for open-loop topology.

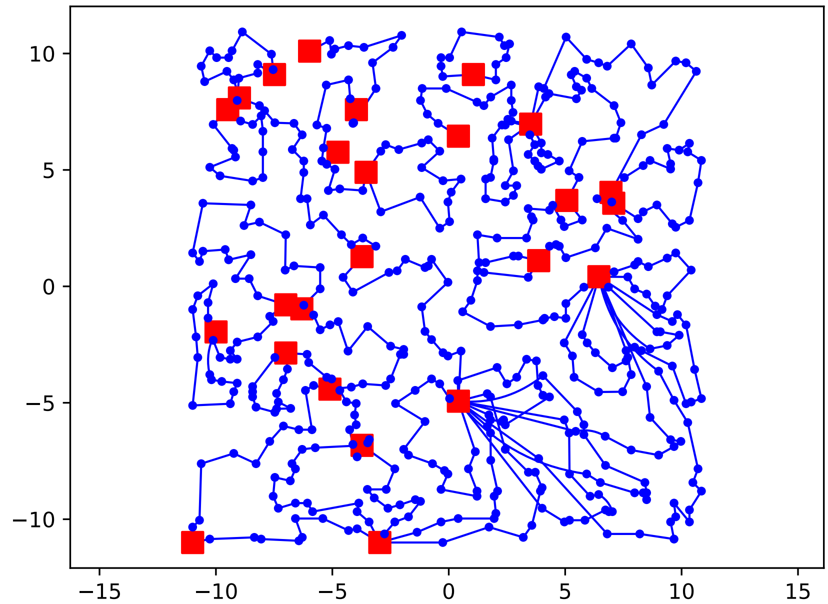

Figure 4. Net_525 solution for loop-feeder topology.

A clear trade-off between solution quality and computation time emerges among the three variants of the proposed method. To examine this trade-off more closely, we perform an additional instance-wise analysis separately for the loop-feeder and open-loop topologies. For each instance i and variant m, let $z_{i,m}$ be the objective value returned by the method and let

$$z_i^{\min} = \min_m z_{i,m} \tag{28}$$

be the best objective value obtained among the three LNS variants on that instance. We define the relative degradation in solution quality as

$$g_{i,m} = 100 \cdot \frac{z_{i,m} - z_i^{\min}}{z_i^{\min}}. \tag{29}$$

Similarly, let $t_{i,m}$ denote the computation time of variant m on instance i and let

$$t_i^{\min} = \min_m t_{i,m}, \qquad t_i^{\max} = \max_m t_{i,m}. \tag{30}$$

We normalize the time of each variant within the same instance as

$$\hat{t}_{i,m} = \begin{cases} \dfrac{t_{i,m} - t_i^{\min}}{t_i^{\max} - t_i^{\min}}, & \text{if } t_i^{\max} > t_i^{\min}, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

In the same way, we normalize the quality degradation as

$$\hat{g}_{i,m} = \begin{cases} \dfrac{g_{i,m}}{\max_m g_{i,m}}, & \text{if } \max_m g_{i,m} > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{32}$$

The quality–time compromise score is then defined as

$$C_{i,m} = \frac{\hat{g}_{i,m} + \hat{t}_{i,m}}{2}. \tag{33}$$

Smaller values of $C_{i,m}$ indicate a better compromise between solution quality and computation time. Tables 6 and 7 report, for every instance, the values of $g_{i,m}$, $\hat{t}_{i,m}$, and $C_{i,m}$ for the loop-feeder and open-loop topologies, respectively.

From the results in the tables, we observe that increasing the neighborhood size generally leads to better solutions, but also to longer computation times. The variant `LNS-Pairs` is consistently the fastest one, since its repair phase only requires the solution of relatively small MILP subproblems associated with pairs of power lines. However, the corresponding solution evolution is also more limited, and the final objective values are generally worse than those obtained by the other two variants.

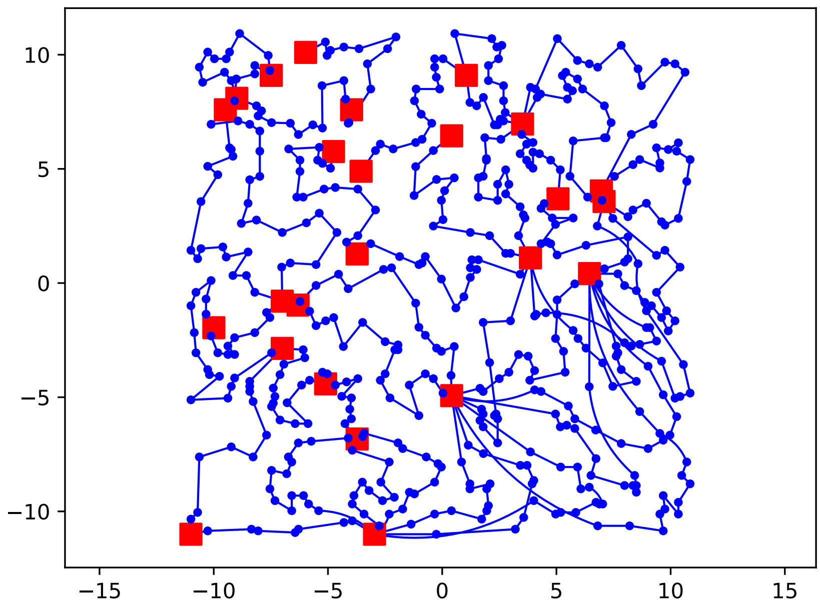

Figure 5. Net_525 solution for open-loop topology.

Table 6. Trade-off between objective value and computation time for the loop-feeder topology. Each entry reports $g_{i,m}/\hat{t}_{i,m}/C_{i,m}$. Smaller $C_{i,m}$ values are better. Winners are marked in bold.

	LNS-Pairs	LNS-Pairs+Trip	LNS-Pairs+Quad
Instance	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$
Net_54	5.3/0.00/0.50	**0.0/0.88/0.44**	0.0/1.00/0.50
Net_78	16.7/0.00/0.50	**3.8/0.65/0.44**	0.0/1.00/0.50
Net_104	**15.1/0.00/0.50**	**0.0/1.00/0.50**	0.7/0.99/0.52
Net_262	14.0/0.00/0.50	**1.5/0.34/0.22**	0.0/1.00/0.50
Net_525	10.7/0.00/0.50	**0.1/0.89/0.45**	0.0/1.00/0.50
Net_630	11.3/0.00/0.50	**1.6/0.32/0.23**	0.0/1.00/0.50
Net_735	11.4/0.00/0.50	**1.9/0.32/0.24**	0.0/1.00/0.50
Net_840	10.5/0.00/0.50	**1.5/0.54/0.34**	0.0/1.00/0.50
Net_945	12.1/0.00/0.50	**0.5/0.58/0.31**	0.0/1.00/0.50
Net_1050	13.9/0.00/0.50	0.7/1.00/0.53	**0.0/0.57/0.28**
Net_1150	12.0/0.00/0.50	**0.5/0.67/0.35**	0.0/1.00/0.50

Table 7. Trade-off between objective value and computation time for the open-loop topology. Each entry reports $g_{i,m}/\hat{t}_{i,m}/C_{i,m}$. Smaller $C_{i,m}$ values are better. Winners are marked in bold.

	LNS-Pairs	LNS-Pairs+Trip	LNS-Pairs+Quad
Instance	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$	$g_{i,m}/\hat{t}_{i,m}/C_{i,m}$
Net_54	6.8/0.00/0.50	**0.0/0.70/0.35**	0.0/1.00/0.50
Net_78	28.2/0.00/0.50	6.7/1.00/0.62	**0.0/0.62/0.31**
Net_104	12.9/0.00/0.50	0.0/1.00/0.50	**0.0/0.54/0.27**
Net_262	23.4/0.00/0.50	0.4/1.00/0.51	**0.0/0.93/0.47**
Net_525	15.3/0.00/0.50	**3.1/0.65/0.43**	0.0/1.00/0.50
Net_630	20.1/0.00/0.50	**1.2/0.75/0.41**	0.0/1.00/0.50
Net_735	14.3/0.00/0.50	0.0/1.00/0.50	**1.3/0.63/0.36**
Net_840	15.5/0.00/0.50	**0.2/0.89/0.45**	0.0/1.00/0.50
Net_945	15.5/0.00/0.50	**2.3/0.80/0.47**	0.0/1.00/0.50
Net_1050	14.2/0.00/0.50	0.0/1.00/0.50	**0.6/0.86/0.45**
Net_1150	15.1/0.00/0.50	**0.7/0.84/0.45**	0.0/1.00/0.50

The variants `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples` solve larger repair MILPs in the final phase and are consequently slower, but they allow stronger modifications of the incumbent solution and usually reach better final solutions. Among them, `LNS-Pairs+Quadruples` often gives the best objective values, whereas `LNS-Pairs+Triples` often provides a more favorable compromise between quality improvement and computation time. It is worth noting that a larger neighborhood does not necessarily imply a better final solution in every single instance. Although larger groups permit more extensive local rearrangements, they may also define different K-means partitions and hence a different sequence of destroy-and-repair moves. The method follows an improving-only acceptance rule, and different neighborhoods may lead the search along different trajectories to different local optima. In addition, the repair MILPs are solved with finite numerical precision and standard feasibility and optimality tolerances. As a result, when different variants generate different restricted subproblems and different sequences of repairs, very small numerical effects may also contribute to the observed non-monotonic behavior. This explains cases such as Net_104 in the loop-feeder topology, where the triple-based variant slightly outperforms the quadruple-based one. At the same time, the increase in neighborhood size from triples to quadruples appears to lead to a runtime increase that is not always fully justified by the corresponding improvement in solution quality. This suggests, at least on the present test set, that exploring even larger neighborhoods may be less convenient in practice.

3.3. Comparison with a Greedy Heuristic

To provide an additional reference for assessing the contribution of the proposed optimal-repair mechanism, we implemented a simple routing-inspired greedy constructive heuristic [19]. The heuristic constructs a feasible set of power lines one line at a time. Each new power line is initialized from a randomly selected unassigned demand node, which is immediately connected to the closest root, then extended by repeatedly connecting the nearest unassigned demand node. This extension continues as long as adding the nearest unassigned demand node is less costly than closing the line at the nearest root and does not violate the maximum power limit allowed for a single power line. In our experiments, this limit was set to 6 MW.

When no further demand node can be conveniently added, the current power line is closed by connecting it to a second root. In the open-loop case, this second root is required to be different from the starting root so that the resulting power line can be supplied from two distinct roots. In the loop-feeder case, the closing root may coincide with the starting root, consistently with the definition of the loop-feeder topology. The procedure is repeated until all demand nodes have been assigned to a power line.

The greedy heuristic is a fast constructive method that produces feasible networks without performing any MILP-based local re-optimization. Its speed comes from its myopic and irreversible construction rule; once a demand node has been assigned to a power line, that assignment is never reconsidered. Therefore, the heuristic may return good solutions on favorable instances where the sequence of local choices is compatible with a high-quality global structure, but it may also return poor solutions when early local decisions prevent more effective configurations. In this sense, it represents a natural constructive alternative to the proposed LNS: it does not require any MILP-based local re-optimization and never revisits previous assignments; hence, it allows us to isolate the added value of the optimal-repair mechanism.

Table 8 compares the greedy baseline with the solution having the lowest cost among those obtained by the proposed LNS variants. For each instance and topology, we report objective values and computation times along with the percentage cost improvement of our LNS approach, computed as follows:

$$\Delta z(\%) = 100 \cdot \frac{z_{\text{Greedy}} - z_{\text{BestLNS}}}{z_{\text{Greedy}}}. \tag{34}$$

The results show that the proposed LNS consistently improves the greedy heuristic on all tested instances and both topologies. On average, the best LNS variant improves the greedy solution by about 28.5% over all cases, with average improvements of about 25.6% for loop-feeder and 31.5% for open-loop, showing that the optimal-repair mechanism is able to recover large cost reductions that cannot be obtained by purely local constructive choices. The results also show a clear trade-off between construction speed and solution quality. As expected, the greedy heuristic is much faster than the LNS variants, since it only performs a one-pass constructive assignment and never revisits previous decisions. However, this speed comes at the cost of substantially worse objective values. The longer computation times of the LNS variants should also be interpreted in light of the planning nature of the problem. Distribution network planning is an offline strategic task that is typically performed once for a given planning scenario, rather than an online operational decision that must be solved in real time; therefore, in practice what matters is whether the method remains computationally tractable on large instances and able to return high-quality feasible designs within acceptable offline computation times. From this perspective, the additional computational effort required by the proposed LNS is justified by the substantial improvement in solution quality over the greedy heuristic.

Table 8. Comparison between the greedy baseline and the best LNS variant. For each instance and topology, "Best LNS" denotes the best objective value obtained among the three independently executed LNS variants. The cost improvement is computed as $\Delta z(\%) = 100 \cdot (z_{\text{Greedy}} - z_{\text{BestLNS}})/z_{\text{Greedy}}$. The labels `Trip` and `Quad` denote `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples`, respectively.

Instance and Topology	Greedy		Best LNS			Δz
	Value	Time (s)	Value	Time (s)	Variant	(%)
Net_54 LF	61.6	1.4	42.4	24.4	Trip	31.2
Net_54 OL	65.3	1.0	42.4	20.1	Trip	35.1
Net_78 LF	45.6	1.2	30.4	132.3	Quad	33.3
Net_78 OL	53.6	1.2	32.2	200.3	Quad	39.9
Net_104 LF	102.9	1.8	73.8	263.0	Trip	28.3
Net_104 OL	106.6	1.9	76.0	457.0	Trip	28.7
Net_262 LF	290.1	8.7	216.5	580.1	Quad	25.4
Net_262 OL	328.5	15.8	224.1	1180.1	Quad	31.8
Net_525 LF	632.2	57.5	491.3	1625.3	Quad	22.3
Net_525 OL	690.5	60.6	502.3	2354.0	Quad	27.3
Net_630 LF	678.4	89.4	507.0	1922.6	Quad	25.3
Net_630 OL	777.5	88.3	522.3	3062.0	Quad	32.8
Net_735 LF	771.6	102.2	599.3	2951.6	Quad	22.3
Net_735 OL	885.5	60.4	618.4	5609.0	Trip	30.2
Net_840 LF	899.2	76.9	683.6	2867.2	Quad	24.0
Net_840 OL	1031.5	127.4	695.4	6058.4	Quad	32.6
Net_945 LF	991.4	151.9	755.6	3464.4	Quad	23.8
Net_945 OL	1132.5	151.7	771.5	5073.5	Quad	31.9
Net_1050 LF	1177.6	184.5	923.9	3197.9	Quad	21.5
Net_1050 OL	1271.7	181.2	940.9	5284.5	Trip	26.0
Net_1150 LF	1311.0	224.5	990.8	4747.0	Quad	24.4
Net_1150 OL	1447.6	219.5	1011.3	6307.3	Quad	30.1

3.4. AC Power Flow Validation

To complement the MILP-based planning feasibility checks, we performed an a posteriori AC power flow validation of the final solutions obtained for the largest instance, Net_1150, under both the loop-feeder and open-loop topologies. The validation considered the final network layouts returned by the proposed LNS and tested their behavior under the most severe single-branch contingencies considered in this study. For each network, we simulated the outage of every branch directly connected to a root node. These contingencies are critical because when the first branch of a power line is unavailable, the remaining part of the line may have to be supplied from the opposite side. The analysis was carried out using 95 mm^2 copper cables at 20 kV, with maximum current rating 0.2 kA, resistance $R = 0.22\ \Omega/\text{km}$, reactance $X = 0.132\ \Omega/\text{km}$, and capacitance $C = 0.216\ \mu\text{F}/\text{km}$. A uniform load power factor of 0.9 was assumed. No voltage regulation devices were included, and the root nodes were modeled as slack buses.

All analyzed contingency cases resulted in convergent AC power flow solutions. Across both the loop-feeder and open-loop networks and over all simulated contingencies, the maximum voltage deviation observed with respect to the nominal voltage was 5.1%, while the maximum branch loading was 94%. This a posteriori validation does not replace a complete operational study including protection coordination, switching procedures, or dynamic post-fault control, but does provides additional evidence that the solutions produced by the proposed planning method are electrically consistent under the considered AC power flow assumptions.

3.5. Ablation Study

To further assess the contribution of the main components of the proposed search scheme, we performed an ablation study on the two largest instances, Net_1050 and Net_1150, under both loop-feeder and open-loop topologies. The reference configuration is `LNS-Pairs+Trip`, which first applies the pairwise matching phase and then the final K-means phase based on triples of power lines. In Table 9 we compare it with three simplified variants. The first, `No_Matching`, skips the pairwise matching phase and directly applies the K-means triple phase. The second, `No_Triples`, stops after the pairwise phase, and consequently does not use the final large-neighborhood phase based on triples. The third, `Random_Triples`, also skips the pairwise matching phase but forms triples of power lines randomly instead of using K-means clustering. This comparison allows us to identify the contribution of the structured construction of triple neighborhoods.

Table 9. Ablation study on the two largest instances. Values in parentheses report the percentage variation with respect to `LNS-Pairs+Trip`; positive percentages indicate worse (more costly) solutions.

Configuration	Instance	Top.	Value	Time (s)
`LNS-Pairs+Trip`	Net_1050	LF	930.6	5173.4
	Net_1050	OL	940.9	5284.5
	Net_1150	LF	995.8	3382.1
	Net_1150	OL	1018.8	5457.9
`No_Matching`	Net_1050	LF	933.6 (+0.3%)	3443.7
	Net_1050	OL	944.6 (+0.4%)	4861.3
	Net_1150	LF	987.3 (−0.9%)	3010.1
	Net_1150	OL	1027.4 (+0.8%)	4499.5
`No_Triples`	Net_1050	LF	1052.6 (+13.1%)	630.6
	Net_1050	OL	1074.5 (+14.2%)	715.3
	Net_1150	LF	1109.7 (+11.4%)	637.5
	Net_1150	OL	1163.6 (+14.2%)	893.2
`Random_Triples`	Net_1050	LF	1673.0 (+79.8%)	423.2
	Net_1050	OL	1382.4 (+46.9%)	567.0
	Net_1150	LF	1547.6 (+55.4%)	578.3
	Net_1150	OL	1666.1 (+63.5%)	543.6

The comparison with the `No_Matching` configuration shows that skipping the pairwise matching phase does not necessarily lead to a large deterioration on these very large instances. In all four cases, the final objective value remains close to that of the reference `LNS-Pairs+Trip` configuration, and in one case it is slightly better. This behavior is consistent with the path-dependent nature of the matheuristic. Changing the sequence of destroy-and-repair moves may lead the search to different local optima; thus, the pairwise matching phase contributes to a more structured search trajectory, but the final triples phase can recover much of the solution quality even when the pairwise phase is skipped.

The comparison with the `No_Triples` configuration confirms the importance of the final large-neighborhood phase. Stopping after the pairwise phase is much faster, but consistently produces worse objective values on all four tested cases. This indicates that the additional MILP repairs on triples of power lines provide a substantial improvement in solution quality at the cost of longer computation times.

The comparison with the `Random_Triples` configuration highlights a different and stronger effect. Forming triples randomly leads to shorter computation times, but the resulting objective values are substantially worse. This behavior was observed even though the random variant was allowed to run with a maximum number of search cycles equal to 60, twice the limit used in the corresponding structured configurations. In many runs, this limit was reached, suggesting that random selection of triples leads to a more erratic search trajectory in which the algorithm continues to perform repair attempts but these attempts do not produce the same systematic improvements as in the structured K-means version. This indicates that K-means grouping identifies spatially coherent groups of power lines that are much more suitable for joint MILP-based repair than randomly selected triples.

Overall, the ablation study supports the design of the proposed method. The final large-neighborhood repair phase is important for improving solution quality, but its effectiveness depends on selecting destroyed groups in which power lines are geographically close. The pairwise matching phase helps to organize the search trajectory, while the K-means phase provides the main large-neighborhood intensification mechanism.

3.6. Sensitivity Analysis on Search Thresholds

We performed a sensitivity analysis on the main threshold parameters controlling the evolution and termination of the search. The analysis was carried out on the largest test instance, Net_1150, under both the loop-feeder and open-loop topologies. In particular, we considered the parameters that were expected to have the largest direct impact on the search dynamics: the average saturation threshold τ_1, which controls the transition from Phase 2 to Phase 3; the relative improvement threshold τ_2, which controls the transition from Phase 3 to Phase 4; and the stopping threshold ε, which is used in the last active search phase. The remaining parameters were not included in this sensitivity analysis because they either have a more structural role or act mainly as safeguards. The μ parameter fixes the maximum size of the Phase 2 repair subproblems and was kept constant to control the computational effort of each MILP repair, while the h parameter only defines the length of the improvement window used in the stopping criterion; finally, $K_{\max} = 30$ is a maximum-iteration safeguard, and was never reached as the active termination condition in the reported experiments.

This analysis was designed to assess the robustness of the proposed method. Across all tested configurations, the algorithm always returned feasible solutions for both topologies, and no parameter setting caused the search to fail or degenerate. Thus, the main effect of the tested thresholds is on the balance between solution quality and computation time, rather than on the stability of the method.

The first block of Table 10 concerns the effect of τ_1. Larger values of τ_1 keep the algorithm longer in Phase 2, where elementary one-customer power lines are aggregated before the matching phase starts. This avoids applying many pairwise destroy-and-repair moves to very small power lines, which would require repeated MILP repair calls with limited restructuring effect. In our tests, changing τ_1 mainly affected the intermediate search trajectory: larger values tend to produce shorter overall computation times, while the final objective values remained very similar.

The second block varies τ_2 and ε, showing the expected trade-off between solution quality and computation time. Large values lead to shorter runs but also to worse objective values, since the algorithm switches phase or stops too early. Smaller values allow for a more intensive search and may improve the final objective value, but at the price of longer computation times. The effect is not perfectly monotonic, as expected for a path-dependent matheuristic with an improving-only acceptance rule; different thresholds may induce different sequences of destroy-and-repair moves.

Overall, the tested configurations confirm that the method is robust with respect to reasonable variations of the main thresholds. The default configuration provides a reasonable compromise between solution quality and computational effort.

Table 10. Sensitivity analysis on the main search thresholds for the largest instance, Net_1150.

Test	τ_1	τ_2	ε	LF Value	LF Time (s)	OL Value	OL Time (s)
Sensitivity with respect to τ_1							
S1	0.50	10^{-3}	10^{-3}	989.5	3810.4	1010.7	6389.6
S2	0.60	10^{-3}	10^{-3}	990.6	3562.8	1011.3	5650.2
S3	0.70	10^{-3}	10^{-3}	993.9	3348.5	1017.9	5402.7
Sensitivity with respect to τ_2 *and* ε							
S4	0.65	5×10^{-2}	5×10^{-2}	1025.8	1096.1	1051.9	1592.2
S5	0.65	10^{-3}	10^{-3}	995.8	3382.1	1018.8	5457.9
S6	0.65	5×10^{-4}	5×10^{-4}	991.2	6360.7	1014.0	10,126.2
S7	0.65	2×10^{-5}	2×10^{-5}	983.3	6632.0	1023.9	6889.7

4. Discussion

The computational results obtained in this study clearly demonstrate the effectiveness of the proposed multi-phase matheuristic and show a remarkable scalability of the proposed method. All tested instances, up to 1150 nodes and about 660,000 candidate arcs, were solved within reasonable computation times. At the same time, the comparison with the available lower bounds provides useful evidence on the quality of the obtained solutions. This indication is stronger when the bound is obtained from the complete MILP and weaker when only the LP relaxation bound is available. These conclusions hold for both target topologies considered, namely, loop-feeder and open-loop.

Scalability is particularly relevant from both a methodological and a practical viewpoint. Power distribution networks in the planning stage tend to become increasingly large and complex, and methods capable of handling large-scale instances are becoming more and more important. Moreover, this planning problem is of major practical relevance. Ensuring a reliable electricity supply is increasingly vital, and the economic values involved in network design are so large that even the modest improvements in solution quality obtained by moving closer to the optimum can translate into substantial savings in practice.

The comparison with the existing literature is particularly significant. To the best of our knowledge, there are currently no exact methods capable of solving fault-tolerant distribution network planning instances of the size considered here. More generally, exact or MILP-based methods in related distribution network optimization studies have often been validated on benchmark systems with only a few dozen nodes, and at most on instances of a few hundred nodes. In the closest exact reference for fault-tolerant meshed distribution planning, provided by [5], the reported computational study reaches instances up to 154 nodes, already showing the computational difficulty of exact optimization in this setting. A similar observation applies to heuristic approaches proposed in related distribution network optimization problems. Even when the methodology is heuristic or metaheuristic, computational studies reported in the literature are often restricted to benchmark systems with sizes remaining in the range of a few dozen or few hundred nodes. By contrast, the proposed multi-phase LNS is able to handle instances that are several times larger while still producing solutions of satisfactory quality within a reasonable time. Thus, for large instances, our multi-phase matheuristic is able to outperform the computationally intractable monolithic models by decomposing complexity into small local subproblems.

As is generally the case with heuristic and matheuristic methods for complex combinatorial optimization problems, the proposed LNS cannot guarantee global optimality. This is the inherent trade-off of heuristic optimization, the purpose of which is to obtain good feasible solutions within computation times substantially shorter than those required by exact methods, especially for instances beyond their computational reach. Accordingly, the proposed approach was specifically designed to solve very large instances.

Our multi-phase structure mitigates premature convergence by changing and progressively enlarging the explored neighborhoods, although it cannot formally guarantee that the global optimum will be reached. The geographical destroy criterion may exclude the joint re-optimization of spatially distant power lines, including lines that may be electrically related. However, geographical proximity is particularly relevant because installation costs are mainly distance-dependent. The ablation study shows that replacing K-means clustering with randomly formed triples increases solution costs by between 46.9% and 79.8% across the experiments, supporting the effectiveness of geographically founded neighborhoods. Neighborhoods based on electrical information, for example impedance or flow sensitivity, may represent an interesting direction for future research.

The comparison among the three variants of the proposed method also provides useful algorithmic insight. The two variants including Phase 4, namely, `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples`, generally produce better solutions than `LNS-Pairs`, confirming the benefit of a final large-neighborhood phase beyond pairwise re-optimization. This improvement comes at the expected price of longer computation times due to the larger repair subproblems induced by the final phase.

The comparison between `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples` suggests that the use of quadruples does not provide consistent improvements over triples, while it tends to increase the overall computation time. This indicates that enlarging the final neighborhood beyond triples may offer limited practical benefit in the present setting. In particular, the current results do not provide strong motivation for exploring even larger group sizes, such as quintuple-based final phases.

A limitation of the computational assessment concerns the strength of the available lower bounds. When the lower bound is obtained from the complete MILP, the resulting gap provides a meaningful indication of the quality of the heuristic solution. For the largest instances, however, the only obtainable lower bound resulted from solving the linear relaxation of the full model. Since this LP relaxation provides a weaker and more optimistic bound, the resulting gap is generally larger and less informative, although it remains the only lower-bound information that can be obtained at that scale. This is not a limitation of the proposed LNS itself but a consequence of the intrinsic difficulty of computing strong lower bounds for very large instances of complete fault-tolerant planning models.

A second limitation is related to the quality of the input data. As in all optimization-based planning approaches, the practical value of the computed solutions depends on the accuracy of the estimated installation costs and of the other input parameters used in the model; therefore, the quality of the final network design is necessarily linked to the reliability of the available planning data.

Regarding reproducibility, each combination of instance, topology, and LNS variant was executed once using a fixed random seed equal to 42 for the randomized components of the algorithm. In particular, this seed was used in the K-means procedure, where the initialization may depend on random choices. DBSCAN, as implemented in scikit-learn, does not require a random seed. The seed value was chosen a priori as a conventional arbitrary value commonly used in programming and machine learning examples, and was not tuned on the test instances.

For completeness, Table 11 reports the numerical values of the algorithmic parameters used in all computational experiments. These values were kept fixed across all instances, topologies, and LNS variants. Although μ could in principle be scaled with the instance size, we kept it fixed because it directly controls the size of the induced repair subproblems, and hence the computational effort required by the exact MILP solver. DBSCAN was used with the default scikit-learn parameters, specifically `eps=0.5` and `min_samples=5`.

Table 11. Algorithmic parameter values used in the experiments.

Param.	Value	Description
τ_1	0.65	Average saturation threshold for switching from Phase 2 to Phase 3
τ_2	10^{-3}	Relative improvement threshold for switching from Phase 3 to Phase 4
μ	20	Maximum number of power lines destroyed simultaneously in Phase 2
ε	10^{-3}	Stopping threshold for the last active search phase
h	1	Number of complete iterations used in the stopping criterion
$K_{\max}$	30	Maximum number of complete search iterations

5. Conclusions

This work presents an optimal-repair Large Neighborhood Search (LNS) matheuristic for the planning of large fault-tolerant distribution networks in loop-feeder and open-loop topologies. The approach exploits a vehicle routing reframing to guide a destroy-and-repair search in which selected subsets of power lines are removed and reconstructed by a topology-specific Mixed-Integer Linear Programming (MILP) repair model solved to local optimality on restricted subgraphs.

From a methodological viewpoint, the main contribution lies in the four-phase optimal-repair matheuristic. Starting from a highly fragmented initial solution with one power line per customer, the algorithm first performs a DBSCAN-based aggregation phase, then a pairwise re-optimization phase based on centroid matching, and finally, when requested, a larger-neighborhood phase based on K-means clustering of triples or quadruples of power lines. This progressive enlargement of the search neighborhood allows the method to combine effective early aggregation with increasingly stronger local re-optimization moves.

From a computational perspective, the main outcome is scalability. While exact MILP approaches are effective on small to medium planning instances, they become impractical as network size grows. The proposed method extends the tractable range by a substantial margin, enabling the solution of instances that are well beyond the reach of exact global optimization within a reasonable time. To the best of our knowledge, alternative heuristics in the literature do not currently offer comparable scalability on the same combination of fault-tolerant topology requirements and electrical feasibility constraints, which limits the possibility of direct heuristic benchmarking at the largest scales addressed here.

To assess solution quality, the computational study was complemented with lower-bound information computed by Gurobi on the complete planning model. Whenever possible, the lower bound was obtained from the full MILP; for the largest instances, where this was computationally prohibitive, the LP relaxation was used instead. An additional comparison against the greedy heuristic further supports the quality of the obtained solutions, showing that the proposed LNS produces good feasible solutions with substantially lower costs than a purely constructive approach, especially on large instances.

The comparison among the three algorithmic variants also provides a useful practical indication. The variants including the final large-neighborhood phase, namely, `LNS-Pairs+Triples` and `LNS-Pairs+Quadruples`, generally improve the solution quality with respect to `LNS-Pairs`, though at the price of longer computation times. At the same time, the experiments show that using quadruples does not always lead to improvements over triples, while it tends to increase the repair effort. This suggests that a final phase based on triples already provides a favorable compromise between solution quality and computational burden. Overall, the results show that embedding an exact topology-aware repair operator within a multi-phase LNS framework yields a practical and effective tool for planning fault-tolerant distribution networks at scales where exact global optimization becomes computationally prohibitive. An a posteriori AC power flow validation on the largest instance further confirmed the electrical consistency of the obtained designs under all tested single-branch outages, with a maximum voltage deviation equal to 5.1% across the tested cases.

Future Work

Several directions can be pursued to strengthen and extend the proposed framework. First, future research could address additional fault-tolerant network topologies beyond the two considered in this work, including more complex configurations that may offer higher levels of resilience. Second, richer electrical modeling features could be incorporated into the repair model while preserving the overall search structure. Third, further destroy and/or repair operators could be introduced, as is commonly done in large neighborhood search frameworks. Finally, stronger lower bounds would allow a more accurate assessment of solution quality on the largest instances.

Author Contributions: Conceptualization, R.B., M.M. and L.N.; methodology, R.B. and L.N.; software, L.N.; validation, R.B., M.M. and L.N.; investigation, R.B., M.M. and L.N.; resources, A.G.; data curation, M.M.; writing—original draft preparation, R.B.; writing—review and editing, R.B., M.M. and L.N.; supervision, R.B.; project administration, A.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available in Mendeley Data at https://doi.org/10.17632/rwb6rn29fs.1 [27].

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

MILP	Mixed-Integer Linear Programming
LNS	Large Neighborhood Search
MDVRP	Multi-Depot Vehicle Routing Problem
DSP	Distribution System Planning
LP	Linear Relaxation
LB	Lower Bound

References

1. Paul, S.; Poudyal, A.; Poudel, S.; Dubey, A.; Wang, Z. Resilience assessment and planning in power distribution systems: Past and future considerations. *Renew. Sustain. Energy Rev.* **2024**, *189*, 113991. [CrossRef]
2. National Renewable Energy Laboratory (NREL). *Current Practices in Distribution Utility Resilience Planning for Hazards: Wildfire, Hurricanes, and Extreme Weather*; Technical Report NREL/TP-5R00-88000; National Renewable Energy Laboratory (NREL): Golden, CO, USA, 2024.
3. *IEEE Std 1366-2022*; IEEE Guide for Electric Power Distribution Reliability Indices. IEEE Standards Association: Piscataway, NJ, USA, 2022. [CrossRef]

4. Li, Z.; Wu, W.; Tai, X.; Zhang, B. A Reliability-Constrained Expansion Planning Model for Mesh Distribution Networks. *IEEE Trans. Power Syst.* **2021**, *36*, 948–960. [CrossRef]
5. Bruni, R.; Geri, A.; Maccioni, M.; Nati, L. Optimal planning of power distribution networks with fault-tolerant configuration. *Comput. Oper. Res.* **2026**, *185*, 107248. [CrossRef]
6. Shaw, P. Using Constraint Programming and Local Search to Solve Vehicle Routing Problems. In *Proceedings of the Fourth International Conference on Principles and Practice of Constraint Programming (CP-98)*; Springer: Berlin, Germany, 1998; pp. 417–431. [CrossRef] [PubMed]
7. Ahuja, R.K.; Ergun, Ö.; Orlin, J.B.; Punnen, A.P. A Survey of Very Large-Scale Neighborhood Search Techniques. *Discret. Appl. Math.* **2002**, *123*, 75–102. [CrossRef]
8. Ropke, S.; Pisinger, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transp. Sci.* **2006**, *40*, 455–472. [CrossRef]
9. Pisinger, D.; Ropke, S. Large Neighborhood Search. In *Handbook of Metaheuristics*, 2nd ed.; Gendreau, M., Potvin, J.Y., Eds.; Springer: Boston, MA, USA, 2010; pp. 399–419. [CrossRef]
10. Vidal, T. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. *Comput. Oper. Res.* **2021**, *140*, 105643. [CrossRef]
11. Saker, A.; Eltawil, A.; Ali, I. Adaptive Large Neighborhood Search Metaheuristic for the Capacitated Vehicle Routing Problem with Parcel Lockers. *Logistics* **2023**, *7*, 72. [CrossRef]
12. Rojas-Cuevas, I.-D.; Olivares-Benitez, E.; Ramos, A.S.; Nucamendi-Guillén, S. A Capacitated Vehicle Routing Model for Distribution and Repair with a Service Center. *Logistics* **2025**, *9*, 28. [CrossRef]
13. Boschetti, M.A.; Letchford, A.N.; Maniezzo, V. Matheuristics: Survey and Synthesis. *Int. Trans. Oper. Res.* **2023**, *30*, 2840–2866. [CrossRef]
14. Ghiaskar, A.; Taheri, F. Reproducible Multi-Metric Benchmarking Framework for Metaheuristic Algorithms in Constrained Engineering Optimization. *Arch. Comput. Methods Eng.* **2026**, 1–42. [CrossRef]
15. Ghiaskar, A.; Taheri, F. Evaluation Protocol Sensitivity in Frequency-Constrained Truss Optimization: A Comparative Study of Adaptive and Evolutionary Metaheuristics. *Biomimetics* **2026**, *11*, 235. [CrossRef] [PubMed]
16. Lavorato, M.; Franco, J.F.; Rider, M.J.; Romero, R. Imposing Radiality Constraints in Distribution System Optimization Problems. *IEEE Trans. Power Syst.* **2012**, *27*, 172–180. [CrossRef]
17. Nati, L; Bruni, R.; Maccioni, M.; Geri, A. Efficient handling of radiality constraints for large-scaled power distribution networks, *Electr. Power Syst. Res.* **2025**, *241*, 111278. [CrossRef]
18. Glamocanin, V.; Filipovic, V. Open Loop Distribution System Design. *IEEE Trans. Power Deliv.* **1993**, *8*, 1900–1906. [CrossRef]
19. Toth, P.; Vigo, D. *Vehicle Routing: Problems, Methods, and Applications*, 2nd ed.; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 2014. [CrossRef]
20. Montoya-Torres, J.R.; Franco, J.L.; Isaza, S.N.; Jiménez, H.F.; Herazo-Padilla, N. A literature review on the vehicle routing problem with multiple depots. *Comput. Ind. Eng.* **2015**, *79*, 115–129. [CrossRef]
21. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*; AAAI Press: Portland, OR, USA, 1996; pp. 226–231.
22. MacQueen, J. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1*; University of California Press: Berkeley, CA, USA, 1967; pp. 281–297.
23. Archetti, C.; Speranza, M.G. A Survey on Matheuristics for Routing Problems. *Euro J. Comput. Optim.* **2014**, *2*, 223–246. [CrossRef]
24. Schlabbach, J.; Rofalski, K.H. *Power System Engineering: Planning, Design, and Operation of Power Systems and Equipment*; Wiley-VCH: Weinheim, Germany, 2008; pp. 50–51.
25. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
26. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2026. Available online: https://gurobi.com (accessed on 11 June 2026).
27. Maccioni, M.; Bruni, R.; Nati, L. Dataset for the planning of Large Fault-Tolerant Distribution Networks. *Mendeley Data* **2026**, V1. [CrossRef]