Deterministic Branch and Cut Algorithm for Multiobjective Optimization of Protective Device Allocation in Radial Distribution Systems

Zbigniew Galias, *Senior Member, IEEE*

***Abstract*—Installing protective devices enhances reliability of power distribution systems by means of failure separation. Finding optimal positions of protective devices to minimize a given objective function can be achieved using various single-objective optimization methods. Solutions obtained for different objective functions may differ significantly. Multiobjective optimization algorithms may be used to solve the optimization problem taking into account more than one objective function. In this work, a new branch and cut algorithm is proposed to solve the multiobjective optimization problem of protective device allocation in radial distribution systems with a single feeder. It is shown that the proposed algorithm can successfully handle very large power distribution systems. The performance of the algorithm is compared with the performance of three other approaches: 1) the exhaustive search; 2) an evolutionary algorithm; and 3) a reinforcement learning algorithm. It is shown that the proposed algorithm outperforms other methods both in terms of the computation time and the quality of the results.**

***Index Terms*—Branch and cut (BC) algorithm, depth-first search (DFS), evolutionary algorithm (EA), power distribution reliability, protective device, reliability index.**

NOMENCLATURE

Notations

n	Number of nodes.
m	Number of line segments.
p	Number of protective devices.
v_k	Node with index k (at the vertex k).
c_k	Line segment from v_k toward the generator.
λ_{v_k}	Average failure rate of the node v_k.
λ_{c_k}	Average failure rate of the line segment c_k.
t_{v_k}	Average total duration of failures for v_k.
t_{c_k}	Average total duration of failures for c_k.
λ_k	Average failure rate for v_k and c_k.
t_k	Average duration of failures for v_k and c_k.
N_k	Number of users at the node v_k.
P_k	Average (active) power of the node v_k.
$\bar{t}$	the total duration of failures.
$\bar{N}$	Total number of users.
$\bar{P}$	Total average power.
p_k	Parent of the vertex k.
C_k	Set of children of the vertex k.
D_k	Set of descendants of the vertex k.
E_k	Union of D_k and $\{k\}$.
$\bar{N}_k$	Number of users for $E_k = \{k\} \cup D_k$.
$\bar{P}_k$	Average power for E_k.
$\bar{t}_k$	Average failure time for E_k .
$\bar{\lambda}_k$	Average failure rate for E_k.
Q	Set of positions of protective devices.
g_{AE}	Gain in AENS.
g_{SD}	Gain in SAIDI.
g_{SF}	Gain in SAIFI.
BC	Branch and cut algorithm.
ES	Exhaustive search method.
EA	Evolutionary algorithm.
RL	Reinforcement learning algorithm.
DFS	Depth-first search.
SAIDI	System average interruption duration index.
SAIFI	System average interruption frequency index.
MAIDI	Momentary average interruption duration index.
MAIFI	Momentary average interruption freq. index.
AENS	Average energy not supplied.

Received 9 November 2023; revised 25 April 2024, 5 July 2024, and 12 October 2024; accepted 20 November 2024. Date of publication 16 December 2024; date of current version 5 March 2025. This work was supported by AGH University of Krakow. Paper no. TII-23-4405.
The author is with the AGH University of Krakow, 30-059 Krakow, Poland (e-mail: Zbigniew.Galias@agh.edu.pl).
Digital Object Identifier 10.1109/TII.2024.3507080

I. INTRODUCTION

CONSEQUENCES of outages in power distribution networks may be reduced by the introduction of protective devices, such as automatic sectionalizing switches or fuses [1]. Isolating a faulty part is realized by a fuse melting or opening a remotely controlled switch located between the energy source and the failure location. The problem of finding optimal number and positions of protective devices is still an active field of research [2], [3], [4]. The number of protective devices which can be installed in a given network is limited by investment and operational costs.

For a fixed number of devices to be installed a single-objective optimization problem is to select positions of these devices to minimize a certain objective function [5], [6], [7]. Reliability indexes characterizing distribution systems which are often used as objective functions in protective device allocation problems include SAIDI and SAIFI. Outages can be classified as sustained (permanent) and momentary (lasting less than 5 min). The SAIDI and SAIFI metrics characterize sustained outages. Less commonly used reliability indexes MAIDI and MAIFI characterize momentary outages. Another performance index frequently used as an objective function is AENS also called EENS [6], [8], [9], which can be used to estimate user interruption costs.

Several heuristic methods to solve a single objective optimization problem are known. A genetic algorithms approach was presented in [5]. A solution based on simulating annealing algorithm was proposed in [6]. A particle swarm optimization algorithm was proposed in [10] to find the optimal number and positions of sectionalizers and breaker switches. Ant colony optimization-based methods were used to minimize AENS in [11]. A method based on dynamic programming and fuzzy logic was described in [12]. Genetic algorithms combined with a fuzzy logic were used in [13]. Switch allocation problem in radial distribution systems with distributed generation was considered in [14]. The sectionalizing strategy for parallel system restoration was discussed in [15]. A brute force method and EAs were applied in [16] to find optimal placement of sectionalizing switches minimizing SAIDI in medium voltage distribution networks.

Efficient methods to solve single-objective allocation problems are based on integer programming. Binary programming was used in [17] to find the types and locations of protective devices to optimize SAIFI. The mixed integer linear programming approach was used to minimize the user interruption costs in conjunction with investment and operational costs in [7]. In [8], mixed integer linear programming techniques were applied to systems with distributed generation. Mixed integer programming was used to solve the switch placement problem considering nonzero switch malfunction probability in [18]. The problem to find the optimal number, type, and location of protective devices to minimize the investment cost with additional constrains based on reliability indexes was solved using the mixed integer linear programming approach in [19]. A mixed-integer nonlinear programming model with constrains to find the type and optimal positions of protective devices was proposed in [20]. Value-based mixed-integer nonlinear programming was used to study the problem of finding optimal places of sectionalizing switches and fuses in networks with distributed generation in [2]. An efficient tree structure-based algorithm to solve the problem of switch allocation for the minimization of a single objective function was proposed in [21]. The algorithm is designed for radial networks with a single feeder and outperforms integer programming optimization algorithms for this class of networks. The problem of switch placement in distribution networks taking into account substation overloading during load transfer was solved using a mixed-integer linear programming model in [3]. Mixed integer linear programming was used in [4] to find the optimal placement of remote-controlled switches minimizing the system risk in the IEEE 33 bus system.

In these approaches, the optimal placement of protective devices is selected based on a single objective function. However, it is known that minimization of one of the objectives is usually not equivalent to the minimization of other ones. Multiobjective optimization aims at finding tradeoffs between various objectives. A solution which cannot be improved in one objective without worsening one or more of other objectives is called Pareto optimal. Multiobjective optimization aims at finding the complete set of Pareto optimal solutions representing the best tradeoffs between objectives.

Various multiobjective approaches to the protective device allocation problem have been described in literature. Particle swarm optimization and the differential search algorithm are used to solve a multiobjective optimization problem in [22] and [23]. The optimization objectives are to minimize the number of customers not supplied and the expected energy not supplied (EENS), respectively. The second optimized parameter in both approaches is the number of switches. The authors formulate optimization problems as a multiobjective optimization problem. However, in both cases the problems can be reformulated as a single objective optimization problem considered for various number of switches. A multiobjective particle swarm optimization technique to find the number and positions of feeders, sectionalizing switches and tie-lines is presented in [24]. The objectives are the minimization of the installation and maintenance costs and the maximization of network reliability. A multiobjective optimization approach using the particle swarm theory to solve the switch placement problem was presented in [22]. Optimization of positions of switches and protective devices to minimize the total equipment cost and two reliability indexes (SAIDI and SAIFI) was considered in [25]. The problem is solved using a memetic EA. In [26], the nondominant sorting genetic algorithm was used for multiobjective optimization of SAIDI, SAIFI, and equipment costs in networks with distributed generation. A tree structure-based algorithm to optimize AENS, SAIDI, and SAIFI was presented in [27]. The economic analysis of isolator placement in radial networks was carried out in [28] using genetic algorithms and the particle swarm optimization approach. Optimization of the cost of energy not served and the SAIDI in small networks composed of 34 and 59 loads were considered.

In this work, we consider the problem to simultaneously minimize two or more objective functions AENS, SAIDI, SAIFI, MAIDI, and MAIFI in radial distribution networks. AENS characterizes user interruption costs for the system operator while reliability indexes help to assess the influence of power supply outages on customers. Multiobjective optimization of AENS and reliability metrics allows a designer of the distribution network to take into account interests of both parties when selecting the solution. To solve the problem a deterministic BC algorithm is proposed. The algorithm uses the directed rooted tree structure of radial distribution networks (compare [21]). In the proposed algorithm, at each node a set of partial solutions with $p \leq p_{\max}$ protective devices is constructed. The number of partial solutions is reduced by skipping those which cannot lead to a Pareto optimal complete solution. The algorithm returns for each $p \leq p_{\max}$ the set of Pareto optimal solutions. Based on the results, for each p one may easily compute profits obtained by installing p protective devices taking into account the reduction in user interruption costs as well as investment and operational costs. The results of the BC algorithm may help the distribution system designer to select the most convenient solution. A detailed description of the algorithm is presented and a pseudo-code for the algorithm is provided. The proposed algorithm can be implemented in a programming language or in the MATLAB environment; no commercial solvers are needed to solve the optimization problem at hand. Case studies including a medium size and a large network are employed to show the effectiveness of the proposed algorithm. The algorithm is compared with three other methods: 1) the ES method; 2) an EA; and 3) a RL algorithm. It is shown that the algorithm can handle very large networks and is orders of magnitudes faster than other methods.

The rest of this article is organized as follows. In Section II-A, several notions regarding radial distribution networks needed to formulate the optimization problem are introduced. In Section II-B, the objective functions are defined and the single- and multiobjective optimization problems are formulated. The multiobjective optimization algorithm is presented in Section III. The ES method, the EA, and the RL algorithm are briefly described. In Section IV, the performance of the proposed algorithm is assessed using two single-feeder distribution networks with a radial topology: 1) the IEEE 123 node test feeder; and 2) the IEEE 8500 test feeder. Finally, Section V concludes this article.

II. Optimization Algorithm

A. Radial Distribution Network

Each radial distribution network with a single feeder can be represented as a directed rooted tree with the supply node designated to be the root. Let us denote by n_u and n_d the numbers of user and distribution nodes, respectively. The network contains a single supply node and hence the total number of nodes is $n = n_\mathrm{u} + n_\mathrm{d} + 1$. Let v_k for $k = 1, 2, \ldots, n-1$ denotes the user or distribution node with the index k and v_n denotes the supply node. The node v_k corresponds to the vertex k in the directed rooted tree structure. Let us denote by p_k the parent of the vertex k, i.e., the vertex connected to the vertex k on the path to the root. Each vertex corresponding to a distribution or a user node has a single parent. The vertex n corresponding to the supply node has no parent. The vertex i is a child of k if k is the parent of i. The vertex i is a descendant of k if k lies on the path from i to the root. By C_k we denote the set of children of k and by D_k we denote the set of descendants of k. The set $E_k = \{k\} \cup D_k$ has the structure of a directed rooted tree with the vertex k being the root. A leaf is a vertex with no children. An internal vertex is a vertex which is neither a leaf nor the root. In radial distribution networks leaves and internal vertices correspond to user and distribution nodes, respectively. The line segment connecting nodes v_k and v_{p_k} is denoted as c_k. The line segment c_k corresponds to the edge (k, p_k) in the rooted tree structure. A network with n nodes (vertices) contains $m = n - 1$ line segments (edges).

Each user node v_k is characterized by the number of users N_k and the average power P_k. The *total average power* $\bar{P}$ and the *total number of users* $\bar{N}$ for the whole network can be computed as

$$\bar{P} = \sum_{k=1}^{n} P_k, \quad \bar{N} = \sum_{k=1}^{n} N_k. \tag{1}$$

We assume that the period of analysis is one year. The *average failure rate* λ_k of the vertex k is defined as the sum of the average number of failures λ_{v_k} of the node v_k and the average number of failures λ_{c_k} of the line segment c_k in the period of analysis (i.e., $\lambda_k = \lambda_{v_k} + \lambda_{c_k}$). The *average total duration of failures* t_k of the vertex k is the sum of the average durations of failures t_{v_k} of the node v_k and the average durations of failures t_{c_k} of the line segment c_k in the period of analysis (i.e., $t_k = t_{v_k} + t_{c_k}$). The units for t_k, t_{v_k}, and t_{c_k} are hours per year. The values λ_k and t_k characterize elements v_k and c_k in terms of failures causing sustained interruptions (SI). Similar parameters may be defined for momentary interruptions (MI).

The *total failure rate* $\bar{\lambda}$ and the *total duration of failures* $\bar{t}$ for the whole network can be computed as

$$\bar{\lambda} = \sum_{k=1}^{n} \lambda_k, \quad \bar{t} = \sum_{k=1}^{n} t_k. \tag{2}$$

B. Multiobjective Optimization Problem

Let us first recall definitions of several objective functions used in the protective device allocation problem [6], [29]. The AENS, the SAIDI, the SAIFI, the MAIDI, and MAIFI can be computed using the following formulas: $\mathrm{AENS} = \sum_{k=1}^{n} U_k P_k$, $\mathrm{SAIDI} = \sum_{k=1}^{n} U_k N_k / \bar{N}$, $\mathrm{SAIFI} = \sum_{k=1}^{n} f_k N_k / \bar{N}$, $\mathrm{MAIDI} = \sum_{k=1}^{n} W_k N_k / \bar{N}$, and $\mathrm{MAIFI} = \sum_{k=1}^{n} h_k N_k / \bar{N}$, where U_k (W_k) is the total duration (in hours per year) of all sustained (momentary) interruptions involving the node v_k, while f_k (h_k) is the average number of sustained (momentary) interruptions involving the node v_k during the period of analysis.

Installing protective devices in a distribution system may reduce the values of objective functions for this network. In case of a failure a protective device is activated which isolates the failure. A protective device may be placed in one of the line segments c_k with $1 \leq k \leq m$.

The device placed at the vertex k (i.e., in the line segment c_k) is denoted as d_k. The problem of allocation of p protective devices is to select from the set of vertices $V = \{1, 2, \ldots, m\}$ a subset Q with p elements. The search space is the set $\Sigma_p = \{Q \subset V : \mathrm{card}(Q) = p\}$, where $\mathrm{card}(Q)$ denotes the number of elements of Q. Its size is $N = \binom{m}{p}$. Let F be one of the objective functions AENS, SAIDI, SAIFI, MAIDI, or MAIFI. The value of the objective function F for the selected set Q is denoted as $F(Q)$. For $p = 0$ ($Q = \emptyset$) each failure results in an outage of the whole system. The outage rate f_j and the total duration of all interruptions U_j are the same for each node: $f_j = f = \sum_{i=1}^{n} \lambda_i = \bar{\lambda}$ and $U_j = U = \sum_{i=1}^{n} t_i = \bar{t}$. In consequence $\mathrm{SAIFI}(\emptyset) = \bar{\lambda}$, $\mathrm{SAIDI}(\emptyset) = \bar{t}$, and $\mathrm{AENS}(\emptyset) = \bar{P} \cdot \bar{t}$.

The single-objective optimization problem involving the objective function F is to find the solution $Q_F(p) \in \Sigma_p$ minimizing the objective function F, i.e., $Q_F(p) = \arg\min_{Q \in \Sigma_p} F(Q)$.

In a multiobjective optimization we search for $Q \in \Sigma_p$ minimizing $N_\mathrm{obj} \geq 2$ objective functions. Let $F_1, F_2, \ldots, F_{N_\mathrm{obj}}$ be the selected objective functions. Let us consider two solutions $Q', Q'' \in \Sigma_p$. We say that Q'' *is dominated* by Q' if for all objective functions $F_k(Q') \leq F_k(Q'')$ and $F_k(Q') < F_k(Q'')$ for at least one objective function. A solution $Q \in \Sigma_p$ is called *Pareto optimal* if it is not dominated by any other solutions in Σ_p. The goal of multiobjective optimization is to find the *Pareto front* $\mathrm{P_F}(p) = \{Q \in \Sigma_p : Q \text{ is Pareto optimal}\}$.

III. Multiobjective Optimization Algorithm

In this section, we present the BC algorithm to solve the multiobjective optimization problem defined in the previous section. A single-objective version of the optimization algorithm is presented in [21]. We also briefly describe three other multiobjective optimization algorithms, which are used for comparison purposes.

A. Branch and Cut Algorithm for Multiobjective Optimization

Let us first derive formulas for objective functions for a selected set Q. Let us denote by $\bar{P}_k$ the sum of average powers, by $\bar{N}_k$ the total number of users, by $\bar{t}_k$ the sum of average failure times, and by $\bar{\lambda}_k$ the sum of failure rates for the set $E_k = \{k\} \cup D_k$ (i.e., the vertex k and its descendants)

$$\bar{P}_k = \sum_{i\in E_k} P_i, \quad \bar{N}_k = \sum_{i\in E_k} N_i, \quad \bar{t}_k = \sum_{i\in E_k} t_i, \quad \bar{\lambda}_k = \sum_{i\in E_k} \lambda_i. \tag{3}$$

Let us select a protective device d_k at the vertex $k \in Q$. Let us define the subset R_k of descendants D_k of the vertex k: the vertex i belongs to R_k if and only if 1) $i \in D_k$ (i is a descendant of k); 2) $i \in Q$ (there is a device d_i at the vertex i); and 3) the path between vertices k and i does not contain protective devices other than d_k and d_i.

The device d_k is activated if there is a failure in the part of the network between vertex k and vertices in R_k. Hence, the average activation time for the protective device d_k can be computed as $\bar{t}_k - \sum_{i\in R_k} \bar{t}_i$. The average number of failures for which the device d_k is activated is $\bar{\lambda}_k - \sum_{i\in R_k} \bar{\lambda}_i$. When the device d_k is activated the average power of active load nodes is $(\bar{P} - \bar{P}_k)$ and the percentage of active users is $(\bar{N} - \bar{N}_k)/\bar{N}$.

Hence, the gain $g_{\mathrm{AE}}(k)$ in AENS obtained by installing the protective device d_k is the product of the average device activation time $\bar{t}_k - \sum_{i\in R_k} \bar{t}_i$ and the average power $(\bar{P} - \bar{P}_k)$ of active load nodes, i.e., $g_{\mathrm{AE}}(k) = (\bar{t}_k - \sum_{i\in R_k} \bar{t}_i)(\bar{P} - \bar{P}_k)$. Similarly, the gains in SAIDI and SAIFI are $g_{\mathrm{SD}}(k) = (\bar{t}_k - \sum_{i\in R_k} \bar{t}_i)(\bar{N} - \bar{N}_k)/\bar{N}$ and $g_{\mathrm{SF}}(k) = (\bar{\lambda}_k - \sum_{i\in R_k} \bar{\lambda}_i)(\bar{N} - \bar{N}_k)/\bar{N}$.

The total gain obtained for the set Q is the sum of gains obtained for all vertices belonging to Q. Thus, we have the following formulas for objective functions:

$$\mathrm{AENS}(Q) = \bar{P} \cdot \bar{t} - \sum_{k\in Q} (\bar{P} - \bar{P}_k)\left(\bar{t}_k - \sum_{i\in R_k} \bar{t}_i\right) \tag{4}$$

$$\mathrm{SAIDI}(Q) = \bar{t} - \sum_{k\in Q} \frac{\bar{N} - \bar{N}_k}{\bar{N}}\left(\bar{t}_k - \sum_{i\in R_k} \bar{t}_i\right) \tag{5}$$

$$\mathrm{SAIFI}(Q) = \bar{\lambda} - \sum_{k\in Q} \frac{\bar{N} - \bar{N}_k}{\bar{N}}\left(\bar{\lambda}_k - \sum_{i\in R_k} \bar{\lambda}_i\right). \tag{6}$$

The formulas for $\mathrm{MAIDI}(Q)$ and $\mathrm{MAIFI}(Q)$ are similar to formulas (5) and (6) for $\mathrm{SAIDI}(Q)$ and $\mathrm{SAIFI}(Q)$. The only difference is that the evaluation should be based on the information regarding MI instead of SI.

Two types of protective devices are considered. The first type is an automatic sectionalizing switch—a recloser. With this type of protective devices in case of a failure occurring behind the switch the whole system experiences a momentary interruption. The device recloses after a certain time interval to allow the fault to repair itself and lockouts on a sustained outage after a number of unsuccessful attempts to reclose. Note that automatic sectionalizing switches may worsen the MAIDI and MAIFI indexes. Therefore, when solving the optimization problem with automatic sectionalizing switches these two indexes are not considered to avoid an exponential increase of the number of Pareto optimal solutions.

The second type of protective devices considered in this work are fuses—low cost automatic sectionalizing devices. We assume that a fuse clearing scheme is employed. In this scheme, in case of a failure, a fuse is cleared (blows) and the rest of the network does not experience even a momentary interruption (users prevented from a momentary interruption may still experience a voltage sag). It follows that installing fuses may improve MAIDI and MAIFI indexes.

First, let us consider the problem of optimal placement of sectionalizing switches with the objective functions AENS, SAIDI, and SAIFI related to SI. The BC algorithm is based on the construction of partial solutions at each vertex of the directed rooted tree structure. A partial solution at the vertex j is a set Q of positions of protective devices such that $Q \subset E_j = \{j\} \cup D_j$. Let us define the gains $g_{\mathrm{AE}}(Q)$, $g_{\mathrm{SD}}(Q)$, and $g_{\mathrm{SF}}(Q)$ in AENS, SAIDI, and SAIFI obtained for the partial solution Q

$$g_{\mathrm{AE}}(Q) = \bar{P}\bar{t} - \mathrm{AENS}(Q) = \sum_{k\in Q} (\bar{P} - \bar{P}_k)\left(\bar{t}_k - \sum_{i\in R_k} \bar{t}_i\right) \tag{7}$$

$$g_{\mathrm{SD}}(Q) = \bar{t} - \mathrm{SAIDI}(Q) = \sum_{k\in Q} \frac{\bar{N} - \bar{N}_k}{\bar{N}}\left(\bar{t}_k - \sum_{i\in R_k} \bar{t}_i\right) \tag{8}$$

$$g_{\mathrm{SF}}(Q) = \bar{\lambda} - \mathrm{SAIFI}(Q) = \sum_{k\in Q} \frac{\bar{N} - \bar{N}_k}{\bar{N}}\left(\bar{\lambda}_k - \sum_{i\in R_k} \bar{\lambda}_i\right). \tag{9}$$

The *average activation time* $a_{\mathrm{SI}}(Q)$ of the partial solution Q at the vertex j is the total time for which devices located at positions in Q are active

$$a_{\mathrm{SI}}(Q) = \begin{cases} \bar{t}_j & \text{if } j \in Q \\ \sum_{i\in R_j} \bar{t}_i & \text{if } j \notin Q. \end{cases} \tag{10}$$

The *average failure number* $b_{\mathrm{SI}}(Q)$ of the partial solution Q at the vertex j is the total number of failures for which devices located at positions in Q are active

$$b_{\mathrm{SI}}(Q) = \begin{cases} \bar{\lambda}_j & \text{if } j \in Q \\ \sum_{i\in R_j} \bar{\lambda}_i & \text{if } j \notin Q. \end{cases} \tag{11}$$

The procedure to find the complete Pareto front is presented as the Algorithm 1. For each vertex j the set S_j of partial solutions is constructed. Each partial solution is represented as an object s with the attributes Q, g_{AE}, g_{SD}, g_{SF}, a_{SI}, and b_{SI}. The DFS algorithm is used for traversing the tree. For each vertex j

all combinations of partial solutions found for children C_j of j are considered. For each combination two partial solutions are constructed: without and with the protective device at the vertex j. In the first case, the parameters $g_{\rm AE}$, $g_{\rm SD}$, $g_{\rm SF}$, $a_{\rm SI}$, and $b_{\rm SI}$ are simply the sums of corresponding parameters for selected partial solutions (lines 9–13). In the second case, the gains $g_{\rm AE}$, $g_{\rm SD}$, and $g_{\rm SF}$ are increased according to (7)–(9) (lines 16–18). $a_{\rm SI}$ and $b_{\rm SI}$ are updated using formulas (10) and (11) (lines 19–20).

Algorithm 1: Find Pareto fronts for $p \le p_{\max}$.

1: **Procedure** FINDPARTIALSOLUTIONS($j, S_j, p_{\max}$)
2: $\{i_1, i_2, \ldots, i_k\} \leftarrow C_j$ ▷indexes of children of v_j
3: **for** $\ell \leftarrow 1,\ k$ **do** ▷process children
4: FINDPARTIALSOLUTIONS($i_\ell, S_{i_\ell}, p_{\max}$)
5: **end for**
6: $S_j \leftarrow \emptyset$ ▷initialize the set of partial solutions
7: **repeat**
8: select $s_\ell \in S_{i_\ell}$ for $\ell = 1, 2, \ldots, k$
9: $s.g_{\rm AE} \leftarrow \sum_{\ell=1}^{k} s_\ell.g_{\rm AE}$
10: $s.g_{\rm SD} \leftarrow \sum_{\ell=1}^{k} s_\ell.g_{\rm SD}$
11: $s.g_{\rm SF} \leftarrow \sum_{\ell=1}^{k} s_\ell.g_{\rm SF}$
12: $s.a_{\rm SI} \leftarrow \sum_{\ell=1}^{k} s_\ell.a_{\rm SI}$
13: $s.b_{\rm SI} \leftarrow \sum_{\ell=1}^{k} s_\ell.b_{\rm SI}$
14: $s.Q \leftarrow \bigcup_{\ell=1}^{k} s_\ell.Q$
15: ADDPARTIALSOLUTION($s, j, S_j, p_{\max}$)
16: $s.g_{\rm AE} \leftarrow s.g_{\rm AE} + (\bar{t}_j - s.a_{\rm SI})(\bar{P} - \bar{P}_j)$
17: $s.g_{\rm SD} \leftarrow s.g_{\rm SD} + (\bar{t}_j - s.a_{\rm SI})(\bar{N} - \bar{N}_j)/\bar{N}$
18: $s.g_{\rm SF} \leftarrow s.g_{\rm SF} + (\bar{\lambda}_j - s.b_{\rm SI})(\bar{N} - \bar{N}_j)/\bar{N}$
19: $s.a_{\rm SI} \leftarrow \bar{t}_j$
20: $s.b_{\rm SI} \leftarrow \bar{\lambda}_j$
21: $s.Q \leftarrow s.Q \cup \{j\}$
22: ADDPARTIALSOLUTION($s, j, S_j, p_{\max}$)
23: **until** no more selections
24: **end procedure**
25: **procedure** FINDPARETOFRONT($p_{\max}$)
26: FINDPARTIALSOLUTIONS($n, S_n, p_{\max}$)
27: **for** $p \leftarrow 1,\ p_{\max}$ **do**
28: $\mathrm{P_F}(p) \leftarrow$ {nondominated solutions from $S_n \cap \Sigma_p$}
29: **end for**
30: **end procedure**

To find the Pareto front the procedure FINDPARTIALSOLUTIONS is called with the parameter n corresponding to the supply node (line 26). On exit, the set S_n contains solutions with the number of protective devices $p \le p_{\max}$. Pareto fronts $\mathrm{P_F}(p)$ are constructed by removing dominated solutions from $S_n \cap \Sigma_p$.

Algorithm 1 includes the procedure ADDPARTIALSOLUTION to add a partial solution s at the vertex j to the set of partial solutions S_j. This procedure is presented as the Algorithm 2. It eliminates partial solutions which cannot produce a Pareto optimal complete solution. Without skipping some partial solutions the size of S_j grows exponentially when we move toward the root. A simple elimination criterion is based on the number of devices. Partial solutions with the number of devices exceeding $p_{\max}$ are skipped (line 16).

Algorithm 2: Add partial solution.

1: **function** ISDOMINATED(r, s, j)
2: **if** $s.g_{\rm AE} - r.g_{\rm AE} \le \max(0, (\bar{P} - \bar{P}_j)(s.a_{\rm SI} - r.a_{\rm SI}))$ **then**
3: **return** false ▷r is not dominated by s (AENS)
4: **end if**
5: **if** $s.g_{\rm SD} - r.g_{\rm SD} \le \max(0, (\bar{N} - \bar{N}_j)(s.a_{\rm SI} - r.a_{\rm SI}))$ **then**
6: **return** false ▷r is not dominated by s (SAIDI)
7: **end if**
8: **if** $s.g_{\rm SF} - r.g_{\rm SF} \le \max(0, (\bar{N} - \bar{N}_j)(s.b_{\rm SI} - r.b_{\rm SI}))$ **then**
9: **return** false ▷r is not dominated by s (SAIFI)
10: **end if**
11: **return** true ▷r is dominated by s
12: **end function**
13: **procedure** ADDPARTIALSOLUTION($s, j, S_j, p_{\max}$)
14: $p \leftarrow \#(s.Q)$ ▷the number of devices in s
15: **if** $p > p_{\max}$ **then** ▷too many devices
16: **return**
17: **end if**
18: $T \leftarrow \{r \in S_j : \#(r.Q) = p\}$ ▷solutions with p devices
19: **for** $r \in T$ **do**
20: **if** ISDOMINATED(r, s, j) **then**
21: $S_j \leftarrow S_j \setminus \{r\}$ ▷remove r from S_j
22: **end if**
23: **end for**
24: $T \leftarrow \{r \in S_j : \#(r.Q) = p\}$ ▷solutions with p devices
25: **for** $r \in T$ **do**
26: **if** ISDOMINATED(s, r, j) **then**
27: **return** ▷do not add s to S_j
28: **end if**
29: **for**
30: $S_j \leftarrow S_j \cup \{s\}$ ▷add partial solution s to S_j
31: **procedure**

The second condition is based on comparing partial solutions in terms of the gains $g_{\rm AE}$, $g_{\rm SD}$, and $g_{\rm SF}$. Let us consider two complete solutions $Q_1, Q_2 \in \Sigma_p$ equal outside the set $E_j = \{j\} \cup D_j$, i.e., $Q_1 \setminus E_j = Q_2 \setminus E_j$. Let us define two partial solutions $Q_1' = Q_1 \cap E_j$ and $Q_2' = Q_2 \cap E_j$. The following lemma can be used to compare complete solutions Q_1 and Q_2 based on partial solutions Q_1' and Q_2'.

Lemma 1: 1) If $g_{\rm AE}(Q_1') - g_{\rm AE}(Q_2') > \max(0, (\bar{P} - \bar{P}_j)(a_{\rm SI}(Q_1') - a_{\rm SI}(Q_2')))$, then $g_{\rm AE}(Q_1) \ge g_{\rm AE}(Q_2)$.

2) If $g_{\rm SD}(Q_1') - g_{\rm SD}(Q_2') > \max(0, (\bar{N} - \bar{N}_j)(a_{\rm SI}(Q_1') - a_{\rm SI}(Q_2')))$, then $g_{\rm SD}(Q_1) \ge g_{\rm SD}(Q_2)$.

3) If $g_{\rm SF}(Q_1') - g_{\rm SF}(Q_2') > \max(0, (\bar{N} - \bar{N}_j)(b_{\rm SI}(Q_1') - b_{\rm SI}(Q_2')))$, then $g_{\rm SF}(Q_1) \ge g_{\rm SF}(Q_2)$.

The proof of the first part of the above lemma is presented in [21]. The proofs of other parts are similar. Note that if assumptions of all three parts of Lemma 1 are satisfied then the partial solution Q_2' cannot lead to a Pareto optimal solution (Q_2 based on Q_2' is dominated by Q_1 based on Q_1') and hence can be skipped. In Section IV, it is shown that the Algorithm 2 successfully reduces the number of partial solutions and in consequence the algorithm works also for networks with thousands of nodes.

In Algorithm 2, the multiobjective optimization with the objectives AENS, SAIDI, and SAIFI is implemented. If there are two objectives, then a certain line in Algorithm 2 should be commented out (for example the line 6, if the objective functions are AENS and SAIFI).

Algorithms 1 and 2 can be used to optimize positions of protective devices in case when protective devices are sectionalizing switches. As mentioned before, in case of fuses one may modify Algorithms 1 and 2 by adding objective functions MAIDI and MAIFI related to MI. This can be achieved by adding attributes characterizing partial solutions from the point of view of MI: the gain g_{MD} in MAIDI, the gain g_{MF} in MAIFI, the average activation time a_{MI}, and the average failure number b_{MI}. The update formulas for g_{MD}, g_{MF}, a_{MI}, and b_{MI} are similar to formulas for updating g_{SD}, g_{SF}, a_{SI}, and b_{SI} with the difference that the values $\bar{t}_j$ and $\bar{\lambda}_j$ are based on MI. One should also add two conditions in the procedure ISDOMINATED comparing partial solutions in terms of MAIDI and MAIFI. The conditions are analogous to conditions in lines 5 and 8 of the Algorithm 2.

The algorithms presented in this section can be used in the design process in the following way. First the maximum number $p_{\max}$ of protective devices to be installed in the network should be specified. Then, the Algorithm 1 is called to find Pareto optimal solutions for all $p \le p_{\max}$. In the final step, the system designer selects p and the Pareto optimal solution which are most convenient taking into account user interruption costs, investment, and maintenance costs as well as reliability indexes.

B. Exhaustive Search

For verification and comparison, three multiobjective optimization algorithms are implemented: 1) the ES algorithm; 2) the EA; and 3) the RL algorithm. In the ES algorithm, the solution set S is constructed by searching the whole search space Σ_p. This algorithm may be used to verify whether results obtained using other algorithms are correct. Initially, the solution set is empty $S = \emptyset$. Then, for each $Q \in \Sigma_p$, the solution set S is updated based on the values of objective functions. Each $Q' \in S$ dominated by Q is removed from S. Q is added to S under the condition that it is not dominated by any $Q' \in S$. After scanning the whole search space Σ_p the solution set S is equal to the Pareto front $\mathrm{P_F}(p)$. The size of the search space is $N = \binom{m}{p}$, where m is the number of line segments.

C. Evolutionary Algorithm

The second algorithm used for comparison purposes is the EA. In the EA algorithm, r independent runs are carried out. In each run, g_n generations are constructed. Each generation contains g_s individuals. The first generation in each run is random. The next generation is constructed from the previous generation using the selection operator combined with the crossover and mutation operators. Nondominated solutions are always kept in the next generation by applying the domination-based selection operator. Two types of mutation are employed. In the global mutation a randomly selected protective device is moved to a randomly selected admissible position. In the local mutation, a randomly selected device is moved to a neighboring line segment. Simulations show that promoting local mutations significantly improves the performance of the EA algorithm. In the crossover operation two individuals are merged by randomly selecting p positions of devices from the combined set of positions existing in both solutions. In each run, we find a solution set S containing solutions not dominated by other solutions found during this run. Solution sets obtained in all runs are merged to construct the final result. The number of test solutions in the EA algorithm is $N = r g_n g_s$.

D. Reinforcement Learning for Multiobjective Optimization

The third algorithm is based on RL. The problem under study belongs to the class of multiobjective combinatorial optimization problems which search in a finite set to find optimal values of multiple cost objective functions. A survey on using RL for solving combinatorial optimization problems can be found in [30] and [31]. An overview of multiobjective RL techniques is presented in [32]. The first step in applying RL approach to the combinatorial optimization problem is to model it as a sequential decision-making process, where the agent performs a sequence of actions (steps) in order to find a solution. Let us denote by $R_k(a_k)$ the gain in a selected objective function obtained by taking action a_k in the kth step. For the problem considered in this work at each node of the network, we have to make a decision whether to install a protective device at this node and hence the number of steps is fixed. In the RL approach, the goal is to find the optimal policy that maximizes the expected discounted sum of rewards obtained in subsequent steps $\sum_{k=1}^{n} \gamma^k R_k(a_k)$, where n is the number of steps and γ is the scalar discount factor ($0 < \gamma \le 1$). Using $\gamma < 1$ encourages short-term rewards. In this study, we use $\gamma = 1$ since we are interested only in the total gain obtained after a fixed number of steps and it does not matter how early the gain is obtained. To improve the predictions by planning several steps ahead we use the knowledge of the environment. The most natural approach is to utilize the tree structure of the power network. This leads to the group of methods known as a Monte-Carlo tree search algorithms. The nodes in a tree are visited using the breadth-first search algorithm, which shows superior performance compared to other choices. An action at a given step is selected according to the current policy. To implement the multiobjective optimization problem the policy promotes currently nondominated solutions. When a nondominated solution is found by the agent the current set of nondominated solutions is updated and the policy is modified by a stochastic gradient ascent. Once the computations are completed the set of nondominated solutions found serves as an approximation of the Pareto front. In order to increase the probability of finding the optimal policy, which allows to find the true Pareto front, the calculations are repeated starting each time from a uniform policy in which actions in a given step have the same probability.

E. Computational Details

All algorithms are implemented in the C++ language. The C++ code implementing algorithms and the data necessary to carry out the computations are available from the author on request and can be downloaded at http://www.zet.agh.edu.pl/tii2025. Input data required for each algorithm are stored in two text files: 1) the line segments file; and 2) the nodes file.

The line segments file contains m lines, where m is the number of line segments in the network. Each line contains six real numbers characterizing a given line segment. The format of each line is “SN EN AFR AFD L LT,” where SN and EN are indexes of the nodes where the line segment starts and ends, AFR is average failure rate of the line segment, AFD is the average failure duration in hours, L is the length of the line segment, and LT is the line type (LT=0 for overhead lines, LT=1 for ground lines).

The nodes file contains $n = m + 1$ lines, where n is the number of nodes. The format of each line is “ID X Y NT P N AFR AFD LT,” where ID is the node index, X and Y are coordinates of the node, NT is the node type (NT=1 for generator, NT=2 for load node, NT=3 for distribution node), P is the average active power of the node, N is the number of users at this node, AFR is the average failure rate for this node, AFD is the average failure duration in hours, and LT is the location type (LT=0 for countryside, LT=1 for town).

When calling a given optimization algorithm the user has to provide the list of functions to be optimized (AENS, SAIDI, SAIFI) and the the number p of protective devices to be placed in the network. The ES and BC algorithms are parameter free. For the EA algorithm the user selects the number of runs r, the number of generations g_n, the generation size g_s, the crossover probability, and the global and local mutation probabilities. For the RL algorithm the user selects the number of runs, the number of iterations in each run and the parameter defining the strength of promoting nondominated solutions in the Monte Carlo search. On output each algorithm returns the number of solutions found and for each of these solutions the positions of protective devices and the values of all optimized functions.

IV. Performance of the Branch and Cut Algorithm

To assess the performance of the BC optimization algorithm two three-phase test feeders are considered [33], [34]. The data for these feeders are available at http://sites.ieee.org/pes-testfeeders/resources/. As a first example the IEEE 123 node test feeder is selected. This distribution system is often studied as an example of medium size distribution systems [35], [36]. The network contains $n = 124$ nodes and $m = 123$ line segments. States of three-phase switches are selected to be as defined in the feeder data in their normal state. The structure of the IEEE 123 node test feeder is shown in Fig. 1. For each node its average active power is defined in the feeder data. There are five underground line segments in the network. The remaining ones are overhead line segments. Reliability parameters for line segments and nodes are selected based on their type. For a given type of the line segment the average failure rate is selected to be proportional to the line segment length.

To test whether the algorithm scales up to large problems the IEEE 8500 Node Test Feeder [37], [38] is selected as a second test example. We consider the version of this network with $n = 3637$ nodes. The tree structure corresponding to this network contains a large number of vertices without siblings (a sibling for a vertex k is a vertex j such that $p_k = p_j$, i.e., k and j have the same parent). Most vertices without siblings correspond to user nodes connected to a transformer (distribution node) supplying only one user node. If k is a vertex without siblings and j is its parent with node parameters $N_j = 0$ and $P_j = 0$, then it is always better to install a protective device at the vertex j (in the line segment c_j) than at the vertex k. Hence, the optimization problem can be simplified by not permitting to install protective devices at vertices with no siblings. Applying this procedure to the IEEE 8500 Node Test Feeder leads to a network with 2250 nodes (see Fig. 2).

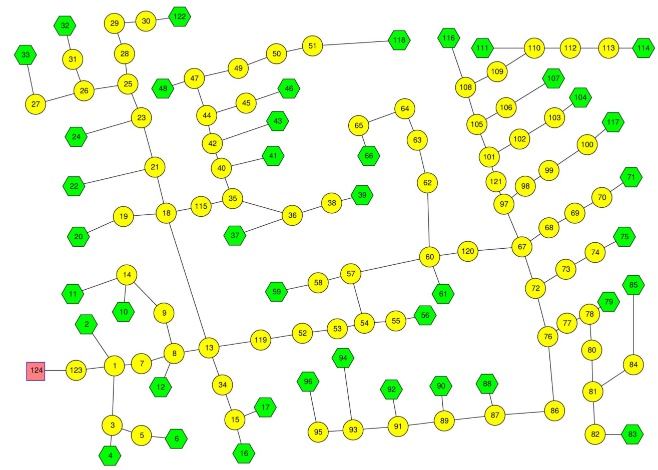

Fig. 1. Structure of the IEEE 123 node test feeder with $m = 123$ line segments and $n = 124$ nodes. The supply node is plotted as a red square. Distribution and load nodes are plotted as yellow circles and green hexagons, respectively.

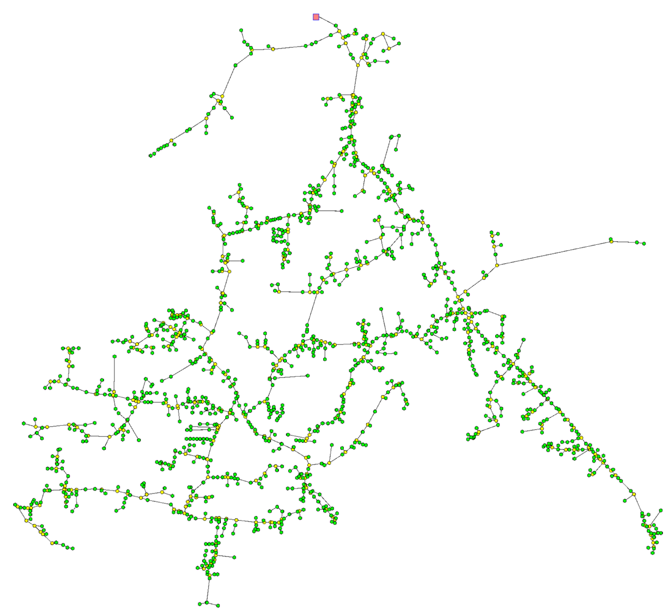

Fig. 2. Structure of the IEEE 8500 Node Test Feeder, the number of nodes is reduced to $n = 2250$ by removing single-child distribution nodes.

A. Optimization Results for the IEEE 123 Node Test Feeder

First, we consider the problem of finding positions of p sectionalizing switches in the IEEE 123 network to simultaneously optimize values of AENS, SAIDI, and SAIFI. Results

TABLE I
PERFORMANCE OF MULTIOBJECTIVE OPTIMIZATION ALGORITHMS FOR THE IEEE 123 NODE TEST FEEDER

		BC		ES		EA				RL			
p	S	N	t[s]	N	t[s]	N	t[s]	S'	R	N	t[s]	S'	R
1	**7**	469	0.00	123	0.00	700	0.02	7	**7**	349	0.00	7	**7**
2	**7**	1093	0.01	7503	0.13	4700	0.19	7	**7**	7856	0.21	7	**7**
3	**15**	2235	0.04	302621	6.02	19200	0.81	15	**15**	45575	1.57	15	**15**
4	**10**	4008	0.07	$9.08 \cdot 10^6$	179.67	54900	2.23	10	**10**	53315	1.68	10	**10**
5	**1**	6354	0.13	$2.16 \cdot 10^8$	2997.92	630000	26.20	1	**1**	25610	0.84	1	**1**
6	**1**	9308	0.23	$4.25 \cdot 10^9$	58900.32	800000	43.95	1	**1**	30985	1.11	1	**1**
7	**1**	13290	0.44			$1.22 \cdot 10^7$	572.93	1	**1**	513917	19.19	1	**1**
8	**1**	18884	0.72			$8.32 \cdot 10^6$	409.61	1	**1**	$1.11 \cdot 10^6$	53.11	1	**1**
9	**5**	26297	0.96			$3.10 \cdot 10^7$	1653.81	5	**5**	$1.62 \cdot 10^6$	96.67	5	**5**
10	**9**	35308	1.24			$6.80 \cdot 10^7$	3755.17	9	**9**	$1.03 \cdot 10^7$	908.27	9	**9**
11	**9**	45661	1.70			$2.92 \cdot 10^8$	13922.03	9	**9**	$3.07 \cdot 10^6$	300.88	9	**9**
12	**8**	57140	2.13			$3.76 \cdot 10^8$	17765.17	8	**8**	$1.07 \cdot 10^7$	1101.11	8	**8**
13	**9**	69207	3.40			$4.00 \cdot 10^8$	20448.57	9	**9**	$1.10 \cdot 10^8$	12347.13	9	**9**
14	**8**	81357	4.86			$1.21 \cdot 10^9$	62742.09	8	**8**	$3.88 \cdot 10^8$	38087.50	8	**8**
15	**9**	93530	6.81			$4.00 \cdot 10^9$	195933.26	7	7	$2.08 \cdot 10^8$	47648.08	9	**9**
20	**16**	142425	11.90			$4.00 \cdot 10^9$	203630.66	12	1	$1.00 \cdot 10^9$	129617.93	15	4

p is the number sectionalizing switches, *S* is the number of Pareto optimal solutions, *t* is the computation time, *N* is number of partial solutions for the BC method and the number of test solutions for other methods, *S'* is the size of the solution set found by the EA and RL methods, *R* is the number of Pareto optimal solutions found by the EA and RL methods.

obtained using four methods (BC—the BC algorithm; ES—the ES method; EA—the EA; and RL—RL algorithm) are shown in Table I. Parameters of the EA and RL algorithms are selected using the trial-and-error approach to optimize their performance. For the EA algorithm the number of generations g_n and the generation size g_s are selected based on p. The maximum values used for large p are $g_n = 2000$ and $g_s = 2000$. The maximum number of runs is $r = 100$. For the RL algorithm, the number of runs varies between 1 and 10 000, while the number of iterations per run varies between 1000 and 100 000.

In Table I, we report the size S of Pareto front and the computation time t. All the calculations are carried out using a single core 3.4-GHz processor. N is the number of partial solutions for the BC algorithm and the number of test solutions for other methods. For the EA and RL algorithms we also report the size S' of the solution set and the number R of Pareto optimal solutions found. Cases in which all Pareto optimal solutions are found ($S = R$) are depicted in boldface. In case of EA and RL algorithms, the computations are stopped once the complete Pareto front is found (which is known from the computations using the BC or ES algorithms). The total number of test solutions considered before finding the Pareto front provides information on convergence speeds of these algorithms.

One can clearly see that the BC algorithm outperforms other methods both in terms of the computation time and the quality of results. The algorithm is deterministic and is capable to find all Pareto optimal solutions. The BC method solves the problem for all $p \leq 20$ in approximately 1 min. The ES method is very slow. It permits solving the optimization problem for $p = 5$ in approximately 50 min.

The EA algorithm works better than the ES method. The computation times for $p \geq 4$ are significantly smaller than for the ES approach. The EA algorithm is, however, orders of magnitude slower that the BC algorithm. The RL algorithm is faster that the EA approach. For example, the computation time to find the complete Pareto front for $p = 9$ using the RL algorithm is more than ten times shorter than for the EA approach. However, it is still considerably slower than the BC algorithm. The main disadvantage of the EA and RL algorithms lies in their heuristic nature—the quality of the results depends on the number of tested solutions. The results returned by the EA and RL algorithm are correct for $p \leq 14$ and $p \leq 15$, respectively. Computation times needed to find all Pareto optimal solutions for $p = 14$ (EA) and $p = 15$ (RL) exceed 10 h.

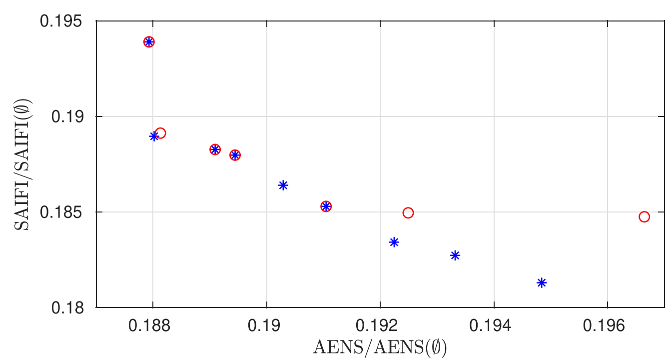

Fig. 3. Multi objective optimization for the IEEE 123 node test feeder and $p = 15$ sectionalizing switches; design objectives AENS and SAIFI, the Pareto front found using the BC algorithm is plotted using blue stars, solutions found using the EA algorithm are plotted with red circles.

The Pareto front found for the optimization of AENS and SAIFI with $p = 15$ switches is plotted in Fig. 3 using blue stars. The solution set found using the EA algorithm is plotted with red circles. Note that the number of Pareto optimal solutions is 9 and the EA algorithm finds only four of them.

Note that the best solution in terms of the AENS objective function (shown in the upper left corner of the plot) reduces the value of AENS below 18.8% of its initial value without any protective devices. The best solution in terms of SAIFI (lower right corner of the plot) reduces its value to approximately 18.1% of its initial value without any protective devices. However, for this solution, the value of AENS is approximately 4% worse than for the best solution in terms of AENS.

Let us now compare memory requirements of the algorithms. Each algorithm uses procedures for the evaluation of all objective functions for a given selection of positions of protective devices. For the evaluation of objective functions one needs to store for each node the list of children and the following values defined in Sections II-B and III: N_k, P_k, λ_k, t_k, $\bar{P}_k$, $\bar{N}_k$, $\bar{t}_k$, and $\bar{\lambda}_k$. During its run, each algorithm stores the current set of nondominated solutions, which at the end of the algorithm become a solution found by the algorithm. In the computational examples considered, the size of the set of nondominated solutions is always a small number not exceeding 30. In case of ES and RL algorithms with multiple runs, the final set of solutions has to be stored for each run.

Let us now study additional memory requirements of the algorithms. The ES algorithm is implemented by performing recursive calls, with the number of levels equal to the number of protective devices (in each call one protective device is added). It does not require any additional data to be stored. It follows that additional memory usage for this algorithm is very low. Memory needed by the EA algorithm depends on the size of generation. This becomes an important component for large generation sizes ($g_s > 1000$). For the BC algorithm one needs to store for each node the set of partial solutions which cannot be eliminated at this node. Without using the elimination procedure this number grows exponentially when the algorithm moves up the tree structure. However, with the elimination procedure described in Section III-A, the number of partial solutions remains small. For the IEEE 123 network and $p \leq 20$ protective devices (see Table I), the maximum number of partial solutions stored at a given node grows with the number of protective devices and in all cases is below 200. Computational examples show that memory usage for the BC algorithm is less demanding than for the EA algorithm with the generation size $g_s > 500$. The RL algorithm needs to store the policy for each node. The memory needed is proportional to the number of nodes. It follows that additional memory usage for this algorithm is very low.

Summarizing, memory usage is low for all algorithms. The least demanding memory requirements are for the ES and RL approaches. For large generations the EA algorithm is the most demanding in terms of memory usage.

B. Optimization Results for the IEEE 8500 Node Test Feeder

As a second example, the IEEE 8500 node test feeder is considered. Optimization results obtained using different methods are reported in Table II. The BC algorithm finds complete Pareto fronts for all $p \leq p_{\max} = 15$. The computation time grows with p. However, even for $p = 15$ it does not exceed 1 h. The ES algorithm works only for $p \leq 2$. This is due to very long computation time needed to complete the algorithm for larger p. For example, for $p = 3$, the ES algorithm requires more than 90 h of computation time to consider all $N \approx 1.89 \cdot 10^9$ test cases. The EA algorithm produces correct results for $p \leq 4$. It is not capable of finding any Pareto optimal solution, i.e., $R = 0$ for $p \geq 5$ in spite of long computation times exceeding 10 h for each p.

TABLE II
PERFORMANCE OF MULTIOBJECTIVE OPTIMIZATION ALGORITHMS FOR THE IEEE 8500 NODE TEST FEEDER

		BC		ES		EA			
p	S	N	t[s]	N	t[s]	N	t[s]	S'	R
1	**1**	11064	0.18	2249	0.40	12700	5.61	1	**1**
2	**1**	35899	1.32	$2.5 \cdot 10^6$	433.61	$2.3 \cdot 10^5$	97.37	1	**1**
3	**1**	84027	4.93			$1.8 \cdot 10^6$	1032.83	1	**1**
4	**1**	$1.6 \cdot 10^5$	17.71			$2.4 \cdot 10^7$	10343.45	1	**1**
5	**2**	$3.0 \cdot 10^5$	45.30			$1.0 \cdot 10^8$	43722.72	3	0
6	**4**	$5.3 \cdot 10^5$	74.65			$1.0 \cdot 10^8$	43504.45	9	0
7	**5**	$8.8 \cdot 10^5$	140.07			$1.0 \cdot 10^8$	44448.56	3	0
8	**8**	$1.4 \cdot 10^6$	209.82			$1.0 \cdot 10^8$	44511.37	8	0
9	**5**	$2.0 \cdot 10^6$	283.47			$1.0 \cdot 10^8$	44937.23	8	0
10	**10**	$2.9 \cdot 10^6$	421.36			$1.0 \cdot 10^8$	43155.97	11	0
11	**10**	$4.0 \cdot 10^6$	721.82			$1.0 \cdot 10^8$	45464.36	3	0
12	**37**	$5.2 \cdot 10^6$	943.34			$1.0 \cdot 10^8$	45527.24	5	0
13	**61**	$6.5 \cdot 10^6$	1387.28			$1.0 \cdot 10^8$	46438.63	10	0
14	**40**	$8.0 \cdot 10^6$	2129.05			$1.0 \cdot 10^8$	47844.22	4	0
15	**39**	$9.7 \cdot 10^6$	3573.59			$1.0 \cdot 10^8$	45460.57	2	0

V. CONCLUSION

The BC optimization algorithm for protective device allocation in single-feeder power distribution networks with a radial topology to simultaneously minimize several objective functions has been proposed. The method is deterministic and is guaranteed to solve the problem under study, i.e., find all nondominated solutions. Possibilities of generalizing the proposed algorithm for other network topologies will be a subject of future research.

The proposed algorithm has been tested using distribution networks of medium and very large sizes. It has been shown that the BC algorithm can successfully handle very large power distribution systems with several thousands of nodes. A comparison with existing techniques has been carried out.

The ES approach finds the complete Pareto front by scanning the whole search space. It may be used for small networks and low number of protective devices only. The solutions found by the BC and ES methods are the same in all the cases for which the ES algorithm solves the problem. This confirms the correctness of the BC algorithm.

The EA and the RL approach are faster than the ES method. These two algorithms are heuristic and for more complex problems fail to find Pareto optimal solutions. The computation times for the EA and RL methods are orders of magnitude larger than for the BC algorithm. For the medium size network considered (123 nodes) and $p = 10$ protective devices, the computation times to find the complete Pareto front exceed 10 h, while the BC algorithm needs 1 s to solve this problem. For the large network with 2250 nodes, the EA and RL methods fail to find any Pareto optimal solution for $p > 4$ in spite of computation times exceeding 10 h. The BC algorithm solves the problem in 400 s for $p = 10$ and less than 1 h for $p = 15$. These results confirm that the proposed method outperforms other tested methods in terms of the computation time and the quality of results.

From the results obtained it follows that solutions found in single-objective optimizations of SAIDI, SAIFI, and AENS are not necessarily the same, i.e., a solution optimizing one of the objective functions is usually not optimal for other objectives. The proposed algorithm permits finding the complete set of Pareto optimal solutions, which provides options for a system designer to choose the localization of protective devices which satisfies the reliability requirements and minimizes investment and maintenance costs as well as user interruption costs.

References

[1] A. Zidan et al., "Fault detection, isolation, and service restoration in distribution systems: State-of-the-art and future trends," *IEEE Trans. Smart Grid*, vol. 8, no. 5, pp. 2170–2185, Sep. 2017.

[2] N. Gholizadeh, S. Hosseinian, M. Abedi, H. Nafisi, and P. Siano, "Optimal placement of fuses and switches in active distribution networks using value-based MINLP," *Rel. Eng. Syst. Saf.*, vol. 217, 2022, Art. no. 108075.

[3] P. Salyani, R. Nourollahi, K. Zare, and R. Razzaghi, "A new MILP model of switch placement in distribution networks with consideration of substation overloading during load transfer," *Sustain. Energy, Grids Netw.*, vol. 32, 2022, Art. no. 100944.

[4] M. Zare-Bahramabadim, H. Farzin, and M. Ehsan, "An MILP model for risk-based placement of RCSS, DDGS, and tie lines in distribution systems with complex topologies," *IET Renewable Power Gener.*, vol. 17, no. 4, pp. 829–839, 2023.

[5] G. Levitin and S. Mazal-Tov, "Optimal sectionalizer allocation in electric distribution systems by genetic algorithm," *Electric Power Syst. Res.*, vol. 35, no. 3, pp. 149–155, 1995.

[6] R. Billinton and S. Jonnavithula, "Optimal switching device placement in radial distribution systems," *IEEE Trans. Power Del.*, vol. 11, no. 3, pp. 1646–1651, Jul. 1996.

[7] A. Abiri-Jahromi, M. Fotuhi-Firuzabad, M. Parvania, and M. Mosleh, "Optimized sectionalizing switch placement strategy in distribution systems," *IEEE Trans. Power Del.*, vol. 27, no. 1, pp. 362–370, Jan. 2012.

[8] A. Heidari, V. G. Agelidis, and M. Kia, "Considerations of sectionalizing switches in distribution networks with distributed generation," *IEEE Trans. Power Del.*, vol. 30, no. 3, pp. 1401–1409, Jun. 2015.

[9] C. Su and C. Lee, "Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution," *IEEE Trans. Power Del.*, vol. 18, no. 3, pp. 1022–1027, Jul. 2003.

[10] S. Golestani and M. Tadayon, "Optimal switch placement in distribution power system using linear fragmented particle swarm optimization algorithm preprocessed by GA," in *Proc. 18th Int. Conf. Eur. Energy Market*, Zagreb, Croatia, 2011, pp. 537–542.

[11] H. Falaghi, M. R. Haghifam, and C. Singh, "Ant colony optimization-based method for placement of sectionalizing switches in distribution networks using a fuzzy multi objective approach," *IEEE Trans. Power Del.*, vol. 24, no. 1, pp. 268–276, Jan. 2009.

[12] A. Esteban and A. Alberto, "Optimal selection and allocation of sectionalizers in distribution systems using fuzzy dynamic programming," *Energy Power Eng.*, vol. 2, pp. 283–290, 2010.

[13] H. N. Alves, "A hybrid algorithm for optimal placement of switches devices in electric distribution systems," *IEEE Latin America Trans.*, vol. 10, no. 6, pp. 2218–2223, Dec. 2012.

[14] Y. Mao and K. N. Miu, "Switch placement to improve system reliability for radial distribution systems with distributed generation," *IEEE Trans. Power Syst.*, vol. 18, no. 4, pp. 1346–1352, Nov. 2003.

[15] L. Sun et al., "Network partitioning strategy for parallel power system restoration," *IET Gener. Transm. Distrib.*, vol. 10, no. 8, pp. 1883–1892, 2016.

[16] W. Bachorek and M. Benesz, "Analysis of sectionalizing switch placement in medium voltage distribution networks in the aspect of improving the continuity of power supply," *Bull. Polish Acad. Sci. Tech. Sci.*, vol. 68, no. 3, pp. 459–466, 2020.

[17] F. Soudi and K. Tomsovic, "Optimized distribution protection using binary programming," *IEEE Trans. Power Del.*, vol. 13, no. 1, pp. 218–224, Jan. 1998.

[18] M. Farajollahi, M. Fotuhi-Firuzabad, and A. Safdarian, "Optimal placement of sectionalizing switch considering switch malfunction probability," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 403–413, Jan. 2019.

[19] Z. Popovic, B. Brbaklic, and S. Knezevic, "A mixed integer linear programming based approach for optimal placement of different types of automation devices in distribution networks," *Electric Power Syst. Res.*, vol. 148, pp. 136–146, 2017.

[20] A. Heidari, Z. Y. Dong, D. Zhang, P. Siano, and J. Aghaei, "Mixed-integer nonlinear programming formulation for distribution networks reliability optimization," *IEEE Trans. Ind. Inform.*, vol. 14, no. 5, pp. 1952–1961, May 2018.

[21] Z. Galias, "Tree-structure based deterministic algorithms for optimal switch placement in radial distribution networks," *IEEE Trans. Power System*, vol. 34, no. 6, pp. 4269–4278, Nov. 2019.

[22] J. R. Bezerra, G. C. Barroso, R. P. S. Leao, and R. F. Sampaio, "Multiobjective optimization algorithm for switch placement in radial power distribution networks," *IEEE Trans. Power Del.*, vol. 30, no. 2, pp. 545–552, Apr. 2015.

[23] S. Ray, A. Bhattacharya, and S. Bhattacharjee, "Optimal placement of switches in a radial distribution network for reliability improvement," *Int. J. Elect. Power Energy Syst.*, vol. 76, pp. 53–68, 2016.

[24] N. Sahoo, S. Ganguly, and D. Das, "Multi-objective planning of electrical distribution systems incorporating sectionalizing switches and tie-lines using particle swarm optimization," *Swarm Evol. Computation*, vol. 3, pp. 15–32, 2012.

[25] A. V. Pombo, J. Murta-Pina, and V. F. Pires, "Multiobjective planning of distribution networks incorporating switches and protective devices using a memetic optimization," *Rel. Eng. Syst. Saf.*, vol. 136, pp. 101–108, 2015.

[26] A. V. Pombo, J. Murta-Pina, and V. F. Pires, "A multiobjective placement of switching devices in distribution networks incorporating distributed energy resources," *Electric Power Syst. Res.*, vol. 130, pp. 34–45, 2016.

[27] Z. Galias, "On multiobjective optimization of switch allocation in radial distribution networks," in *Proc. IEEE Innov. Smart Grid Technol.–Asia*, 2019, pp. 563–567.

[28] M. K. Madhav and K. B. Yadav, "Economic analysis of isolator placement in the radial network for enhancement of reliability indicators," *Elect. Eng.*, vol. 106, pp. 891–904, 2024.

[29] C. B. Martínez-Anidoa, R. Boladoa, L. D. Vriesb, G. Fulli, M. Vandenbergha, and M. Maseraa, "European power grid reliability indicators, what do they really tell?," *Electric Power Syst. Res.*, vol. 90, pp. 79–84, 2012.

[30] J. Zhou et al., "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[31] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *Comput. Operations Res.*, vol. 134, 2021, Art. no. 105400.

[32] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 3, pp. 385–398, Mar. 2015.

[33] M. Bazrafshan and N. Gatsis, "Comprehensive modeling of three-phase distribution systems via the bus admittance matrix," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 2015–2029, Mar. 2018.

[34] A. Stanisavljević, V. Katic, B. Dumnić, and B. Popadic, "A brief overview of the distribution test grids with a distributed generation inclusion case study," *Serbian J. Elect. Eng.*, vol. 15, pp. 115–129, 2018.

[35] R. R. Jha, A. Dubey, C.-C. Liu, and K. P. Schneider, "Bi-level volt-VAR optimization to coordinate smart inverters with voltage control devices," *IEEE Trans. Power Syst.*, vol. 34, no. 3, pp. 1801–1813, May 2019.

[36] X. Chen et al., "Full coverage of optimal phasor measurement unit placement solutions in distribution systems using integer linear programming," *Energies*, vol. 12, no. 8, 2019, Art. no. 1552.

[37] R. F. Arritt and R. C. Dugan, "The IEEE 8500-node test feeder," in *Proc. IEEE Transmiss. Distrib. Conf. Expo.*, 2010, pp. 1–6.

[38] K. P. Schneider et al., "Analytic considerations and design basis for the IEEE distribution test feeders," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3181–3188, May 2018.

Zbigniew Galias (Senior Member, IEEE) received the M.Sc. degree in electronics from the AGH University of Krakow, Kraków, Poland, in 1990, the M.Sc. degree in mathematics from the Jagiellonian University, Kraków, Poland, in 1992, and the Ph.D. and the senior doctorate degrees in electrical engineering from the AGH University of Krakow, Kraków, Poland, in 1996 and 2004.

In 2015, he received the title of Professor. He is currently a Professor with the Department of Electrical Engineering and Power Engineering, AGH University of Krakow. His research interests include complex and nonlinear circuits and systems, electrical networks, chaos, sliding mode control, interval arithmetic and computer assisted proofs.

Dr. Galias served/is serving as an Associate Editor for IEEE Transactions on Circuits and Systems I and II, *IEEE Circuits and Systems Magazine*, and the *International Journal of Bifurcation and Chaos*.