Medium-Voltage Distribution Network Expansion Planning with Gene-pool Optimal Mixing Evolutionary Algorithms

Hoang N. Luong[1], Marinus O.W. Grond[2], Peter A.N. Bosman[1], and Han La Poutré[1]

[1] Center for Mathematics and Computer Science (CWI)
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
[2] Eindhoven University of Technology (TU/e)
Department of Electrical Engineering, Electrical Energy Systems
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

Abstract. Medium-voltage distribution network expansion planning involves finding the most economical adjustments of both the capacity and the topology of the network such that no operational constraints are violated and the expected loads, that the expansion is planned for, can be supplied. This paper tackles this important real-world problem using realistic yet computationally feasible models and, for the first time, using two instances of the recently proposed class of Gene-pool Optimal Mixing Evolutionary Algorithms (GOMEAs) that have previously been shown to be a highly efficient integration of local search and genetic recombination, but only on standard benchmark problems. One GOMEA instance that we use employs linkage learning and one instance assumes no dependencies among problem variables. We also conduct experiments with a widely used traditional Genetic Algorithm (GA). Our results show that the favorable performance of GOMEA instances over traditional GAs extends to the real-world problem at hand. Moreover, the use of linkage learning is shown to further increase the algorithm's effectiveness in converging toward optimal solutions.

Keywords: Evolutionary Algorithms, Linkage Learning, Distribution Network, Power System Expansion Planning

1 Introduction

The Gene-pool Optimal Mixing Evolutionary Algorithm (GOMEA) combines genetic recombination as is reminiscent of Genetic Algorithms (GAs) with model-building as is reminiscent of Estimation of Distribution Algorithm (EDAs) and direct improvements as is reminiscent of Local Search (LS) [1]. The model used in GOMEA describes linkage relations between variables, i.e. which variables should be copied jointly when performing genetic recombination. Various subclasses of the general linkage model are possible, ranging from allowing only fully independent linkage relations to allowing overlapping linkage relations. Based on

the chosen and then learned linkage structure, GOMEA performs variation by intensively mixing building blocks as identified by the linkage relations in a greedy manner. The efficiency of GOMEA has so far been shown on a number of academic benchmarks [1, 2], but not yet on real-world optimization problems.

A medium-voltage (MV) distribution network carries electricity from the (sub)transmission network to MV consuming units [3]. MV distribution network expansion planning (DNEP) is an important real-world engineering problem. As the loads (i.e. power consumptions) at different locations increase and/or newly appeared loads need connections to the network, various electrical components in the distribution network require replacement or new components must be installed. Both the capacities of the components and the topology of the network have to be taken into account. There exist various MV network layouts but the two most common topologies are: radial topology and open loop topology [3]. Radial topologies, in which every consuming unit is supplied by only one electrical feed path, are often used in distribution networks with overhead lines, especially for rural areas [3]. This paper focuses on the open loop layout, which is used for distribution networks with underground cables, typically found in urban areas of dense populations. Such MV networks contain groups of several consuming units (load points). In each group, consuming units are physically connected one by one by cables forming the shape of a loop. However, in normal operation, due to management and protection policies, one cable of every loop is put into an inactive state which creates an opening in the loop so that the network operates in a radial manner. Those cables are put in reserve to be used for reconfiguring the MV network when unexpected faults happen on active cables [3]. A feasible expansion plan is one that satisfies all operation and configuration constraints. An optimal plan is one that is feasible and has minimum expansion costs. In this paper, investment expenses are of sole interest.

There exist numerous studies into DNEP but the problem modelling is still far from being standardized. Every network operator has a different policy regarding the operation constraints of their power systems and different repositories of electrical facilities. Most studies evaluate the reliability of distribution networks based on the average failure rates and restoration times of components, in which reserve cables are considered as options to enhance the network reliability [4, 5]. The result of such reliability analysis can then be capitalized into customer outage cost to include in the overall cost to be optimized [4] or can be treated as a separate objective function [5]. However, it is shown that reliability in practice is a relative index as its calculation involves many intricate problems with high uncertainty [6]. In this paper, we therefore consider the capacity of reserve cables, from a different and more practically relevant perspective, as a network configuration constraint, which is termed as reconfigurability. Reconfigurability requires the network to have enough reserve cables with adequate capacities to bring the network back to operation when an outage happens on some active cable. Although the cost function to be optimized is relatively simple and the problem variables are even pairwise independent in it, the constraint functions are far more involved and require dedicated electrical

engineering computations (e.g. power flow calculations) that involve the entire network, effectively introducing dependencies between problem variables. It is therefore interesting, in addition to comparing the effectiveness of GOMEA with the commonly employed traditional GA, to see whether the use and usefulness of linkage learning also extends from traditional benchmark problems to real-world problems such as the one at hand.

The remainder of this paper is organized as follows. Section 2 outlines GOMEA and explains its components. Section 3 presents the anatomy of a conventional distribution network and important constraints. Section 4 shows and discusses the experimental results, while Section 5 concludes the paper.

2   Gene-pool Optimal Mixing Evolutionary Algorithm

Classic GAs have difficulty solving an optimization problem that has optimal solutions made of multivariate building blocks whose constitutive problem variables are scattered over the solution representation string [7]. Traditional recombination operators of GAs are either not able to juxtapose building blocks of nonconsecutive variables (i.e. 1- or 2-point, or uniform crossover) or too disruptive to preserve enough long building blocks (i.e. in case of uniform crossover). EDAs were developed with an emphasis on linkage learning to help to detect and preserve multivariate dependencies, but in EDAs this comes at the cost of estimating the complete probability distribution, which is expensive and may be unnecessary. On the other hand, problems with hierarchical dependencies provide a huge challenge for a classic GA as its genetic recombination is only horizontal and hierarchical dependencies (i.e. building blocks of building blocks) cannot be exploited directly. The reason for this is that there is no intermediate checking for improvements during genetic recombination, causing higher-level building blocks to automatically overwrite and undo the effects of mixing lower-level building blocks. GOMEA overcomes these issues by effectively integrating local search into variation, making its overall procedure closer to that of genetic local search [8]. For solving DNEP, GOMEA is therefore a strong candidate optimization algorithm.

2.1   Family of Subsets

The GOMEA uses the concept of family of subsets (FOS) as the linkage model to match the structure of optimization problems [1]. A FOS, denoted $\mathcal{F}$, is a set of subsets of a certain set S, which means $\mathcal{F} \subset \mathcal{P}(S)$, i.e. the powerset of S. Normally, set S is the set of all variable indices $\{1, 2, \ldots, l\}$. A FOS $\mathcal{F}$ can be written as $\mathcal{F} = \{\boldsymbol{F}^1, \boldsymbol{F}^2, \ldots, \boldsymbol{F}^{|\mathcal{F}|}\}$ where $\boldsymbol{F}^i \subseteq \{1, 2, \ldots, l\}$, $i \in \{1, 2, \ldots, |\mathcal{F}|\}$. To ensure all decision variables are considered in the variation operator, every variable index is contained in at least one subset in $\mathcal{F}$, i.e. $\forall i \in \{1, 2, \ldots, l\}$: $(\exists j \in \{1, 2, \ldots, |\mathcal{F}|\} : i \in \boldsymbol{F}^j)$. In this paper, we consider two FOS structures.

Univariate Structure: This structure, which is arguably the simplest structure possible, considers every decision variable to be independent from each

other. The corresponding FOS $\mathcal{F}$ thus contains only singleton subsets $\boldsymbol{F}^i = \{i\}, i \in \{1, 2, \ldots, l\}$. As there is only one possible configuration, no linkage learning is required. The use of the univariate structure is perhaps best known from GAs, where it translates into the well-known uniform crossover operator (UX).

Linkage Tree Structure: The linkage tree (LT) structure represents dependencies among decision variables in a hierarchical manner. The bottom level of the tree (i.e. leaf nodes) contains all singleton subsets, i.e. the univariate structure. Intermediate levels contains subsets $\boldsymbol{F}^i$ having more than one decision variable index. Any bivariate or multivariate subset $\boldsymbol{F}^i$ is the result of combining two subsets $\boldsymbol{F}^j$ and $\boldsymbol{F}^k$ such that $\boldsymbol{F}^j \cap \boldsymbol{F}^k = \emptyset, |\boldsymbol{F}^j| < |\boldsymbol{F}^i|, |\boldsymbol{F}^k| < |\boldsymbol{F}^i|$ and $\boldsymbol{F}^j \cup \boldsymbol{F}^k = \boldsymbol{F}^i$. The top level (root node) is the set S itself containing all decision variable indices. This root node, which indicates that all variables are jointly dependent, is excluded from the linkage tree FOS as performing building block mixing based on this subset for any two solutions only results in the same solutions.

The LT is learned from the selected candidate solutions at every generation by performing a hierarchical clustering procedure where distances between clusters are computed using the average pair-wise distance over all pairs of variables. For details about clustering algorithms and different distance metrics, please refer to the literature [1, 2]. Here, we used mutual information (MI) as the basis of distance between two variables (higher MI values mean a lower distance). We further note that in this paper, variables are not binary but rather have a larger bounded integer domain. However, since the search space is still Cartesian, the extension of MI from binary to integer variables is straightforward. The GOMEA variant that uses the LT structure as its linkage model is also known as Linkage Tree Genetic Algorithm (LTGA)[1]. It is worthwhile to mention that the computational complexity of learning an LT is low compared to typical higher-order models in EDAs (i.e. $O(nl^2)$ versus $O(nl^3)$).

2.2   Optimal Mixing and Forced Improvements

GOMEA uses a procedure called Gene-pool Optimal Mixing (GOM) as its variation operator [1]. For each existing parent solution in the population, exactly one offspring is generated by mixing building blocks of that parent with those of other solutions following the linkages specified by subsets in FOS $\mathcal{F}$. First, the parent solution is cloned. Then, the FOS is traversed and for each subset $\boldsymbol{F}^i \in \mathcal{F}$ a donor solution is chosen randomly from the population. The values in the donor corresponding to the variables in the linkage group are copied into the parent solution. If such mixing results in an improvement, the changes are accepted, otherwise the changes are reverted. Bosman et al. [2] showed that if GOM also accepts changes that generate equally good solutions, better performance can be achieved.

If a solution cannot be improved by GOM alone, a procedure called forced improvement (FI) is performed [9]. In essence, FI is an additional GOM operation with the current best solution always as the donor. However, in this case, optimal mixing stops as soon as any single improvement is achieved. Because

```
GOMEA //population size n
```
$\mathbf{for}\ i \in \{1, 2, \ldots, n\}\ \mathbf{do}$
    $\mathcal{P}_i \leftarrow \textsc{CreateRandomSolution}()$
    $\textsc{EvaluateFitness}(\mathcal{P}_i)$
$\boldsymbol{x}^{best} \leftarrow argmax_{\boldsymbol{x} \in \mathcal{P}}\{fitness[\boldsymbol{x}]\}$
$t \leftarrow 0; t^{NIS} \leftarrow 0$
$\mathbf{while}\ \neg\textsc{TerminationConditionsSatisfied}\ \mathbf{do}$
    $\mathcal{S} \leftarrow \textsc{TournamentSelection}(\mathcal{P}, n, 2)$
    $\textsc{LearnModel}(\mathcal{S})$
    $\mathbf{for}\ i \in \{1, 2, \ldots, n\}\ \mathbf{do}$
        $\mathcal{O}_i \leftarrow \text{FI-GOM}(\mathcal{P}_i)$
    $\mathcal{P} \leftarrow \mathcal{O}$
    $\boldsymbol{x}^{best} \leftarrow argmax_{\boldsymbol{x} \in \mathcal{P}}\{fitness[\boldsymbol{x}]\}$
    $\mathbf{if}\ fitness[\boldsymbol{x}^{best}(t)] > fitness[\boldsymbol{x}^{best}]\ \mathbf{then}$
        $t^{NIS} \leftarrow 0; \boldsymbol{x}^{best} \leftarrow \boldsymbol{x}^{best}(t)$
    $\mathbf{else}$
        $t^{NIS} \leftarrow t^{NIS} + 1$
    $t \leftarrow t + 1$

```
FI-GOM(x)
```
$\boldsymbol{b} \leftarrow \boldsymbol{o} \leftarrow \boldsymbol{x}; fitness[\boldsymbol{b}] \leftarrow fitness[\boldsymbol{o}] \leftarrow fitness[\boldsymbol{x}]; changed \leftarrow false$
$\mathbf{for}\ i \in \{1, 2, \ldots, |\mathcal{F}|\}\ \mathbf{do}$
    $\boldsymbol{p} \leftarrow \textsc{Random}(\{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\})$
    $\boldsymbol{o}_{\boldsymbol{F}i} \leftarrow \boldsymbol{p}_{\boldsymbol{F}i}$
    $\mathbf{if}\ \boldsymbol{o}_{\boldsymbol{F}i} \neq \boldsymbol{b}_{\boldsymbol{F}i}\ \mathbf{then}$
        $\textsc{EvaluateFitness}(\boldsymbol{o})$
        $\mathbf{if}\ fitness[\boldsymbol{o}] \geq fitness[\boldsymbol{b}]\ \mathbf{then}$
            $\boldsymbol{b}_{\boldsymbol{F}i} \leftarrow \boldsymbol{o}_{\boldsymbol{F}i}; fitness[\boldsymbol{b}] \leftarrow fitness[\boldsymbol{o}]; changed \leftarrow true$
        $\mathbf{else}$
            $\boldsymbol{o}_{\boldsymbol{F}i} \leftarrow \boldsymbol{b}_{\boldsymbol{F}i}; fitness[\boldsymbol{o}] \leftarrow fitness[\boldsymbol{b}]$
$\mathbf{if}\ \neg changed\ \text{or}\ t^{NIS} > 1 + \lfloor \log_{10}(n) \rfloor\ \mathbf{then}$
    $changed \leftarrow false$
    $\mathbf{for}\ i \in \{1, 2, \ldots, |\mathcal{F}|\}\ \mathbf{do}$
        $\boldsymbol{o}_{\boldsymbol{F}i} \leftarrow \boldsymbol{x}^{best}_{\boldsymbol{F}i}$
        $\mathbf{if}\ \boldsymbol{o}_{\boldsymbol{F}i} \neq \boldsymbol{b}_{\boldsymbol{F}i}\ \mathbf{then}$
            $\textsc{EvaluateFitness}(\boldsymbol{o})$
            $\mathbf{if}\ fitness[\boldsymbol{o}] > fitness[\boldsymbol{b}]\ \mathbf{then}$
                $\boldsymbol{b}_{\boldsymbol{F}i} \leftarrow \boldsymbol{o}_{\boldsymbol{F}i}; fitness[\boldsymbol{b}] \leftarrow fitness[\boldsymbol{o}]; changed \leftarrow true$
            $\mathbf{else}$
                $\boldsymbol{o}_{\boldsymbol{F}i} \leftarrow \boldsymbol{b}_{\boldsymbol{F}i}; fitness[\boldsymbol{o}] \leftarrow fitness[\boldsymbol{b}]$
            $\mathbf{if}\ changed\ \mathbf{then\ breakfor}$
$\mathbf{if}\ \neg improved\ \mathbf{then}$
    $\boldsymbol{o} \leftarrow \boldsymbol{x}^{best}; fitness[\boldsymbol{o}] \leftarrow fitness[\boldsymbol{x}^{best}]$

Fig. 1: Pseudo-code for GOMEA [2]

accepting solutions of equal quality can potentially stall the algorithm indefinitely on a fitness plateau, GOMEA is found to have better performance if FI is also triggered when the number of continuous generations that the best solution is not updated, which is termed as no-improvement stretch (NIS), is larger than $1 + \lfloor \log_{10}(n) \rfloor$ [2]. FI is reported to ensure efficient convergence while not continuously reducing population diversity [9]. The pseudo-code for GOMEA with GOM and FI is outlined in Figure 1. Note that GOMEA typically does a lot more evaluations per generation than a classic GA would do, but GOMEA also typically requires far smaller population sizes and far less generations to converge.

3   MV Distribution Network Expansion Planning

Distribution network expansion planning (DNEP) involves decision making about what, where, when and how electrical components in a power distribution sys-

tem should be adjusted to meet the forecasted growth in power demands at consuming units. In this paper, we take a traditional conservative approach and consider only the highest possible peak load for each consuming unit in the network. The network must be configured such that it can handle those loads and thus it is tested with that load profile. In this paper, we focus on a key part of the problem: deciding upon the locations and the types of adjustments. Available enhancement options are: changing existing devices, and installing new devices in the network, without specifying the time horizon. This paper considers two kinds of electrical devices: cables and transformers. An optimal expansion plan requires minimum investment cost while satisfying all operation and configuration constraints (see Section 3.2).

3.1   MV Distribution Network Encoding

An MV distribution network can be seen as a graph with a set of nodes (vertices) and a set of branches (edges). A node can be a substation, which is the source of power supply, or it can be a consuming unit, which demands and consumes power. Every branch connects two nodes, and all branches together form feed paths for electric currents flowing from power substations to consuming units. In a DNEP problem, the power supply capacities of substations and power demands of consuming units form the inputs. The outputs are decisions about capacities of all branches. Available options are: whether to connect two nodes by a branch (an overhead line or an underground cable, or a transformer if two nodes have different voltages), the capacity of the branch, and whether the branch should be active or in reserve.

   To solve the DNEP for a network, we need to specify all the currently existing branches and a restricted set of potential candidate branches that can be newly added into the network. This set of potential branches is often determined by using expert knowledge to disregard unnecessary branches. Let l denote the total number of branches, and let m denote the total number of nodes. We represent a distribution network as a vector of length l of integer-value elements

$$\mathbf{x} = (x_1, x_2, \ldots, x_l), \quad x_i \in \Omega(x_i), \quad i \in \{1, 2, \ldots, l\} \tag{1}$$

where each x_i corresponds with the i-th branch of the network. The set of possible devices $\Omega(x_i)$ that can be installed at each branch x_i depends on policies and the repository of each network operator. We use an integer number to indicate which device to install at a branch. The status of each x_i is defined as follows

- $x_i = 0$: There is no device at the i-th branch. This means that the previously existing device is removed or that no device is decided to be installed at the i-th branch.
- $x_i = id > 0$: A device with identification number $id \in \Omega(x_i)$ is installed at the i-th branch.
- $x_i = -id < 0$: A device with identification number $id \in \Omega(x_i)$ is put in reserve at the i-th branch. The device is installed into the network but it does not take part in the normal operation. It is used to reconfigure the system in emergency cases.

Note that the original MV network has the x_i of currently non-existing branches set to 0.

3.2   Optimization Problem Formulation

Let $\mathbf{x} = (x_1, x_2, \ldots, x_l)$ be the original network and let $\mathbf{x}' = (x'_1, x'_2, \ldots, x'_l)$ be an adjusted network. DNEP minimizes the investment cost as follows

$$\textbf{Min} \quad f(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{l} cost(x_i, x'_i) \tag{2}$$

where

$$cost(x_i, x'_i) = \begin{cases} 0 & \text{if } x_i = x'_i \\ \text{cost of changing } x_i \text{ to } x'_i & \text{if } x_i \neq x'_i \end{cases} \tag{3}$$

For a given (test) load profile, the following constraints must be satisfied:

I **Voltage constraints**

$$|V_i|^{MIN} \leq |V_i| \leq |V_i|^{MAX}, \quad i \in \{1, 2, \ldots, m\} \tag{4}$$

where $|V_i|$ is the voltage magnitude at node i, and $[|V_i|^{MIN}, |V_i|^{MAX}]$ is the allowable range of voltage magnitude at node i. We quantify the degree of the voltage constraint violation of a network by summing the amount of out-of-bound voltage magnitude at every node (i.e. $(|V_i|^{MIN} - |V_i|)$ if $|V_i| < |V_i|^{MIN}$ or $(|V_i| - |V_i|^{MAX})$ if $|V_i| > |V_i|^{MAX}$).

II **Line flow constraints (or device capacity constraints)**

$$|S_i| \leq |S_i|^{MAX}, \quad i \in \{1, 2, \ldots, l\} \tag{5}$$

where $|S_i|$ is the power flow through the device installed at branch x_i, i.e. a cable or a transformer, and $|S_i|^{MAX}$ is the nominal capacity of that device. There should be no overload at any device. We quantify the degree of the line flow constraint violation of a network by summing the amount of overload at every branch (i.e. $(|S_i| - |S_i|^{MAX})$ if $|S_i| > |S_i|^{MAX}$).

III **Radial operation constraint**: All the active cables together have to form a radial configuration. This means that any consuming unit is supplied electricity via one single feed path in normal operation.

IV **Reconfigurability constraint**: When, during normal operation, faults happen on an active branch, that branch is isolated from the network by opening its corresponding switches. The network is then reconfigured by closing the switches of reserve branches so that disconnected consuming units are served again. The network may operate with loops in an emergency situation and can endure a mild overload in a short time while the faulty branch is being repaired. The degrees of emergency capacity of equipments are decided by network operators. In this paper, we assume that equipment emergency capacity is 120% of its nominal capacity.

Constraints I, II, and III are commonly adopted in the literature [4, 5]. The constraint IV is employed here due to reasons mentioned in Section 1.

3.3   Solution Evaluation

As DNEP is a constrained optimization problem, the fitness evaluation for an expansion plan involves both the investment cost calculation and constraint evaluations. When we need to compare any two solutions, as in selection or the optimal mixing procedures, we use the concept of constraint domination. A feasible solution is one that satisfies all constraints. A feasible solution is always better than an infeasible one, a cheap feasible solution is better than a more expensive one, and if both solutions are infeasible then the one with less or equal degree of violation of all constraints and strictly less violation of at least one constraint is the better solution.

While calculating investment cost is a trivial operation, constraint evaluations are computationally expensive. For each expansion plan, we must perform a power flow calculation (PLC) [10] to obtain the value of the voltage at each node and the power flowing through each branch. These are used to check the constraints (I) and (II). In essence, a PLC involves solving a system of non-linear equations, called the AC power flow model. Due to inherent technical reasons, the commonly used cheaper linear DC model cannot be used for distribution network evaluation without a significant compromise on accuracy. For details of PLC, see e.g. [10]. Therefore, constraints evaluations are computationally expensive.

A complete fulfilment of the reconfigurability constraint requires performing a single-line contingency for every branch in the network: a branch is assumed to be failed, the network is then reconfigured back to operation, and the power flowing in each branch is re-calculated. This paper considers a computationally cheaper constraint evaluation commonly adopted in practice. It performs single line contingency only on cables branching directly from substations as these cables carry the heaviest loads before distributing power to subsequent nodes.

4   Experiments

4.1   Test cases and experiment setup

Based on real-world data, we designed two MV distribution networks as optimization benchmarks.

- **Network 1**: an MV distribution network of one open loop contains 18 nodes (1 substation, 9 consuming units, in which each transformer is represented by 2 nodes having different base voltages) and 25 possible branches (10 existing cables, 8 existing transformers, and 7 potential cable connections). The topology and experiment current and forecasted loads of Network 1 can be found in Fig. 2 and Table 1.
- **Network 2**: an MV distribution network of two open loops contains 31 nodes (1 substation and 30 consuming units) and 59 possible branches (32 existing cables and 27 potential cable connections). Further details are withheld for reasons of confidentiality.

In this paper, we consider 5 common types of MV cables, differentiated by their areas of conductor: 120, 150, 240, 400, and 630 mm^2. We also consider 5 common options of transformers, denoted by their nominal capacities: 100, 160, 250, 400, and 630 kVA.

Table 1: Network 1: current loads and forecasted loads at each consuming unit. P_D and Q_D are the active and reactive power demands, which make up the load at each node. Other nodes have the base voltage of 10 kV and do not have power demand.

Node ID	Base voltage (kV)	Current Load		Forecasted Load	
		P_D	Q_D	P_D	Q_D
3	10	0.6735	0.3951	3.6735	0.3951
11	0.4	0.187	0.1159	0.287	0.1159
12	0.4	0.272	0.1686	0.372	0.1686
13	0.4	0.2818	0.1747	0.2818	0.1747
14	0.4	0.272	0.1747	0.272	0.1686
15	0.4	0.255	0.158	0.355	0.158
16	0.4	0.0808	0.050	0.3808	0.05
17	0.4	0.1785	0.1106	0.2785	0.1106
18	0.4	0.2975	0.1844	0.3975	0.1844

We test 3 optimizers: GOMEA-LT (GOMEA with linkage tree FOS), GOMEA-UNI (GOMEA with univariate FOS), and a traditional genetic algorithm (GA) with uniform crossover and tournament selection similarly configured as in [1]. For every optimizer, we test it with 10 different population sizes which are exponentially increased from 2^1 to 2^{10}. For every population size that we consider, we perform 30 independent runs of each optimizer. Each run starts with a population of randomly generated expansion plans (network topology and the equipment type at each element). We terminate a run only when the whole population converges to the same solution because in practice, the optimum is not know beforehand and we would like to see the best solutions that each optimizer possibly can obtain.

4.2   Results

Fig. 2 shows MV Network 1 before enhancement and the best found expansion plan. To satisfy the forecast load demand, a new cable should connect node 1 (the substation) and node 3. The branch connecting node 2 and 3 should be put in reserve so that the network can operate radially. There are five overloaded transformers, and all of them should be replaced by ones with higher capacities.

Fig. 3 shows the capability of GOMEA-LT, GOMEA-UNI, and GA in minimizing the investment cost for the enhancement of Network 1 as the number of fitness evaluations increases. Fitness evaluations for each candidate expansion plan involve power flow calculations, which are the most computationally expensive operations in the optimization process. Thus, different from academic benchmarks, fitness evaluation for the DNEP problem, truly dominates the computing time of all 3 optimizers. Hence, we use the number of fitness evaluations that each optimizer needs to perform from beginning until convergence as an indicator of computing time. Fig. 3 shows both instances of GOMEA have better performances than the traditional GA. The traditional GA consumes much more
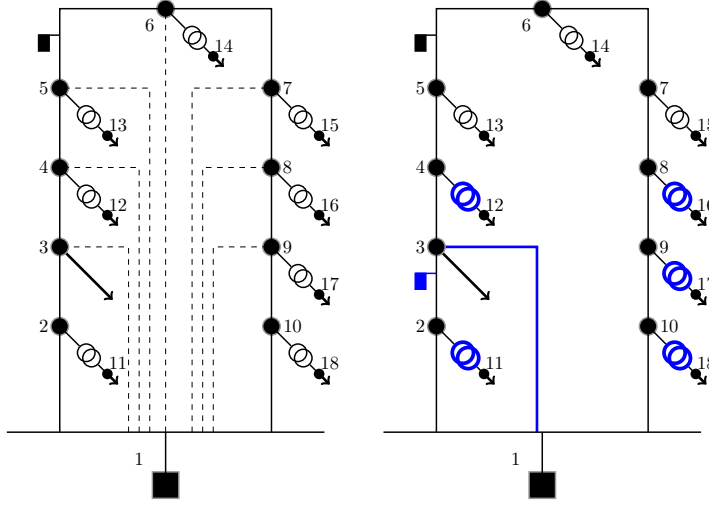
Fig. 2: Network 1. Original topology: Potential cables are represented by dashed lines. Reserve cables are marked with flag symbols. Transformers are denoted by pairs of overlapped circles. Arrow symbols indicate power demands at consuming units. After enhancement: Highlighted components are suggested to be replaced or newly installed.
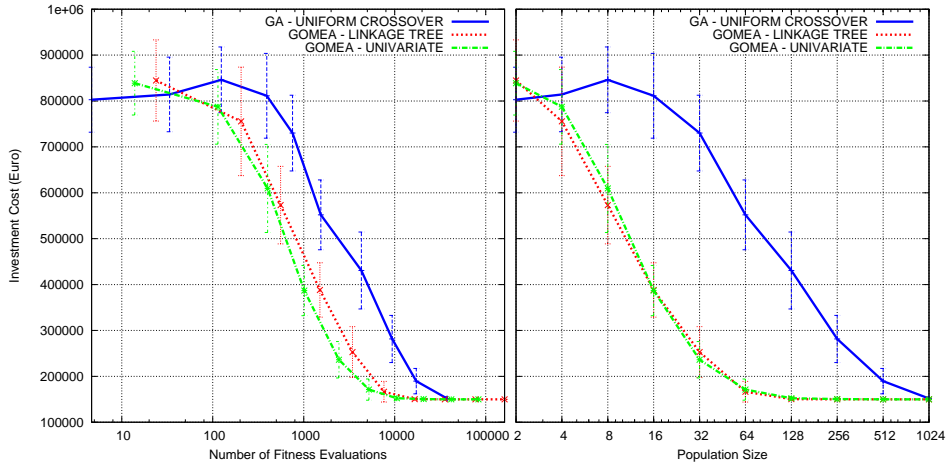

Fig. 3: Performance of GOMEA-LT, GOMEA-UNI and GA on minimizing the investment cost for enhancement of Network 1. Error bars show standard deviation.

computing time to come close to GOMEA but even for population size 1024, the traditional GA still cannot converge reliably to the same best solution obtained by GOMEA. If we use a too small population size, it is difficult to find feasible solutions, which explains why the line representing GA goes up first (feasible solutions can be more expensive than infeasible solutions) before it starts to go down when feasible solutions are found. Network 1 is a small distribution network containing only 25 branches (i.e. the number of decision variables), and while the variables are independent when evaluating the investment cost func-

tion, they are also linked when evaluating the constraints. However, depending on the problem instance, these linkages may be weak and of little influence, especially if the problem size is small. This explains why GOMEA-UNI, which assumes no dependencies among variables, requires less computing times than GOMEA-LT, which has an overhead of learning linkage trees and evaluating unnecessary mixings of (weak) linkage groups. This calls for the need of filtering spurious linkage groups in the linkage learning process as pointed out in [2]. It should be noted that when considering reliable convergence (30/30 runs) to the best solution ever found, GOMEA-LT requires less evaluations. The convenience of independent decision variables that GOMEA-UNI can exploit is not available in more complicated networks, which can be seen in case of Network 2.
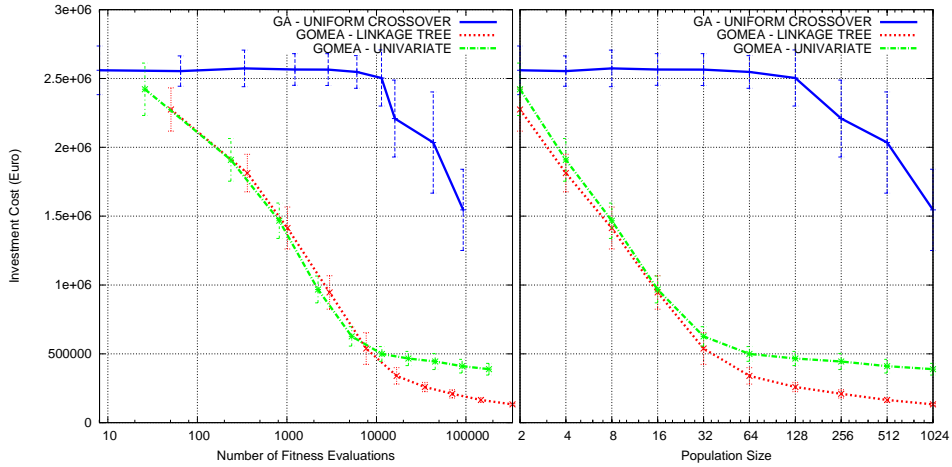

Fig. 4: Performance of GOMEA-LT, GOMEA-UNI and GA on minimizing the investment cost for enhancement of Network 2. Error bars show standard deviation.

Fig. 4 shows the experimental results of 3 optimizers on solving DNEP for Network 2. This test case has a much larger and more complicated search space compared to Network 1. It can be seen that if we continue to run the optimization process with larger population sizes (and hence more power flow calculations), better solutions may still be obtained. Here, GOMEA-LT demonstrates that it has the best performance in comparison with the other 2 optimizers. The traditional GA has difficulty finding feasible solutions, let alone the optimum. GOMEA-UNI has a good performance here due to the intensive optimal mixing variation operator. However, without linkage learning, GOMEA-UNI does not obtain solutions of high quality as those found by GOMEA-LT. GOMEA-UNI can locate good solutions only if the decision variables are independent or weakly linked as in case of Network 1. Otherwise, GOMEA-UNI cannot efficiently find solutions that require the juxtaposition of multivariate linkage groups, e.g. as in the classic trap function benchmarks. GOMEA-LT wins over its univariate sibling in these cases.

5   Conclusions

The recently-developed gene-pool optimal mixing evolutionary algorithm (GOMEA) has so far been benchmarked on various theoretical optimization problems in the literature. Meanwhile, the long-existing traditional genetic algorithm (GA) has been widely used for numerous real-world optimization tasks. In this paper, we tackled the real-world problem of medium-voltage distribution network expansion planning (DNEP) with two instances of GOMEA: one with the univariate structure and one with the linkage tree. GOMEA was found to have much better performance than the traditional GA in terms of computing time and quality of the obtained solutions. Moreover, experimental results showed that linkage learning is truly beneficial for finding (near-)optimal solutions, not only in theoretical benchmarks but also in this engineering problem, further underlining the robustness of GOMEA and encouraging further applications of GOMEA on other real-world optimization problems.

References

1. Thierens, D., Bosman, P.A.N.: Optimal mixing evolutionary algorithms. In: 13th Annual Genetic and Evolutionary Computation Conference, GECCO 2011, Proceedings, Dublin, Ireland, July 12-16, 2011, ACM (2011) 617–624
2. Bosman, P.A.N., Thierens, D.: More concise and robust linkage learning by filtering and combining linkage hierarchies. In: Genetic and Evolutionary Computation Conference, GECCO '13, Amsterdam, The Netherlands, July 6-10, 2013, ACM (2013) 359–366
3. Puret, C.: Mv public distribution networks throughout the world. Technical Report 155, Merlin Gerin Group (March 1992)
4. Falaghi, H., Singh, C., Haghifam, M.R., Ramezani, M.: Dg integrated multistage distribution system expansion planning. International Journal of Electrical Power & Energy Systems **33**(8) (2011) 1489 − 1497
5. Carrano, E.G., Soares, L.A.E., Takahashi, R.H., Saldanha, R.R., Neto, O.M.: Electric distribution network multiobjective design using a problem-specific genetic algorithm. IEEE Transactions on Power Delivery **21**(2) (2006) 995 − 1005
6. Slootweg, J.G., Van Oirsouw, P.M.: Incorporating reliability calculations in routine network planning: Theory and practice. In: Proceedings of the 18th International Conference and Exhibition on Electricity Distribution - CIRED 2005. (2005) 1–5
7. Thierens, D., Goldberg, D.E.: Mixing in genetic algorithms. In: Proceedings of the 5th International Conference on Genetic Algorithms, Urbana-Champaign, IL, USA, June 1993, Morgan Kaufmann (1993) 38–47
8. Jaszkiewicz, A., Kominek, P.: Genetic local search with distance preserving recombination operator for a vehicle routing problem. European Journal of Operational Research **151**(2) (2003) 352 − 364
9. Bosman, P.A.N., Thierens, D.: Linkage neighbors, optimal mixing and forced improvements in genetic algorithms. In: Genetic and Evolutionary Computation Conference, GECCO '12, Philadelphia, PA, USA, July 7-11, 2012, ACM (2012) 585–592
10. Grainger, J.J., Stevenson, W.D.: Power System Analysis. McGraw-Hill Education (2003)