



HAL
open science

A Multi-Objective Memetic Optimization Method for Power Network Cascading Failures Protection

Yan-Fu Li, Enrico Zio, Giovanni Sansavini, L. R. Golea

► **To cite this version:**

Yan-Fu Li, Enrico Zio, Giovanni Sansavini, L. R. Golea. A Multi-Objective Memetic Optimization Method for Power Network Cascading Failures Protection. PSAM11 & ESREL 2012, Jun 2012, Finland. pp.1-10. hal-00737086

HAL Id: hal-00737086

<https://hal-centralesupelec.archives-ouvertes.fr/hal-00737086>

Submitted on 1 Oct 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Multi-Objective Memetic Optimization Method for Power Network Cascading Failures Protection

Y.F. Li^{a*}, E. Zio^{a,b}, G. Sansavini^b, L.R. Golea^b

^aChair on Systems Science and the Energetic Challenge, European Foundation for New Energy-Electricite' de France, at Ecole Centrale Paris-Supelec, Paris, France

^bEnergy Department, Politecnico di Milano, Milano, Italy

Abstract: Reliable and safe power grid operation requires the anticipation of cascading failures and the establishment of appropriate protection plans for their management. In this paper, we address this latter problem by line switching and propose a multi-objective memetic algorithm (MOMA), which combines the binary differential evolution algorithm with the non-dominated sorting mechanism and the Lamarckian local search. The 380 kV Italian power transmission network is used as a realistic test case.

Keywords: cascading events, line switching, multi-objective optimization, memetic algorithm, differential evolution, non-dominated sorting, Lamarckian local search

1. INTRODUCTION

In today's deregulated markets, critical infrastructures (CIs) (e.g. electric power grids, telecommunication networks, transportation networks, etc) are often run in stressful conditions which render their components more sensitive and vulnerable to natural and/or man-made disturbances [1]. In such systems, failure of one component may lead to a cascade of failures of other components and this can result in serious economical and social damages, as shown, for example, by recent large-scale blackouts of power grids [2-3].

We focus on power grids, for which a number of analytical/simulation models have been built to anticipate the impact of cascading failures [4]. These models encompass network protection (NP) techniques that aim at hampering the propagation of cascading failures. Such NP techniques are broadly divided in: 1) network interdiction [5-6], which enhances the network protection by designing components and allocating redundancies to avoid failures, and 2) line switching [7-9], which hinders the failure propagation by cutting off the possible 'directions' along which the failures can spread within the network. Line switching has been shown to be more capable of dealing with cascading failures since it requires proactive and economical actions that can be implemented immediately after the cascading initiation [8, 10]. The success of this technique relies on the search for the optimal set of lines to be switched-off, with the objective of minimizing the extent of cascading failures [10-12]. The related optimization problem is, in general, large-scale, nonlinear and combinatorial, for which heuristic algorithms (e.g. evolutionary algorithms (EAs)) can be effectively applied [13].

As recent extensions of EAs, memetic algorithms (MAs) are population-based meta-heuristic search methods combining global search strategies (e.g. EAs) with local search techniques (e.g. Lamarckian search) [14]. The rationale behind MAs is that the deficiency of EAs in local exploitation can be compensated by the inclusion of local search techniques which, on their account, are often inadequate in global exploration. MAs have been reported not only to converge to high quality solutions but also to be more efficient than conventional EAs [15].

In this paper, we formulate the optimal transmission line switching problem as a bi-objective optimization problem aiming at minimizing connectivity losses in the system at both global and local levels. We then propose a memetic solution approach composed of a multi-objective binary differential evolution algorithm (MOBDE) combining binary differential evolution (DE) [16] with fast non-dominated sorting [17] (as global search algorithm), and a Lamarckian local search (LLS) algorithm [15] (local search technique). The rest of this paper is organized as follows: Section 2 presents the cascading failure simulation model, the measure of cascading failure consequences, and the protection strategy, i.e. transmission line switching; Section 3 presents the general formulation of multi-objective optimization problems; Section 4 describes the detailed procedures of the proposed multi-objective memetic algorithm (MOMA); Section 5 presents the experiment results of MOMA and the comparison to MOBDE; Section 6 concludes this study and points out future research directions.

2. CASCADING FAILURE MODEL AND PROTECTION STRATEGIES

2.1 Cascading failure model

In general, the simulation of cascading failures in large-scale networks is computationally expensive. In order to focus on analyzing the failure propagation process and devising the prevention and/or mitigation measures, an abstract model has been developed by some of the authors [10]. A topological representation is used to abstract the electrical properties of the network, while retaining its structural properties. The representation is given in the form of a graph G consisting of N nodes (substations or buses) and K links (transmission lines). The N nodes are divided into two types: N_G generation nodes, i.e. sources of power, and N_D distribution nodes, i.e. loads or substations. The network structure is described by an $N \times N$ adjacency matrix $\{e_{ij}\}$: if there is a link between node i and node j , the entry $e_{ij} = 1$, otherwise $e_{ij} = 0$. The power is assumed to flow along the generator-distributor shortest paths [11-12, 18-20].

The load (or stress) on a network component, e.g. a node or transmission line, is modeled as dependent on the number of shortest paths transiting through it, when the power flow is sent from the available generation nodes to the distribution nodes. More precisely, the load, L_j of node j is measured by the node betweenness [21-22], calculated as the fraction of the generator-distributor shortest paths passing through that node. Likewise, the load, L_{ij} of a link ij , is measured by the edge betweenness, calculated as the fraction of the generator-distributor shortest paths through that link [23].

Each component in the network has a definite capacity, i.e. a maximum load it can sustain. The capacity of node j (or link ij) is proportional to its nominal load L_j (or L_{ij}) at which it is designed to operate, $C_j = (1 + \alpha)L_j$ (or $C_{ij} = (1 + \alpha)L_{ij}$), where $\alpha \geq 0$ is the tolerance parameter assumed the same for all elements of the entire network. Despite the simplicity of the concept of α , it can be regarded as an operating margin allowing safe operations of the components under possible load increments. When $\alpha = 0$, the system is working at its limit capacity: any further load added to a component would result in its failure and, possibly, in cascading failures affecting a large portion of the network.

In the case that a component's load L_j (or L_{ij}) exceeds its capacity C_j (or C_{ij}), it is considered as failed, and thus, removed from the network. This leads to a redistribution of the shortest paths in the network and, consequently, to a change in the loads of some working components. If the loads on some components exceed their capacities upon the redistribution of the shortest paths, the components fail and, consequently, a new redistribution follows. The process continues until there are no further failures or all the components are failed.

2.2 Measuring cascading failure consequences

To measure the effects of cascading failures, the connectivity loss C_L is used [10-12]. It quantifies the decrease of the ability of distribution substations to receive power from generators:

$$C_L = 1 - \frac{1}{N_D} \sum_{i=1}^{N_D} \frac{N_G^i}{N_G} \quad (1)$$

where N_G and N_D represent the numbers of generators and substations in the initial state of the network respectively, and N_G^i represents the number of generators able to supply power to distribution node i after the cascade of failures takes place. C_L is an indicator that measures the effects of cascading failures at the global system level. On the other hand, it fails to identify the critical locations to which power supply has to be guaranteed under any contingency. Therefore, a second, local indicator C_{L_A} is proposed to measure the effects of cascading failures on an identified critical region A. It quantifies the decrease of the ability of the distribution substations within the area A to receive power supply from any generator in the whole network:

$$C_{L_A} = 1 - \frac{1}{N_{D_A}} \sum_{i=1}^{N_{D_A}} \frac{N_G^i}{N_G} \quad (2)$$

where N_G represents the number of the generation nodes at the initial state of the network, N_{D_A} represents the number of distribution nodes within the area A at the initial state of the network, and N_G^i represents the number of generation units able to supply power flow to distribution node i in area A, after the cascade of failures takes place.

2.3 Cascading failure protection strategy

In this study, the NP is modeled as an operator intervention targeting at minimizing the effects of the cascade failure propagation at both system and local levels by switching off a set of transmission lines immediately after a cascading failure is triggered. Due to the rapid unfolding of a cascading failure, it is assumed that the protection intervention takes place only once after the cascade is triggered: no further actions to correct the effects of this only protective action are taken [10]. Then, the problem arises of what is the best set of lines whose disconnection hinders the cascade propagation at the maximum. This issue is crucial because the intentional disconnection of lines may worsen the effects of the cascade in the same way as failure propagation does [10].

As previously explained, the two indicators, C_L and C_{L_A} , can be used to quantify the effects of cascading failures. The global protection optimization seeks interventions that minimize the objective function $f(\mathbf{x}^*) = \min_{\mathbf{x}} C_L(\mathbf{x})$, i.e. the connectivity loss of the network configuration $\mathbf{x} = \{x_j | x_j \in \{0,1\}, j = 1, \dots, K\}$, where $x_j = 1$ if link j is operating, or 0 otherwise. On the other hand, the local optimization searches for the optimal intervention in terms of the line switching set \mathbf{x} that minimizes the connectivity loss C_{L_A} of a specific area A, $f(\mathbf{x}^*) = \min_{\mathbf{x}} C_{L_A}(\mathbf{x})$.

In order to fully take advantage of both perspectives on network protection, we look into a ‘‘hybrid’’ protection strategy that finds the set of lines to be switched-off, \mathbf{x} , that minimize both the connectivity loss of a pre-identified area A, $C_{L_A}(\mathbf{x})$, and the connectivity loss of the whole network, $C_L(\mathbf{x})$. This originates a multi-objective (MO) optimization problem, which is here tackled by a MOMA as is detailed in Section 4. The algorithm for simulating the combined effects of cascading failures and protection strategies proceeds in successive stages as follows [10].

- step 1. Initially, all components, N nodes and K links, are characterized by initial loads L_i and L_{ij} , respectively, and maximum capacities, C_i and C_{ij} , respectively.
- step 2. The initiating event occurs, i.e. the failure of the most critical component in terms of load, which is removed from the network. This triggers the cascading failure. Each component that is operating is tested for failure: for $i = 1, \dots, N$, if node i is working and $L_i > C_i$ then node i fails; if link ij is working and $L_{ij} > C_{ij}$ then link ij fails.
- step 3. The operator intervenes by switching off a set of lines, in order to limit the cascade failure effects, expressed by the objective functions relative to the hybrid optimization strategy. The MOMA optimization algorithm returns the most appropriate network configuration that minimizes the objective functions at the end of the current step $t = 1$.
- step 4. The components loads are re-distributed taking into account the new, protected network topology. Each component that is operating is tested for failure: for $i = 1, \dots, N$, if node i is working and $L_i > C_i$ then node i fails; if link ij is working and $L_{ij} > C_{ij}$ then link ij fails.
- step 5. The stage counter n is incremented by 1 and the algorithm is returned to step 4. The algorithm stops when no more working nodes fail.

3. GENERAL FORMULATION OF MULTI-OBJECTIVE OPTIMIZATION

Real world applications involve the simultaneous optimization of several objective functions, which are often competing or/and conflicting with one another, and subject to a number of equality and inequality constraints. In general, these multi-objective problems can be formulated as follows (in terms of minimization):

$$\text{Minimize} \quad f_o(\mathbf{x}), \quad o = 1, \dots, O \quad (3)$$

$$\text{Subject to} \quad \begin{cases} g_l(\mathbf{x}) = 0, & l = 1, \dots, L \\ h_m(\mathbf{x}) \leq 0, & m = 1, \dots, M \end{cases} \quad (4)$$

where f_o is the o -th objective function, \mathbf{x} is a decision vector that represents a solution, O is the number of objectives, g_l is the l -th of the L equality constraints and h_m is the m -th of the M inequality constraints. The objective functions $f_o(\mathbf{x})$ must be evaluated in correspondence of each decision variable vector \mathbf{x} in the search space. The final goal is to identify a set of optimal decision variable vectors \mathbf{x}_p^* , $p = 1, 2, \dots, P$, instead of a single optimal solution. In this set of optimal solutions, no one can be regarded to be better than any other with respect to all the objective functions. The comparison of solutions may be achieved in terms

of the concepts of Pareto optimality and dominance [24]: in case of a minimization problem, solution \mathbf{x}_a is regarded to dominate solution \mathbf{x}_b ($\mathbf{x}_a > \mathbf{x}_b$) if both following conditions are satisfied:

$$\forall i \in \{1, 2, \dots, O\}, f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b) \quad (5)$$

$$\exists j \in \{1, 2, \dots, O\}, f_j(\mathbf{x}_a) < f_j(\mathbf{x}_b) \quad (6)$$

If any of the above two conditions is violated, the solution \mathbf{x}_a does not dominate the solution \mathbf{x}_b , and \mathbf{x}_b is said to be nondominated by \mathbf{x}_a . The solutions that are nondominated within the entire search space are denoted as *Pareto-optimal* and constitute the *Pareto-optimal set*, and the corresponding values of the objective functions form the so called *Pareto-optimal front* in the objective functions space. The goal of a multi-objective optimization algorithm is to guide the search for solutions in the Pareto-optimal set, while maintaining diversity so as to cover well the Pareto-optimal front and thus allow flexibility in the final decision on the solutions to be actually implemented.

4. MULTI-OBJECTIVE MEMETIC ALGORITHM

In general, MA consists of two types of operations: global exploration and local exploitation. To design an effective and efficient MA for global optimization, the exploration abilities of a global search algorithm and the exploitation abilities of a local search algorithm need to be well-balanced [25]. In this Section, we present the operation procedures of the MOMA have developed by combining MOBDE and LLS for the bi-objective NP problem of interest.

4.1 Global search algorithm

DE has been originally proposed as a population-based global optimization algorithm for real-valued optimization problems [26]. The standard DE algorithm is simple and efficient and has been successfully applied in various scientific and engineering fields [27-29] often with superior performance than alternative optimization algorithms, e.g. particle swarm optimization and GAs [30]. Modified binary differential evolution (MBDE) is a binary version of DE developed to tackle single-objective binary-coded optimization problems [16]. In order to solve the combinatorial multi-objective problem of interest, we introduce into MBDE the non-dominated sorting, ranking, and elitism techniques utilized in non-dominated sorting genetic algorithm-II (NSGA-II) [13]. This new version of MBDE, named MOBDE, proceeds with the following steps:

Step 1. Initialization of parameters

Define the values of: the population size NP , the crossover rate CR , the scaling factor F , and the maximum number of generations N_{max} .

Step 2. Generation of initial population

Set the generation number t equal to 1. Initialize the population $\mathbf{X}_t = \{\mathbf{x}_1^t, \dots, \mathbf{x}_M^t\}$ which contains M real-valued parameter vectors of length K . Each vector is also called a *chromosome* and forms a candidate solution to the optimization problem. Each element of each vector $x_{ij}^t = \{x_{ij}^t | x_{ij}^t \in \{0, 1\}\}$, $i = 1, 2, \dots, N$, $j = 1, \dots, K$ takes a value (b_{ij}) from the set $\{0, 1\}$ with probability equal to 0.5: the element takes value '1' if the corresponding line is to be switched-off, 0 otherwise.

Step 3. Generation of intermediate population

Apply the binary tournament selection operator [13] to the population \mathbf{X}_t to generate a trial population $\mathbf{X}'_t = \{\mathbf{x}'_1, \dots, \mathbf{x}'_M\}$, which undergoes the evolution operations of mutation and crossover to become an intermediate population $\mathbf{V}_t = \{\mathbf{v}_1, \dots, \mathbf{v}_M\}$.

Step 3.1 Mutation

Apply the mutation operator (7) onto each binary chromosome of \mathbf{X}'_t [16]:

$$P(x'_{ij}) = 1 / (1 + \exp(\frac{2b[x'_{r1,j} + F(x'_{r2,j} - x'_{r3,j}) - 0.5]}{1 + 2F})) \text{ where } j = \{1, \dots, K\} \quad (7)$$

where b is a positive real constant, often set to values around 6, F is the scaling factor, $x_{r_1,j}^{t'}$, $x_{r_2,j}^{t'}$, and $x_{r_3,j}^{t'}$ are the elements at the j -th position of the three randomly chosen chromosomes $\mathbf{x}_{r_1}^{t'}$, $\mathbf{x}_{r_2}^{t'}$, and $\mathbf{x}_{r_3}^{t'}$, with indexes $r_1 \neq r_2 \neq r_3 \neq i$. After applying (7) onto the current chromosome, the noisy vector is generated as,

$$u_{ij}^t = \begin{cases} 1, & \text{if } rand \leq P(x_{ij}^{t'}) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

where $rand$ is a uniformly distributed random number within the interval $[0,1]$.

Step 3.2 Crossover

Apply the crossover operator (9) to mix the noisy and target vectors to create an intermediate vector \mathbf{v}_i . The vector inherits different pieces from the noisy and target vectors, as regulated by the crossover rate CR . The commonly used binomial crossover is defined as:

$$v_{ij}^t = \begin{cases} u_{ij}^t, & \text{if } U(0,1) \leq CR \text{ or } j = irand(M) \\ x_{ij}^{t'}, & \text{if } U(0,1) > CR \text{ or } j \neq irand(M) \end{cases} \quad \forall j \in \{1,2, \dots, K\} \quad (9)$$

where $U(0,1]$ is a uniform random value $\in (0,1]$, $irand(M)$ is a uniform discrete random number in the set $\{1,2, \dots, M\}$, j is the index of the dimensionality and K is the length of the chromosome.

Step 4. Evaluation

Evaluate the two objectives C_L and C_{L_A} for the M chromosomes in the population \mathbf{V}_t .

Step 5. Union and sorting

Combine the parent and the intermediate populations to obtain a union population $\mathbf{R}_t = \mathbf{X}_t \cup \mathbf{V}_t$. Rank the chromosomes in the population \mathbf{R}_t by running the fast non-dominated sorting algorithm [13] with respect to the objective values, and identify the ranked non-dominated fronts F_1, F_2, \dots, F_k where F_1 is the best front, F_2 is the second best front and F_k is the least good front.

Step 6. Selection

Select the first NP chromosomes from \mathbf{R}_t to create a new parent population \mathbf{X}_{t+1} . The crowding distance measure is used in this step to compare the chromosomes with the same rank (a more 'crowded' chromosome has lower priority than a less 'crowded' one), where crowding refers to the density of solutions present in a neighborhood of a chromosome of specified radius [13]. Increase the generation number t by 1 and go to Step 3.

The algorithm stops once the pre-defined maximum number of generations is reached.

4.2 Local search algorithm

Local search can enhance the capability of global search. In single-objective MAs, the local search operator is driven by the same objective function as the global search operator. However, the design of a MOMA is different, because the local search is generally guided by a single-objective for finding a single optimal solution while the global search is supposed to search for a number of non-dominated solutions with respect to multiple objectives [15]. Therefore, the local search needs to be modified to handle multiple objectives. In our study, the Pareto dominance is considered to guide also the local search (i.e. to determine if the newly obtained solution is better than the current one).

Local search design issues include the length of the local search and the selection of the individuals to undergo local search. The length of local search is critical to the successful design of memetic algorithms because a too short local search may not efficiently explore the neighborhood of the current solution and therefore bring little improvement to the search quality. Conversely, an overly lengthy local search may consume unnecessary fitness function evaluations and therefore hinder the efficiency of the memetic algorithm altogether. For the selection of individuals whose neighborhood should be explored, in principle

the local search operator should be applied to those individuals which are the most likely ones to direct the search towards the global optimum. In evolutionary computation, the individuals with best fitness values are generally regarded as the most preferable individuals for reproduction.

To take into account the above design issues, we have modified the 1-Opt LLS algorithm [31] to handle multiple objectives. 1-Opt searches for improvements by randomly flipping one bit of the chromosome at each iteration. As soon as an improved (in terms of Pareto dominance) solution is found, the LLS algorithm is terminated and the modified solution replaces the original solution. If no improvement is found, 1-Opt LLS is stopped upon reaching a maximum number of iterations, N_{Lmax} . The individual undergoing the local search is randomly selected from the best front F_1 in the current generation.

5. CASE STUDY AND RESULTS

The 380 kV Italian power transmission network (Figure 1) has been taken as case study. The network can be modeled as a graph of $N = 127$ nodes ($N_G = 30$ generators and $N_D = 97$ distributors) and $K = 171$ edges.



Figure 1. The 380 kV Italian power transmission network [32]

In a previous study by some of the authors [10], it was found that the removal of link 107 (connecting nodes 78 and 81) results in the largest damage in terms of $C_L = 0.96$. Since the region of Lombardy includes the largest number of distribution nodes in the power grid (densely-dotted area in the upper side of Figure 1), i.e. 21 distributors, the local protection strategy will focus on the minimization of the loss of connectivity of this specific area, C_{L_A} . Figure 2 presents the evolution of the cascade after the initial failure, without intervention. It is shown that the cascading failure stops after the step $t=2$.

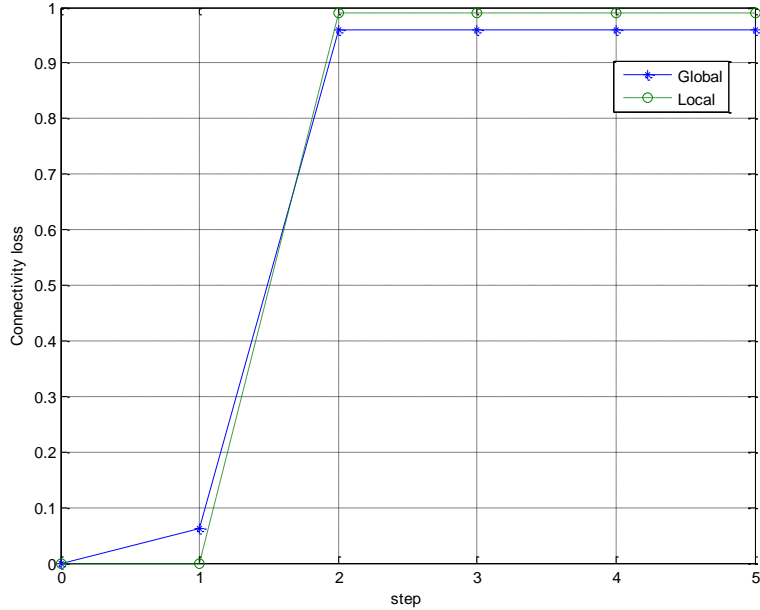


Figure 2. The cascade development in terms of global connectivity loss, C_L , and local connectivity loss, C_{L_A} , after the failure of line 107 without intervention

According to the protection strategy presented in Section 2.3, it is assumed that the protection intervention (line switching) takes place only once when $t = 1$ after the failure of the transmission line 107. The configurations of the MOMA parameters are summarized in Table 1. For the purpose of comparison, the MOBDE without local search is also run. To ensure that both algorithms do the same number of fitness evaluations (one-step cascade failure calculations, in this case), the population size of MOBDE is set to 40. The remaining parameter settings of MOMA and MOBDE are identical.

Table 1. The parameters of the MOMA algorithm

MOMA parameters	
Population size NP	30
Dimensionality of solutions K	171
Crossover rate CR	0.8
Scaling factor F	0.2
Minimum fitness error	10^{-4}
Maximum number of generation	300
Maximum number of local search	10

Figure 3 illustrates the Pareto fronts for the bi-objective (global connectivity loss, C_L , and local connectivity loss, C_{L_A}) problem obtained by MOBDE and MOMA at $t = 1$. It is shown that MOMA achieves a better Pareto front than MOBDE does, with the same number of fitness evaluations.

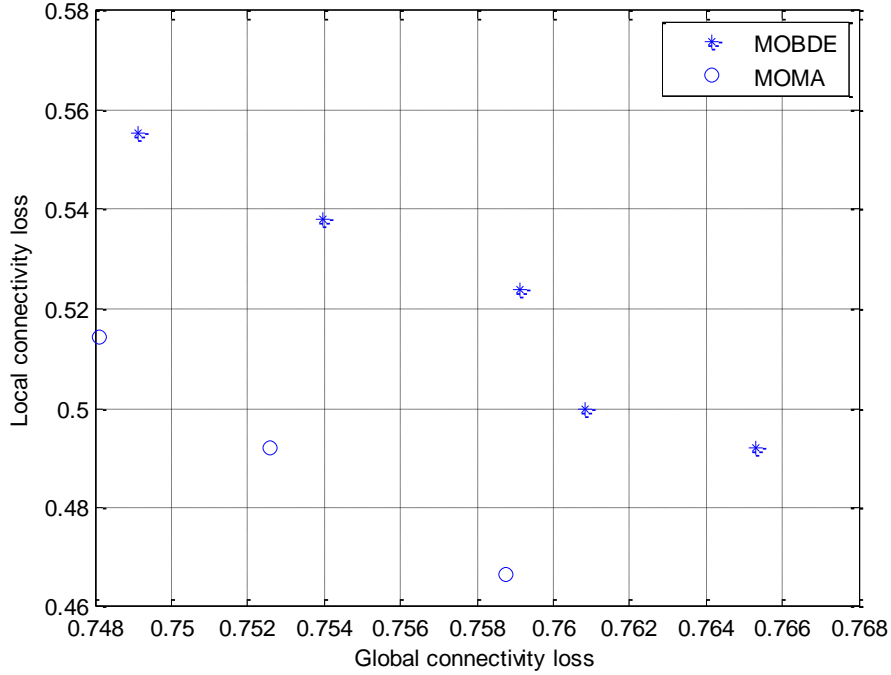


Figure 3. The Pareto fronts obtained by MOBDE and MOMA at $t = 1$

Table 2 shows the complete evolutions of the cascades with optimal interventions (obtained by MOBDE and MOMA, respectively) at $t = 1$, and without intervention. It is interesting to observe that the optimal protection strategies at $t = 1$ do not necessarily guarantee the minimization of C_L at the end of the cascades. For example, solution #4 obtained by MOBDE has final $C_L = 0.875$ and $C_{L_A} = 0.860$, while it has $C_L = 0.765$ and $C_{L_A} = 0.492$ at $t = 1$; solution #3 obtained by MOMA has final $C_L = 0.870$ and $C_{L_A} = 0.900$, while it has $C_L = 0.753$ and $C_{L_A} = 0.492$ at $t = 1$. Solution #3 by MOMA dominates solution #4 by MOBDE at $t = 1$, but they are mutually non-dominated at the end of the cascade. Another observation is that solutions #1 and #2 obtained by MOBDE show that the cascade propagation ends at $t = 1$ and they are the best results obtained at the end of the cascade propagation.

These observations lead to a direction of future work aiming at improving the optimal protection strategy by taking into consideration the final results of cascades rather than only a short horizon of one step cascading.

Table 2. Results of cascading failures w/o optimal protections

Method	Steps	Solution #1		Solution #2		Solution #3		Solution #4		Solution #5	
		C_L	C_{L_A}	C_L	C_{L_A}	C_L	C_{L_A}	C_L	C_{L_A}	C_L	C_{L_A}
MOBDE	0	0.062	0.000	0.062	0.000	0.062	0.000	0.062	0.000	0.062	0.000
	1	0.754	0.538	0.761	0.500	0.759	0.524	0.765	0.492	0.749	0.556
	2					0.870	0.862	0.875	0.860	0.816	0.683
MOMA	0	0.062	0.000	0.062	0.000	0.062	0.000				
	1	0.748	0.514	0.759	0.467	0.753	0.492				
	2	0.949	0.970	0.856	0.811	0.870	0.900				
No intervention	0	0.062	0.000								
	1	0.960	0.980								

6. CONCLUSIONS

For optimal NP against cascading failures, we have proposed a MOMA combining a binary differential evolution algorithm with a Lamarckian local search to identify the transmission lines to be switched-off for minimizing the global and local effects on the network.

Numerical application on a mid-size network has proved both feasible and satisfactory. The comparison with the MOBDE shows that MOMA can achieve superior solutions given the same number of evaluations.

References

- [1] Alguacil, N., Carrión, M., and Arroyo, J. M., "Transmission network expansion planning under deliberate outages," *Electrical Power and Energy Systems*, vol. 31, pp. 553–561, 2009.
- [2] Sachtjen, M. L., Carreras, B. A., and Lynch, V. E., "Disturbances in a power transmission system," *Physical Review E*, vol. 61, pp. 4877–4882, 2000.
- [3] Glanz, J. and Perez-Pena, R., "90 Seconds That Left Tens of Millions of People in the Dark," in *New York Times*, ed, 2003.
- [4] Dobson, I., Benjamin, A. C., Lynch, V. E., and Newman, D. E., "Complex systems analysis of series of blackouts: Cascading failure, critical points, and self-organization," *Chaos*, vol. 17, pp. 026103-1-026103-13, 2002.
- [5] Rocco, S. C. M. and Ramirez-Marquez, J. E., "Deterministic network interdiction optimization via an evolutionary approach," *Reliability Engineering and System Safety*, vol. 94, pp. 568–576, 2009.
- [6] Arroyo, J. M. and Fernández, F. J., "A Genetic Algorithm Approach for the Analysis of Electric Grid Interdiction with Line Switching," presented at the Intelligent System Applications to Power Systems, ISAP '09, 15th International Conference on, Curitiba, 2009.
- [7] Granelli, G., Montagna, M., Zanellini, F., Bresesti, P., Vailati, R., and Innorta, M., "Optimal network reconfiguration for congestion management by deterministic and genetic algorithms," *Electrical Power and Energy Systems*, vol. 76, pp. 549-556, 2006.
- [8] Shao, W. and Vittal, V., "Corrective switching algorithm for relieving overloads and voltage violations," *IEEE Transactions on Power Systems*, vol. 20, pp. 1877–1885, 2005.
- [9] Abou EL Ela, A. A. and Spea, S. R., "Optimal corrective actions for power systems using multi-objective genetic algorithms," *Electrical Power and Energy Systems*, vol. 79, pp. 722-733, 2009.
- [10] Zio, E., Golea, L. R., and Sansavini, G., "Optimizing protections against cascades in network systems: a modified binary differential evolution algorithm," accepted for publication in *Reliability Engineering and System Safety*, 2012.
- [11] Albert, R., Albert, I., and Nakarado, G. L., "Structural vulnerability of the North American power grid," *Physical Review E*, vol. 69, 2004.
- [12] Dueñas-Osorio, L. and Vemuru, S. M., "Cascading failures in complex infrastructures systems," *Structural safety*, vol. 31, pp. 157-167, 2009.
- [13] Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T., "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182-197, 2002.
- [14] Moscato, P., "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms," Caltech, Pasadena, CA1989.
- [15] Ishibuchi, H., Hitotsuyanagi, Y., Tsukamoto, N., and Nojima, Y., "Implementation of Multiobjective Memetic Algorithms for Combinatorial Optimization Problems: A Knapsack Problem Case Study," in *Studies in Computational Intelligence*. vol. 171, ed: Springer, 2009, pp. 27-49.
- [16] Wang, L., Fu, X., Menhas, M. I., and Fei, M., "A Modified Binary Differential Evolution Algorithm," in *Life System Modeling and Intelligent Computing, Lecture Notes in Computer Science*. vol. 6329, ed, 2010, pp. 49-57.
- [17] Iorio, A. W. and Li, X., "Solving rotated multiobjective optimization problems using differential evolution," in *AI 2004: ADVANCES IN ARTIFICIAL INTELLIGENCE; Lecture Notes in Computer Science*, 2004, pp. 861-872.
- [18] Motter, A. E. and Lai, Y. C., "Cascade-based attacks on complex networks," *Physical Review E*, vol. 66, 2002.
- [19] Newman, M. E. J. and Girvan, M., "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, 2004.

- [20] Zio, E. and Sansavini, G., "Modeling Interdependent Network Systems for Identifying Cascade-Safe Operating Margins," *IEEE Transactions on Reliability*, vol. 60, pp. 94 - 101, 2011.
- [21] Little, R. G., "Controlling Cascading Failure: Understanding the Vulnerabilities of Interconnected Infrastructures," *Journal of Urban Technology*, vol. 9, pp. 109-123, 2002.
- [22] Freeman, L. C., Borgatti, S. P., and White, D. R., "Centrality in valued graphs: A measure of betweenness based on network flow," *Social Networks*, vol. 13, pp. 141-154, 1991.
- [23] Girvan, M. and Newman, M. E. J., "Community structure in social and biological networks," presented at the Proceedings of National Academy of Sciences of USA, Piscataway, NJ, 2002.
- [24] Sawaragy, Y., Nakayama, H., and Tanino, T., *Theory of multiobjective optimization*. Orlando, FL: Academic Press, 1985.
- [25] Ishibuchi, H., Yoshida, T., and Murata, T., "Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling," *IEEE Transactions on Evolutionary Computation* vol. 7, pp. 204-223, 2003.
- [26] Storn, R. and Price, K., "Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, CA.1995.
- [27] Saka, M. P., "Optimum design of steel sway frames to BS5950 using harmony search algorithm," *Journal of Constructional Steel Research*, vol. 65, pp. 36-43, 2009.
- [28] Mahdavi, M., Fesanghary, M., and Damangir, E., "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, pp. 1567-1579, 2007.
- [29] Geem, Z. W., Kim, J. H., and Loganathan, G. V., "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, pp. 60-68, 2001.
- [30] Ponsich, A. and Coello, C. A., "Differential Evolution performances for the solution of mixed integer constrained Process Engineering problems," *Applied Soft Computing*, vol. 11, pp. 399-409, 2009.
- [31] Noman, N. and Iba, H., "Accelerating Differential Evolution Using an Adaptive Local Search," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 101-125, 2008.
- [32] Rosato, V., Issacharoff, L., and Bologna, S., "Influence of the topology on the power flux of the Italian high-voltage electrical network," *Europhysics Letters* 2010 (in press).