

Preventing Overloading Incidents on Smart Grids: A Multiobjective Combinatorial Optimization Approach

Nikolaos Antoniadis¹[0000-0001-9266-3247], Maxime Cordy¹[0000-0001-8312-1358], Angelo Sifaleras²[0000-0002-5696-7021], and Yves Le Traon¹[0000-0002-1045-4861]

¹ Interdisciplinary Centre for Security, Reliability and Trust (SnT)
University of Luxembourg, Luxembourg, Luxembourg
{nikolaos.antoniadis, maxime.cordy, yves.lettraon}@uni.lu

² Department of Applied Informatics
University of Macedonia, Thessaloniki, Greece
sifalera@uom.gr

Abstract. Cable overloading is one of the most critical disturbances that may occur in smart grids, as it can cause damage to the distribution power lines. Therefore, the circuits are protected by fuses so that, the overload could trip the fuse, opening the circuit, and stopping the flow and heating. However, sustained overloads, even if they are below the safety limits, could also damage the wires. To prevent overload, smart grid operators can switch the fuses on or off to protect the circuits, or remotely curtail the over-producing/over-consuming users. Nevertheless, making the most appropriate decision is a daunting decision-making task, notably due to contractual and technical obligations. In this paper, we define and formulate the overloading prevention problem as a Multiobjective Mixed Integer Quadratically Constrained Program. We also suggest a solution method using a combinatorial optimization approach with a state-of-the-art exact solver. We evaluate this approach for this real-world problem together with Creos Luxembourg S.A., the leading grid operator in Luxembourg, and show that our method can suggest optimal countermeasures to operators facing potential overloading incidents.

Keywords: Smart grids · Electrical safety · Combinatorial Optimization · Integer linear programming

1 Introduction

The so-called smart grid paradigm was motivated by the need to manage the increasing complexity of today's electricity grids. It aims to follow the rising demand for energy, e.g., by integrating renewable energies or by providing innovative services, mainly driven by sensors and two-way communications between smart meters and electricity providers.

Part of the smart grid's power system [7] in Luxembourg [5], the low-voltage distribution grid, carries electric energy from distribution transformers to smart

meters of end customers. This low-voltage network is, in general, more complex and meshed than the medium-voltage one, and it is harder to track its disturbances.

Each distribution substation comprises the part of a power system that delivers electric energy to industrial and residential users, through feeder pillars, (i.e., cabinets and cables). Distribution cabinets control the distributed power and provide overload protection to the network lines, through their fuses. Between the service cable and each user installation, a smart meter is installed to measure the electric consumption and to manage loads, through its relay triggering feature. The number of connected components of the multigraph mentioned above is equal to the number of distribution substations, meaning that each service cable can only be connected to precisely one substation.

Every cable starts from a fuse in a cabinet and ends in another fuse in another cabinet. If the ending cabinet of the cable does not have any cable that starts from it, it is called *dead end*. The state of each fuse can be either open or closed; this information, combined with the topology of the grid, can be used to determine the reachability of each cable on the network, from each one's substation. The consumption values for each user are given through its smart meter. Each cable's current load is the summary of the production and consumption values of all the users on this cable.

The current load of each cable can be approximated using methods such as [12]. Accordingly, the load percentage of a cable is obtained by dividing its current load by its maximum ampacity multiplied by one hundred. Then the cable is at risk of *overloading* if its current load is over a predefined threshold.

1.1 Preventing an overloading incident

Grid operators typically consider that there is a risk of overloading incident when the current load percentage on a cable exceeds a predefined threshold (set by the grid operator). Then, they can apply different countermeasures to reduce cable loads, thereby avoiding the overloading to occur. The preferred solution consists of limiting the over-production remotely (e.g., solar panels on a sunny day) or over-consumption of specific users (e.g., charging EVs); this countermeasure is commonly named *load curtailment* [20]. However, some users have such contracts that prevent the operator from regulating their power capacity. Therefore, curtailment is not, in such cases, an option. More generally, if curtailments cannot result in a stable state (i.e., without risk of overloading), the operators have to reconfigure the topology of the grid, by switching fuses, using the *intertrip* [2] method to shift reserves from one network to another, even if intertrip is complicated for the meshed low-voltage network [2].

Changing fuse states require technicians to visit the corresponding cabinets physically. Therefore, minimizing the number of visiting cabinets is an object of considerable solicitude to the grid operator to minimize the restoration time of a potential incident. Another concern is the minimization of the number of fuses that have to be switched on or off, as grid's configuration should remain nearly the same to its initial state.

Avoid disconnecting users, especially critical ones, such as patients is a matter of great concern to the grid operators. Still, this may happen as a last resort to prevent cascading overloads [2] and to avoid any damage to the power line, when there is an insufficient operating reserve. In this case, the number of disconnected users should remain minimal.

1.2 Contribution

Given the above requirements, finding the ideal solution(s) to prevent overloading incident is a daunting decision-making task that humans can hardly solve without support. Therefore, in this paper, we propose a multiobjective combinatorial optimization approach to define, model, and solve the overloading prevention problem for a low-voltage network. Our approach can also model a medium-voltage smart grid or a standalone microgrid with a minimum number of changes. Mathematical optimization methods have been successfully applied to solve a wide range of decision problems [18], including in the energy industry (see Section 2).

Given the physical network data (i.e., substations, cabinets, cables, connections between cabinets), that are assumed to remain constant, the initial state of the fuses and the power values from the users' smart meters, we approximate the current load percentage on each cable by solving a linear system described in [12]. To create the matrices defining this linear system, we compute the reachable cables from every substation based on the fuses' state and the physical network data. We also detect parallel cables (i.e., multiple edges in the grid's multigraph), since computations involving those are slightly different.

Once the risk of overload is detected (i.e., the approximated current load percentage exceeds the predefined threshold), we store the current states of the fuses and the smart meter values. Then we solve our optimization model to suggest the most appropriate countermeasures. Curtailment of compliant users is first attempted.

If this curtailment cannot establish a stable state, the second action we should take is to switch fuses on or off. On every possible change of fuses' state, a new linear system has to be defined and solved in order to approximate the current loads on the cables. Moreover, simultaneously connecting multiple substations should be avoided, as we cannot calculate the power flow cycles between substations; otherwise, the load calculation could return a wrong result [12]. In the end, our solution aims to maximize the number of connected users while minimizing the number of visited cabinets and the number of changes applied to fuses.

We evaluate the applicability of our approach through a benchmark set comprising ten grid topologies for five substations; similar to an area of a small village in Luxembourg, and another set containing a gradually increasing number of substations, by steps of five, from ten to fifty; similar to an area of a medium-size city in Luxembourg. The topologies are generated by a tool we developed based on real-world statistics provided by Creos Luxembourg S.A., the only grid operator in Luxembourg and our project partner. Our results show that our approach is capable of suggesting solutions for all topologies in due time

(up to about 15 min). Moreover, a detailed analysis of the curtailed and disconnected users reveals that, curtailment alone is not enough to prevent overloading incidents, which emphasizes the need for automated solutions to reconfigure the grid, and more sophisticated demand response programs. The remainder of this paper is structured as follows. Section 2 discusses related work. Afterward, Section 3 provides the mathematical model for this work. Then, in Section 4, we detail the implementation of our proposed solution method, which is evaluated in Section 5. Finally, we conclude in Section 6.

2 Related work

Despite the fact that, the prevention of overloading incidents definitely concerns the grid operators, as of today, this problem is not studied enough. Several research works investigate how to prevent overload incidents using demand response programs. To the best of our knowledge, there is no detailed work that examine the overloading prevention problem in respect to demand response, for both producers and consumers, and grid reconfiguration on the same time.

Ramaswamy and Deconinck [15] define the grid reconfiguration problem as a multiobjective non-convex one and argue that a genetic algorithm is probably a good optimization method to solve it.

Han and Piette [11] describe different incentive-based demand response programs; usually reducing demands with a financial benefit for the customers. They present different methods such as direct load control, interruptible/curtailable rates, emergency demand response programs, capacity market program, and demand bidding/buyback programs. To prevent overloads, Bollen [2] present different curtailment schemes averting the operating reserve from getting insufficient, that could lead to overloads. In his work, the general directions of curtailment are given without giving many details about modeling and solving of the curtailment problem. Furthermore, Simao et al. [20] formulate the problem of planning short-term load curtailment in a dense urban area, as a stochastic mixed-integer optimization problem. They implement three approximation policies, and test them with a baseline policy where all curtailable loads are curtailed to the maximum amount possible. Even if in their work short-term planning is implemented, overloads are allowed, and the curtailment applies only to consumers of the grid.

In addition to the previous studies, Pashajavid et al. [13] present an overload management strategy that controls the supporting floating batteries in an autonomous microgrid and decides any possible connection between it and its neighboring microgrids, by monitoring the microgrids' frequency. However, in their work, no demand response program is considered.

Furthermore, Shahnian et al. [16] developed a dynamic multi-criteria decision-making algorithm to manage microgrid overloads. They also deploy a cloud theory-based probabilistic analysis to contemplate the uncertainties in the considered distribution network. Nevertheless, they were not considering reactive power in their approach to define overloading. Recently, Babalola et al. [1] proposed a multi-agent algorithm that, it does not require load shedding to prevent

cascading failures, such as overloaded lines after a contingency occurs. Nonetheless, their work is focused on power generators only.

3 Mathematical model

The overloading prevention problem can be defined on a complete undirected multigraph $G = (V, E)$. The set $V = \{1, 2, \dots, o\}$ is the vertex set, i.e., the set of the cabinets of the grid, $E = \{(i, j) \in V^2, i \neq j\}$ is the multiple edge set, i.e., the multiset of the cables that connect the cabinets of the grid. The problem, we previously described, can be modeled as a Mixed Integer Quadratically Constrained Program (MIQCP) formulation as follows:

$$\max \sum_{i=1}^n r_i \sum_{k=1}^m uc_{ki} \quad (1)$$

$$\min \sum_{b=1}^o dfcab_b \quad (2)$$

$$\min \sum_{f=1}^{2n} |x_f - x_f^0| \quad (3)$$

subject to:

$$A \cdot wp = P \quad (4)$$

$$A \cdot wq = Q \quad (5)$$

$$l_i < \lambda, \forall i \in \{1, \dots, n\} \quad (6)$$

Given G , the first objective (1) defines the fuses' state to maximize the serviced users of the grid. At the same time, the second objective (2) sets the state of each fuse to minimize the number of visiting cabinets. According to Creos Luxembourg SA, the cost of reconfiguration is nearly analogous to the number of the cabinets the technicians have to visit. The third objective (3) minimizes the number of fuses' changes to keep the initial fuses' state as much as possible.

Curtailement policy to the users is applied when any producer or consumer has amperage over I_{LP} and I_{LC} , respectively. Equations (4) and (5) approximate the current loads, as in [12]. To avoid overload cables, (6) constraint the current load percentage on each cable under the predefined threshold. The notation used is presented in the Appendix.

4 Implementation

As the problem above is formulated as a MIQCP, a state-of-the-art mathematical programming solver, Gurobi [10], is chosen to address it. Smart grid's data are imported to our program, and a pre-computational phase is taking place. Vectors, substations, cabinets, as well as the edges, cables, are stored, and the

multigraph of the smart grid is created. Additionally, the initial fuses' states, the smart meters, their connecting cables, and their consumption and production values are being read and stored. Moreover, the fundamental set of cycles of the multigraph [14,17] are being found, eliminating any connections between substations and investigating any multi-edges, multiple cables between cabinets, on the graph. During this pre-processing phase, the dead-ends cabinets are also defined, to help us compute the load, using the depth-first-search algorithm [21], and stored. Having this information about the topology, we construct the potential linear equations assuming that, all the fuses are closed. This phase ends by calculating the loads [12] by using Singular Value Decomposition [9] for solving the over-determined linear system of equations and check if the initial state has any overloaded cables or not. If an overload is inspected, then, the variables are being initialized and, using the depth-first-search algorithm [21], the reachability vector r is constructed. After the reachability cable state is initialized, we can create the actual linear equations, the cable, cabinet, dead-end and circle ones [12].

4.1 Linear transformation

As Gurobi does not support quadratic equality constraints, we need to transform the constraints (4) and (5) into a linear form. Firstly, we rewrite the constraints (4) and (5) as:

$$P_j = \sum_{f=1}^{2n} A_{jf} w p_f, \forall j \in \{1, \dots, leq\} \quad (7)$$

$$Q_j = \sum_{f=1}^{2n} A_{jf} w q_f, \forall j \in \{1, \dots, leq\} \quad (8)$$

We introduce, for each quadratic term in the above summations, new variables $z p_{jf} = A_{jf} w p_f$ and $z q_{jf} = A_{jf} w q_f$. As $A_{jf} \in \{-1, 0, 1\}$:

$$z p_{jf} = \begin{cases} -w p_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ w p_f, & A_{jf} = 1 \end{cases} \quad (9) \quad z q_{jf} = \begin{cases} -w q_f, & A_{jf} = -1 \\ 0, & A_{jf} = 0 \\ w q_f, & A_{jf} = 1 \end{cases} \quad (10)$$

Using the (9) and (10) we can rewrite the (7) and (8) as:

$$P_j = \sum_{f=1}^{2n} z p_{jf}, \forall j \in \{1, \dots, leq\} \quad (11) \quad Q_j = \sum_{f=1}^{2n} z q_{jf}, \forall j \in \{1, \dots, leq\} \quad (12)$$

To be able to compute the $z p_{jf}$ and $z q_{jf}$, we need to binary transform the above piecewise functions using indicator constraints [3]. Thus, for every coefficient matrix element, we introduce three additional variables as:

$$-1y_{jf1} + 0y_{jf2} + 1y_{jf3} = A_{jf}, \quad (13) \quad y_{jf1} + y_{jf2} + y_{jf3} = 1 \quad (14)$$

$$\forall j \in \{1, \dots, leq\}, \forall f \in \{1, \dots, 2n\}, y_{jf1}, y_{jf2}, y_{jf3} \in \{0, 1\}$$

In (13) it is ensured that A_{jf} can only take values from its domain where (14) ensures that, only one variable could take value one. Using the equations (13) and (14), the equations (9) and (10) become:

$$zp_{jf} = \begin{cases} -wp_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wp_f, & y_{jf3} = 1 \end{cases} \quad (15) \quad zq_{jf} = \begin{cases} -wq_f, & y_{jf1} = 1 \\ 0, & y_{jf2} = 1 \\ wq_f, & y_{jf3} = 1 \end{cases} \quad (16)$$

4.2 Solving model

The final step is to calculate the difference between the initial and the current state of each fuse. Moreover, the binary cabinet visit indicator for each cabinet is computed. To solve the model, we are using the lexicographic approach [4] for the objectives, to reach any Pareto optimal solution. This approach assigns a priority to each objective, and optimizes for the objectives in decreasing priority order. At each step, the current objective is optimized, and a constraint is introduced to guarantee that the higher-priority objective functions preserve their optimal value [4,10]. We are specifying an absolute order of importance along with our partner, Creos Luxembourg S.A. After getting the preference information, our first objective (1) has the highest importance, the second one (2) has lower importance and, the third one (3) has the least importance.

5 Evaluation

To be applicable in practice, our method has to provide solutions to the overloading prevention problem sufficiently fast. According to our partner Creos Luxembourg S.A., the computation time should not exceed 15 minutes (which corresponds to the interval of time between two smart meter data reports). Hence, our first research question concerns the scalability of our approach concerning increasingly-large grids.

Our primary focus is to analyze the presented solution qualitatively, that is, how much our approach manages to satisfy the requirements of not disconnecting, if possible, the users. The absolute numbers, of course, depend on the particular cases considered. Therefore, our second research question concerns a relative analysis: how well different curtailment policies allow avoiding user disconnections in different overload scenarios.

5.1 Dataset and experimental setup

A topology generation software tool to evaluate our proposed method was first developed. Using this tool, we create ten realistic smart grid graphs based on real topology data. On each instance, we consider five substations, to answer the second research question, topologies that resemble the size of a small village in Luxembourg. For every grid graph, we consider 216 scenarios³ as a combination

³ Interested readers may find all the presented results for the 216 instances from <https://github.com/nikosantoniadis/PrevOvrlIncidentsResults>

of different percentage of overload producers, overload consumers, producers, and consumers that can be curtailed. Moreover, as we do not notice a significant difference, when changing the percentage of overloading and curtailment in timing, we study the scalability only with regard to the grid size by creating another nine realistic smart grid graphs. Each instance contains a gradually increasing number of substations, by steps of five, from ten to fifty.

On these graphs, three to four cabinets are connected on each substation, where the number of cabinets is uniformly random. Two of these cabinets are connected by two edges (cables) to the substation. Under the first level of cabinets, three to five cabinets are connected, where the number of cabinets is uniformly random. Under the second level of cabinets, zero to two cabinets are connected, where the number of cabinets is uniformly random. During the experiments' creation, it is assured that one cabinet, either on the second or the third level of the graph, is connected to another substation's cabinet, so that intertrip can be applied. For each cable, the material, the size and the maximum ampacity are generated uniformly randomly from real data. On each cable, up to 21 smart meters are connected. The number of smart meters was sampled from a uniform discrete distribution with range $[0, 21]$.

To create consumption and production energy data, we analyzed the historical data we acquired from Creos Luxembourg S.A. More specifically, we analyzed for the 215 consumers and the seven producers, the four electrical values from their smart meters, active energy consumption and production and reactive energy consumption and production. The data consisted of 9 months of measurements, with 96 measurements per day. Mean and standard deviation, as well as minimum and maximum value for each user, was computed to produce their consumption and production profiles. For each smart meter, a random profile is selected and, from the corresponding distribution, an electrical value is generated. Additionally, at most 10% of the users are selected to produce energy. To create a different percentage of overloaded and curtailed users, we shuffle the producers and consumers vectors using the Fisher-Yates algorithm [8,6]. Then, we pick the corresponding number of users from the shuffled vectors.

A *soft curtailment* [2] is applied if a producer overpasses the threshold of 60A, i.e., 80% of 75A, the typical roof-top solar panel installation amperage, or if a consumer overpasses the threshold of 32A, i.e., 80% of 40A, the typical amperage supplied by residential meters. If a producer or a consumer is picked for curtailment, its active energy is limited to 20A; a value picked together with Creos Luxembourg S.A. The experiments were conducted on a standard MacBook Pro with a 2.6 GHz Intel Core i7 processor, macOS Mojave 10.14.6 operating system, 16 GB 2133 MHz LPDDR3 memory using Java JDK 1.8.0-162 and Gurobi Optimizer 8.1.1 – Academic Version [10].

5.2 Results and discussion

In what follows, each experiment was run for ten times, for the ten different topologies, and the 216 different scenarios. The average time and the 99% confi-

dence interval for the 21600 experiments (5 substations), was found to be equal to $6.363 \text{ sec} \pm 1.527 \text{ sec}$.

Table 1. Grid topologies and computation times.

Subst.	t(sec.)	t(sec.)
	w/o curt.	w/curt.
10	28.5	31.2
15	70.5	75.5
20	105.4	112.4
25	172.1	167.5
30	274.1	298.1
35	436.0	408.8
40	520.2	529.3
45	782.0	806.0
50	887.0	911.5

Table 2. Sample results of our method (5 substations).

POv ⁴	COv ⁴	PCur ⁴	CCur ⁴	ConU ⁴	VisCab ⁴	FusesCh ⁴
0-25%	0-10%	100%	100%	100%	6.98%	3.47%
0%	25%	-	100%	99.96%	7.04%	3.5%
0%	50%	-	100%	95.07%	11.55%	5.5%
25%	25%	100%	100%	99.8%	7.18%	3.56%
10%	50%	50%	100%	94.7%	12%	5.78%
50%	10%	100%	100%	99.94%	7.02%	3.49%
10%	0%	0%	-	94.57%	10.75%	5.37%
10%	0%	50%	-	97.68%	8.47%	4.19%
50%	25%	100%	100%	99.35%	7.67%	3.76%
50%	25%	50%	50%	61.57%	31.6%	17.18%
50%	25%	0%	0%	41.77%	35.83%	23.06%

We observe that our method can propose solutions, quickly, to help grid operators to prevent overloading incidents. To check if our method could be applied to a larger scale smart grid, we create and test nine different topologies, and the results of these experiments are shown in Table 1. Indeed, even in the most complex case, that resembles the size of a medium-size city in Luxembourg, our approach finds a solution in about the allowed time (15 min). Moreover, we notice that when the size of the graph doubles, the average computation time is approximately five times higher. For the second research question, as shown in Table 2, if the percentage of overloaded consumers remains at most 10%, while the percentage of overloaded producers remains at most 25%, and curtailment is applied for all the users of the grid, no disconnection is needed. Nonetheless, for the same as the above scenarios, the percentages of cabinets to visit and changed fuses remain low; 6.98% and 3.47%, respectively.

On the opposite, with no curtailment, even when a tenth of the producers is overloading, $5.43\% \pm 0.93\%$ of the users should be disconnected to prevent overload.

From our findings, it is shown that curtailment policies lead to fewer disconnections to prevent overloads. Additionally, less cabinet visits, and less changed fuses are needed, avoiding additional costs for the electrical companies, while keeping the grid on a stable configured state, as possible. Nevertheless, in the long term, electrical companies should increase their operational reserves to decrease the possibility of disconnections [2]. Moreover, solar panel producers should install batteries to minimize their losses due to the curtailment policies [2].

⁴ *POv*: overload producers' percentage, *COv*: overload consumers' percentage, *PCur*: curtailed producers' percentage, *CCur*: curtailed consumers' percentage, *ConU*: connected users percentage, *VisCab*: cabinets to visit percentage, *FusesCh*: fuses changed percentage

6 Conclusions and future work

We defined and formulated the overloading prevention problem in smart grids as a Multiobjective MIQCP and suggested a solution method using a state-of-the-art exact solver. It is shown that this approach can be included in the grid operator’s decision-making process as it can successfully and rapidly help to prevent challenging overloading incidents in a smart grid of about the size of a medium city in Luxembourg, minimizing the disconnections of the grid’s users.

Our method has been integrated into a grid visualization tool that, allows operators to observe the grid cable states, detect (risk of) overloading incident, and call our algorithm to find appropriate countermeasures. In the longer term, the integrated software will be used directly by Creos Luxembourg operators. Moreover, our approach can be parallelized to analyze every substation subgraph independently from other ones, as in [12].

As future work, we plan to analyze the intermediate states to find the optimal order of fuses’ change. During the analysis of these intermediate states, a “trade-off” metric should be calculated, as the difference between the maximum and the minimum load on the grid. This metric should offer an optimal trade-off between the number of actions to perform and the maximal overload that any cable or substation reaches during the execution of the actions. Furthermore, we plan to apply a dynamic soft curtailment policy [2] to the grid’s users. Another interesting addition should be the appliance of a fairness policy to avoid curtailing the same users repetitively over time. Such considerations, raise the need for considering the future states of the grid and their inherent stochasticity, as the recovery response solution should guarantee stability over the next 24 hours. Inevitably, the aforementioned considerations complexify the problem, increasing the size of the problem and its solution space. As such, exact methods may not be suitable to address those new concerns. Thus, we also plan to exploit metaheuristic methods [19] to solve the overloading prevention problem.

Acknowledgement. The authors would like to thank Yves Reckinger and Robert Graglia from Creos Luxembourg S.A. for their support.

Appendix: Nomenclature

The next list describes several symbols that are used within the body of the document

Indices

b	cabinet index, $b \in \{1, \dots, o\}$
f	fuse index, $f \in \{1, \dots, 2n\}$
i	cable index, $i \in \{1, \dots, n\}$
j	linear equation index, $j \in \{1, \dots, leq\}$
k	user index, $k \in \{1, \dots, m\}$

Parameters

δ	measurement frequency coefficient; e.g. $\frac{60}{15} = 4$, for 15 min interval
λ	maximum allowed current load percentage for all cables, e.g. 80%
aE_k	active energy for user k , $aE_k = aEC_k - aEP_k$, $aE_k \in \mathbb{R}$
aEC_k	active energy consumption for user k , $aEC_k \in \mathbb{R}_+$
aEP_k	active energy production for user k , $aEP_k \in \mathbb{R}_+$
cc_{bf}	fuse cabinet indicator; 1 if fuse f belongs to the cabinet b , 0 otherwise
cl_i	maximum allowed current load in cable i , e.g. 100A
cur_k	amperage of user k , $cur_k = \frac{\sqrt{aE_k^2 + rE_k^2}}{\sqrt{3} \cdot 230}$
I_R	curtailed amperage for users, e.g. 20A
I_{LC}	maximum allowed amperage for consumers, e.g. 32A
I_{LP}	maximum allowed amperage for producers, e.g. 60A
leq	number of linear equations, $leq \in \mathbb{N}^*$
m	number of users, $m \in \mathbb{N}^*$
n	number of cables, $n \in \mathbb{N}^*$
o	number of cabinets (including substations), $o \in \mathbb{N}^*$
Pl_i	initial active energy for cable i , $Pl_i = \delta \sum_{k=1}^m uc_{ki} RaE_k$
Ql_i	initial reactive energy for cable i , $Ql_i = \delta \sum_{k=1}^m uc_{ki} rE_k$
RaE_k	real active energy consumption for user k , $RaE_k = aE_k$, if $cur_k < I_{LC}$, (consumer) or $cur_k < I_{LP}$ (producer), and $RaE_k = RGaE_k$ otherwise
rE_k	reactive energy for user k , $rE_k = rEC_k - rEP_k$, $rE_k \in \mathbb{R}$
rEC_k	reactive energy consumption for user k , $rEC_k \in \mathbb{R}_+$
rEP_k	reactive energy production for user k , $rEP_k \in \mathbb{R}_+$
$RGaE_k$	curtailed active energy for user k , $RGaE_k = \sqrt{[230^2 \cdot 3 \cdot I_R^2 - rE_k^2]}$, $RGaE_k \in \mathbb{R}_+$
uc_{ki}	user cable indicator; 1 if user k is connected with cable i , 0 otherwise
x_f^0	initial fuse state; 1 if fuse f is closed, and 0 otherwise; if $f = 2i$, x_f^0 denotes the initial state of the <i>start</i> fuse of cable i , else if $f = 2i + 1$, x_f^0 denotes the initial state of the <i>end</i> fuse of cable i

Variables

A_{jf}	coefficient matrix element; for equation j and fuse f , $A_{jf} \in \{-1, 0, 1\}$
$dfcab_b$	cabinet visit indicator; 1 if $\sum_{f=1}^{2n} cc_{bf} x_f - x_f^0 \geq 1$, 0 otherwise
l_i	actual current load percentage, at cable i ; $l_i = \max\left(\frac{100\sqrt{wp_{2i}^2 + wq_{2i}^2}}{230cl_i\sqrt{3}}, \frac{100\sqrt{wp_{2i+1}^2 + wq_{2i+1}^2}}{230cl_i\sqrt{3}}\right)$
P_j	active load vector element; $P_j = Pl_i \cdot r_i$, if equation j is describing the current flow of cable i , and 0 otherwise, $P_j \in \mathbb{R}$
Q_j	reactive load vector element; $Q_j = Ql_i \cdot r_i$, if equation j is describing the current flow of cable i , and 0 otherwise, $Q_j \in \mathbb{R}$
r_i	reachability cable state; 1 if cable i is powered and 0 otherwise
wp_f	actual active energy vector energy element for fuse f ; $wp_f \in \mathbb{R}$
wq_f	actual reactive energy vector energy element for fuse f ; $wq_f \in \mathbb{R}$
x_f	fuse state; 1 if fuse f is closed, and 0 otherwise; if $f = 2i$, x_f denotes the current state of the <i>start</i> fuse of cable i , else if $f = 2i + 1$, x_f denotes the current state of the <i>end</i> fuse of cable i

References

1. Babalola, A.A., Belkacemi, R., Zarrabian, S.: Real-time cascading failures prevention for multiple contingencies in smart grids through a multi-agent system. *IEEE Transactions on Smart Grid* **9**(1), 373–385 (Jan 2018)
2. Bollen, M.H.: The smart grid: Adapting the power system to new challenges. *Synthesis Lectures on Power Electronics* **2**(1), 1–180 (2011)
3. Bonami, P., Lodi, A., Tramontani, A., Wiese, S.: On mathematical programming with indicator constraints. *Mathematical programming* **151**(1), 191–223 (2015)
4. Branke, J., Deb, K., Miettinen, K., Slowiński, R.: *Multiobjective Optimization: Interactive and Evolutionary Approaches*, vol. 5252. Springer-Verlag (2008)
5. Creos Luxembourg S.A.: Le réseau de transport d' électricité de Creos Luxembourg S.A. (2019), <https://www.creos-net.lu/creos-luxembourg/infrastructure/reseau-deelectricite.html>
6. Durstenfeld, R.: Algorithm 235: Random permutation. *Communications of the ACM* **7**(7), 420– (1964)
7. Elgenedy, M.A., Massoud, A.M., Ahmed, S.: Smart grid self-healing: Functions, applications, and developments. In: 1st SGRE. pp. 1–6. IEEE (Mar 2015)
8. Fisher, R.A., Yates, F.: *Statistical tables for biological, agricultural and medical research*. Oliver and Boyd, London, 3rd, rev. and enl. edn. (1949)
9. Golub, G., Reinsch, C.: Singular value decomposition and least squares solutions. *Numerische Mathematik* **14**(5), 403–420 (1970)
10. Gurobi Optimization, L.: *Gurobi Optimizer Reference Manual* (2018)
11. Han, J., Piette, M.: Solutions for summer electric power shortages: Demand response and its applications in air conditioning and refrigerating systems. *Refrigeration, Air Conditioning, & Electric Power Machinery* **29**(1), 1–4 (2008)
12. Hartmann, T., Moawad, A., Fouquet, F., Reckinger, Y., Klein, J., Le Traon, Y.: Near real-time electric load approximation in low voltage cables of smart grids with models@run.time. In: Proc. of SAC 2016. pp. 2119–2126. ACM Press (2016)
13. Pashajavid, E., Shahnia, F., Ghosh, A.: Overload management of autonomous microgrids. In: 11th IEEE PEDS. pp. 73–78 (June 2015)
14. Paton, K.: An algorithm for finding a fundamental set of cycles of a graph. *Communications of the ACM* **12**(9), 514–518 (1969)
15. Ramaswamy, P.C., Deconinck, G.: Relevance of voltage control, grid reconfiguration and adaptive protection in smart grids and genetic algorithm as an optimization tool in achieving their control objectives. In: ICNSC. pp. 26–31 (2011)
16. Shahnia, F., Bourbour, S., Ghosh, A.: Coupling neighboring microgrids for overload management based on dynamic multicriteria decision-making. *IEEE Transactions on Smart Grid* **8**(2), 969–983 (2017)
17. Sichi, J., Kinable, J., Michail, D., Naveh, B., Contributors: *Jgrapht - graph Algorithms and Data Structures in java (Version 1.3.0)*. <http://www.jgrapht.org> (2018)
18. Sifaleras, A., Paparrizos, K., Demyanov, V.F.: Advances in discrete optimization. *Optimization* **62**(8), 1003–1006 (2013)
19. Sifaleras, A., Salhi, S., Brimberg, J. (eds.): *Variable Neighborhood Search - ICVNS 2018, Lecture Notes in Computer Science*, vol. 11328. Springer, Cham (2019)
20. Simão, H.P., Jeong, H.B., Defourny, B., Powell, W.B., Boulanger, A., Gagneja, A., Wu, L., Anderson, R.N.: A robust solution to the load curtailment problem. *IEEE Transactions on Smart Grid* **4**(4), 2209–2219 (2013)
21. Tarjan, R.: Depth-first search and linear graph algorithms. *SIAM Journal on Computing* **1**(2), 146–160 (1972)