Electrical Power and Energy Systems 162 (2024) 110213

Contents lists available at ScienceDirect

International Journal of Electrical Power and Energy Systems

journal homepage: www.elsevier.com/locate/ijepes

Solving the security constrained unit commitment problem: Three novel approaches

Alessandro Francesco Castelli [a], Iiro Harjunkoski [b], Jan Poland [c], Marco Giuntoli [b], Emanuele Martelli [a,*], Ignacio E. Grossmann [d,*]

[a] Politecnico di Milano, Department of Energy, Via Lambruschini 4, Milano 20154, Italy
[b] Hitachi Energy Research, Mannheim, Germany
[c] Hitachi Energy Research, Baden-Dättwil, Switzerland
[d] Department of Chemical Engineering, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, USA

ARTICLE INFO

Keywords:
Unit Commitment
Optimization
Decomposition
N-1 Reliability
Grid Operation

ABSTRACT

This work proposes three novel approaches to speed up the solution of the Security Constrained Unit Commitment problem: an improvement of an active-set iterative approach taken from literature, an approach using solver callback functions for the evaluation of system and security constraints in the branch-and-bound tree, and one based on a shrinking horizon decomposition integrated with the use of callback functions. The three approaches were tested over five different case studies and compared against an approach taken from literature to assess scalability and performance. Results show that the modified iterative approach is always faster than the original one reported in the literature (between −58% and −93% run time), while the callback-based method does not reduce the computational time of large-scale instances. Finally, the shrinking-horizon-based approach was proved to be the fastest (up to −98% less time) despite not guaranteeing optimality (about 1% suboptimal).

1. Introduction

In the last two decades many changes have happened to electricity markets worldwide, mainly due to deregulation. This opened the markets to many operators in the power sector, providing competition and lowering the energy costs. However, this increased participation made grid operation more challenging for Independent System Operators (ISOs). This is because the ISO is in charge of determining the optimal operational scheduling of every generator connected to the grid for a fixed time horizon, with the aim of minimizing the total operational cost of satisfying the demand at each bus (consumption point). The resulting decisions must comply with the generators' operational constraints, as well as with those related to the transmission system (e.g., no lines overload). In addition, N-1 reliability is enforced by guaranteeing that no line will be overloaded under any contingency in which at most one of them is out of service. What we have described is known in the literature as the Security Constrained Unit Commitment (SCUC) problem [1] and it is characterized by high combinatorial complexity, requiring the development of special-purpose mathematical optimization tools for its solution.

The mathematical optimization model involves both linear and non-linear equations describing the generators performance and the AC nature of the grid. This, together with the N-1 reliability consideration in the problem, makes the model hard to solve. One of the first works proposing a method to solve the AC SCUC problem is the one by Fu et al. [2]. In this work the original problem is split into a master and a subproblem, that are solved iteratively. Lagrangean relaxation and Dynamic Programming are used for solving the master problem, while the subproblem finds the solution of the Optimal Power Flow (OPF) problem and checks if any of the system and security constraints are violated. If so, Benders cuts are added to the master problem, corresponding to the violations found. A more recent work proposed by Gupta et al. [3] relies on a similar concept: Generalized Benders Decomposition is used to solve the Mixed Integer Non-Linear Programming (MINLP) model the AC SCUC problem by formulating a Mixed Integer Quadratic Programming (MIQP) master problem and an Non-Linear Programming (NLP) subproblem. As before, the NLP subproblem is used for the evaluation of the violated network constraints to be added to the master problem. Dipan Biswas et al. [4] showed how their two-stage algorithm solving an Mixed Integer Linear Programming (MILP) and an NLP problem sequentially proved to be yield solutions that are only 2 % worse than

* Corresponding authors.
E-mail addresses: emanuele.martelli@polimi.it (E. Martelli), grossmann@cmu.edu (I.E. Grossmann).

https://doi.org/10.1016/j.ijepes.2024.110213
Received 6 March 2024; Received in revised form 12 July 2024; Accepted 27 August 2024
Available online 22 October 2024
0142-0615/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Nomenclature

Abbreviations

AC	Alternating Current
DC	Direct Current
ED	Economic Dispatch
LODF	Line Outage Distribution Factor
LSF	Linear Sensitivity Factor
O&M	Operation and Maintenance
PTDF	Power Transfer Distribution Factor
QP	Quadratic Programming
SCUC	Security Constrained Unit Commitment
SSNC	System and Security Network Constraint
TSO	Transmission System Operator

Sets

Γ', Γ''	Sets of the active system and security network constraints
B	Set of the buses in the network
G	Set of the available generators
T	Set of the timesteps within the horizon considered

Binary Variables

$\delta_{i,t}^{on}, \delta_{i,t}^{off}$	Start-up/shut-down flagger for unit i at time t. 1 if unit i started-up/shut-down at t, 0 otherwise
$z_{i,t}$	On/off status of unit i at time t. 1 if unit i is on at time t, 0 otherwise

Continuous Variables

$d_{b,t}^{shed}$	Demand shed at bus b at time t [MWh]
$d_{b,t}^{shed,QO}$	Demand shed at bus b at time t in the QP subproblem [MWh]
$f_{c,t}^{d,QP}$	Post-outage power flowing through line c at time t resulting from outage of line d in the QP subproblem [MW]
$f_{c,t}^{QP}$	Pre-outage power flowing through line c at time t in the QP subproblem [MW]
$p_{i,t}$	Energy generated by unit i at time t [MWh]
$p_{i,t}^{QP}$	Generation that unit i at time t should have to meet the considered SSNCs in the QP subproblem [MWh]

Parameters

Δt^{adv}	Time advancement of the shrinking horizon algorithm (equal to the number of discrete timesteps considered in each shrunk horizon)
$c_{i,t}, c_i^{on}, c_i^{su}, c_i^{sd}$	Variable [$/MWh], fixed hourly [$/h], stat-up [$/start-up] and shut-down [$/shut-down] operational costs for unit i
c^{shed}	Virtual cost related to shed load [$/MWh]
$c_{b,t}^{d}$	Virtual revenue linked to the amount of demand of bus b at time t that is met [$/MWh]
$D_{b,t}$	Demand related to a timestep duration dt to meet at bus b at time t [MWh]
dt	Timestep duration [h]
$\overline{F}_c$	Maximum power flow allowed through line c [MW]
$ISF_{m,n}^{b}$	ISF element related to power flowing on branch between nodes m, n due to an injection/withdrawal at node b
$LODF_{m,n}^{j,k}$	LODF element related to the redistribution on branch between nodes m, n of the power flowing on the branche between nodes j, k as a consequence of its outage
$\underline{P}_i, \overline{P}_i$	Minimum and maximum allowed generation unit i within dt [MWh]
$\tilde{t}$	Number of original timesteps considered in the aggregated one.

the solution of the original problem. Another work focusing on the loss of optimality associated to model linearization can be found in Jiang et al. [5], where different linearized models are considered and compared.

A methodology based on Ordinal Optimization [6] was proposed by Nan et al. [7]. This iterative approach resembles in the sequential solution of an approximated and detailed model, with the aim to find a "good" solution rather than the optimal one. In the single test case considered it was shown that the computational time was significantly reduced compared to Benders Decomposition, while providing a similar solution.

A tri-level decomposition approach was proposed by Amjady et al. [8], based on finding the unit commitment considering AC line constraints and wind generation worst-case scenario. The methodology proved to be effective in finding a robust solution for the small test case considered, but scalability was not assessed.

As it can be seen, the computational burden associated with solving the original AC SCUC is characterized by the small test instances (e.g., 24, 118 bus systems) considered in the previously cited studies. This gives rise to the need of problem linearization and decomposition. Moreover, most of the works focused on solving the AC SCUC problem consider simple test cases, not providing evidence of the applicability to real world case studies. Furthermore, they are significantly slower than current methods.

For this reason, today's state of the art consists of solving the linearized version of the SCUC problem, which considers Direct Current (DC). The resulting DC SCUC problem is described by means of an MILP model that is easier to solve. The non-linear security and network constraints are substituted by linear inequalities described by sensitivity coefficients that are grouped in the Power Transfer Distribution Factors (PTDF) and Line Outage Distribution Factors (LODF) matrices. For a DC system, the PTDF matrix describes how the power supplies/withdrawn to/from each bus flows across each line, while the LODF matrix described how the power flowing on a certain line is diverted on the others in case of contingency. Different methods for the evaluation of these matrices can be found in the literature, aiming at improving the computational performance during their computation (e.g., [9,10;11]) or at describing more complex systems comprising both AC and DC lines (see [12] for an example).

Despite the reduction in the computational burden achieved by considering the DC SCUC problem, the model still features a very large number of system and security constraints (e.g., given a number of timesteps t and a number of lines L, this number is equal to $t{*}L(L+1){*}2$ – see Appendix for more). This makes the problem extremely hard to solve due to the memory requirements associated with the model size, therefore highlighting the need of developing special purpose solution approaches to address the problem. In their work, Xavier et al. [13] proposed an iterative approach that consists of solving an initial MILP problem with no system and security constrains, evaluating the OPF based on the resulting injections, checking constraint violation and adding them to the model. In this way only the active network constraints are added to the model, which is a smaller subset, and therefore requires significantly less memory. While in this method the new active constraints are "learned" as new solutions are evaluated, other approaches are used to filter out in advance the active constraints. In a different work [14] the same authors proposed different machine learning tools for the prediction of the active network constraints and a starting solution based on past, optimal instances. This data driven approach was able to decrease the size of the problem, while speeding up the solution time, while still guaranteeing optimality. Other similar approaches have been developed with the aim of predicting the correct pool of active constraints so as to avoid any redundancy. More recent approaches propose other data-driven filtering methods to improve the overall solution time. Yang et al. [15] used stacked denoising autoencoders (deep neural network used for dimensionality reduction) during the training phase to evaluate the non-linear relationship between active constraints and the operational solutions. An expanded sequence-to-sequence (E-Seq2Seq)-based data-driven SCUC expert system for dynamic multiple-sequence mapping samples is proposed by Yang et al. [16], with the aim of providing a solution to the SCUC problem based on trained data and avoiding the optimization process.

Other, non-data-driven approaches can be found in the literature as well, like the cost-driven screening method proposed by Porras et al. [17] where valid inequalities are introduced to tighten the feasible region of the relaxed SCUC LP problem. The relaxed solution is then used for evaluating the active constraints and discard the redundant ones, thus being independent from any training phase (unlike the previous methods). As a general statement, many works in the literature show how applying these constraint filtering approaches to iterative solution methods can significantly reduce the computational time.

Finally, some well-known works address the issue of solving the SCUC under uncertainty (e.g., due to the increased penetration of non-dispatchable renewable energy sources). For example, Wang et el. [18] propose an iterative approach to solve a two-stage stochastic SCUC with different forecast scenarios related to wind generation. The deterministic commitment solution of the master problem is evaluated over a set of wind generation scenarios to check violations of the network constraints and Benders cuts are added if they occur. Another remarkable work by Bertsimas et al. [19] proposes an adaptive robust optimization model for solving the SCUC problem. The two-level method is based on a Benders decomposition type algorithm that is able to evaluate a robust solution for any realization of the uncertain demands at each bus within a predefined, deterministic uncertainty set. In another paper by Wang et al. [20] a specific decomposition algorithm is proposed for the parallel solution of a stochastic SCUC characterized by uncertain renewable energy generation. Lagrangean decomposition is applied for solving the stochastic problems where the uncertain parameters are the generators and lines' reliability. In the work by Wu et al. [21], the authors consider line outage scenarios instead of the N-1 constraints, thus building a large scenario tree.

Despite their capability of including the uncertainties of the parameters in the problem, the approaches above mentioned require long computational times, which is not suitable for industrial practice (ISOs usually need to update the generators' schedule multiple times throughout the day, e.g., on an hourly basis). Given this requirement, and by considering that updating the solution frequently is an effective method to cope with forecast uncertainty, effort should be placed in solving the DC SCUC problem as quickly as possible. One way to achieve this is by adopting data-driven network constraints filtering tools (as the ones cited before), since they have proven to be effective on relatively small test cases. However, they require large dataset for the training phase. Since for large systems this data is either not available or the evaluation of optimal solutions requires a long time (sometimes the problem is even intractable), this approach might become impractical.

For these reasons, this work proposes three novel solution approaches aimed at improving the solution time of the DC SCUC problem. The first one is an improvement of the method presented in Xavier et al. [13], the second one consists of the use of solver callback functions for the addition of the network constraints as Lazy Constraints to the model (similar to what done by Liu et al. in [22]), and the last one is a Shrinking Horizon algorithm (similar to approaches found in stochastic Model Predictive Control [23] and scheduling [24] problems) featuring callback functions for contingency and system constraint discovery. Contrary to the previously mentioned data-driven methods (which largely rely on Machine Learning), the ones proposed in this work are based on modern optimization-based approaches that do not make them dependent on training datasets while guaranteeing the feasibility of the predicted solutions. The three novel features of the proposed methods are: (a) using a more sophisticated constraint filtering method that is able to reduce the computational time needed by the new iterative approach; (b) embedding the filtering approach within the solver's branch-and-cut algorithm; (c) a shrinking-horizon approach featuring non-uniform time discretization that embeds as adaptation the filtering method in approach (b) for operational feasibility.

2. Problem statement

As previously mentioned, the problem addressed in this paper is known in literature as Security Constrained Unit Commitment problem, and it consists of evaluating the operational schedule of each generation unit available on the considered power grid (which unit to be turned on/off and its generation level) with the aim of meeting the demand at each bus of the network while avoiding overload on any transmission line under nominal condition, and for whichever contingency in which at most one line is out of service (N-1). The problem can be therefore stated as follows:

Given:

- The set of generation units and their operational parameters (e.g., generation bounds, ramping limits),
- The set of network components and their operational parameters (e.g., transmission line limits),
- The network topology,
- The specified time horizon,
- The expected energy demand at each bus,
- The status of the units and transmission lines at the beginning of the considered time-horizon (on/off and current power generation),

Find the optimal commitment (schedule) of the units, which minimizes the total generation cost/maximizes the net operational revenue subject to the following constraints:

- Units' operational limits (maximum and minimum allowed load, ramping limits, minimum up- and down-times)
- Energy balances at each node
- Transmission limits on each line
- Reserve constraints
- N-1 contingency constraints

3. Methodology

The proposed methodology consists of two main steps: (1) the formulation of a tight and efficient MILP model for the SCUC problem, (2) the development of algorithms specifically designed to reduce the computational burden of finding its optimal solution. The three different solution approaches are presented in this section: the first one is a variation of the well-known iterative method based on the active set-strategy and presented in [13], the second approach uses solver's callback functionalities for the evaluation and addition of security and safety constraints, and the third one is a Shrinking Horizon algorithm inheriting the same callback functions featured in the third approach.

3.1. Mixed integer linear programming model

The SCUC problem considers a linearized (DC) version of the original non-linear problem and therefore can be described with a Mixed Integer Linear Programming (MILP) model. The network and generation units are modelled by means of linear constraints, divided into the following four groups: generation units operational constrains, reserve constraints, energy balance constraints and network security and systems constraints. Since the model shares many of the features belonging to other works in literature, the references related to the model's constraints are here shown. The set of constraints in the MILP model can be divided into duration constraints, ramping limits, reserve constraints, balance constraints and system and system and security network constraints. The duration constraints are taken from Morales-España et al. [25], the ramping limits come from Constante-Flores et al. [26], and reserve constraints were formulated starting from the ones found in Tejada-Arango et al. [27]. A more detailed description can be found in the Appendix where above listed constraints are shown.

In this study, two different objective functions were considered for the sake of evaluating the impact on the solution and computational time of minimizing the operational cost or maximizing the social welfare. These aspects are present in the next subsection.

- Objective function

The typical objective function to minimize in a SCUC problem is the total operational cost over the specified time planning horizon, namely:

$$OF1 = \sum_{i\in G}\sum_{t\in T}\left(c_{i,t}p_{i,t} + c_i^{on}z_{i,t} + c_i^{su}\delta_{i,t}^{on} + c_i^{sd}\delta_{i,t}^{off}\right) + c^{shed}\sum_{b\in B}\sum_{t\in T}d_{b,t}^{shed} \tag{1}$$

where G is the set of generation units, T the set of timesteps in the horizon, B the set of buses in the network, $c_{i,t}p_{i,t}$ is the term related to the variable Operation and Maintenance (O&M) cost based on generation output $p_{i,t}$ for unit i at time t, $c_i^{on}z_{i,t}$ the term related to the fixed O&M hourly cost (being $z_{i,t}$ the binary on/off variable), $c_i^{su}\delta_{i,t}^{on}$ the start-up cost (being $\delta_{i,t}^{on}$ the binary start-up flagger variable), $c_i^{sd}\delta_{i,t}^{off}$ the shut-down cost (with $\delta_{i,t}^{off}$ the binary shut-down flagger variable) and $c^{shed}d_{b,t}^{shed}$ the virtual cost linked to the amount of shed load ($d_{b,t}^{shed} \in [0; D_{b,t}]\forall b \in B, \forall t \in T$, with $D_{b,t}$ the expected demand at node b at time t). Note that in this case c^{shed} must be large enough in order to minimize the unmet demand. While $OF1$ is considered as the reference objective function for this study, another objective function $OF2$ is considered with the aim of assessing the impact on the computational time and solution:

$$OF2 = \sum_{i\in G}\sum_{t\in T}\left(c_{i,t}p_{i,t} + c_i^{on}z_{i,t} + c_i^{su}\delta_{i,t}^{on} + c_i^{sd}\delta_{i,t}^{off}\right) - \sum_{b\in B}\sum_{t\in T}c_{b,t}^{d}\left(D_{b,t} - d_{b,t}^{shed}\right) \tag{2}$$

Here the term $c_{b,t}^{d}\left(D_{b,t} - d_{b,t}^{shed}\right)$ represents the virtual revenue gained by serving the demand. By setting the parameter $c_{b,t}^{d}$ large enough the optimal solution will maximize the amount of served demand at the minimum cost (similarly to maximizing the social welfare). Of course, when considering $OF2$, the upper bound $d_{b,t}^{shed} \leq D_{b,t}$ must be set ($\forall b \in B$, $\forall t \in T$).

3.2. Solution approaches

By looking at Table 3 it can be noted how the size of the SCUC MILP model is very large even for small size problems (e.g., ieee118). This is because of the presence of SSNCs. For this reason, different solution approaches have been developed in this study, with the aim of solving the SCUC problem faster. The solution approaches considered in this work are four: one is taken from literature and used as benchmark while the other three contain elements of novelty. The first method (M1) is taken from literature and considered as reference [13]. It consists of an iterative approach that adds security constraints based on the violation of the current solution. The second method (M2) is a novel variation of M1 which includes additional features for the prediction of new violated constraints. The third one (M3) is a callback-based method, which integrates the features of M1 by means of solver callback functions. Finally, a novel shrinking horizon approach with non-uniform time discretization is presented as fourth solution approach (M4).

In all the proposed methods the initial optimization model does not consider any of the system and security network constraints, as these are added during the optimization algorithm. This is done to limit the memory requirements and given the fact that most of the times only a small subset is needed (as previously mentioned in Section 1). However, the addition of network constraints ensures that the optimal solution found is feasible and optimal for the original SCUC model too. It is worth noting that although all the proposed approaches do not consider any Umbrella Constraints Discovery algorithm (e.g., [28]), one can still be added for the sake of evaluating the starting set of system and contingency constraint.

Finally, although each solution approach was developed to solve the deterministic version of the SCUC problem, they can be adapted to tackle both its Robust and Stochastic versions. However, this might need the development of a nested approach where an inner algorithm is used to solve the Robust/Stochastic MILP.

3.2.1. Reference approach: Iterative method based on constraint violation (M1)

The method considered as reference approach can be found in Xavier at el. [1314] and the flowchart describing the algorithm can be seen in Fig. 1 (left). This solution approach is based on an active-set strategy that aims at finding new System and Security Network Constraints (SSNCs) iteratively.

At first, a SCUC model with no SSNCs is loaded (namely an MILP defined as for Eqs. (A.1)-(A.23) – see Appendix). Then, the empty sets Γ' and Γ'' are defined and added to the model. These sets will contain those SSNCs (described by Eqs. (A.27)-(A.30) – see Appendix) that results being violated in following iterations. In particular, Γ' and Γ'' refers to the pre- and post-contingency flow limits respectively. Please note that since these sets belong to the MILP model, their constraints will define its feasible.

At each iteration, once a Unit Commitment solution is found (which comprise the optimal nodal injections and withdrawals), the pre- and post-contingency power flows on each branch ($f_{c,t}$ and $f_{c,t}^{d}$) are evaluated by using the PTDF and LODF matrices respectively. Based on these, it is determined whether the flow limits at each timestep are met. If not, the magnitude of violation is computed (namely, the difference between the pre- and post-contingency power flows coming from the current solution and their limits), all violated SSNCs in the considered horizon are ranked accordingly and the one with the highest violation is added to the set Γ' (if the violation is related to a pre-contingency flow) or Γ'' (if the violation is related to a post-contingency flow). Then, the k SSNCs with greatest infringement per each time-step are added to Γ' and Γ''. Please note that the authors of [13] and [14] suggested k between 5 and 15, depending on the network considered. In this study k was set equal to 15 for all cases for the sake of consistency (no sensitivity on this parameter was performed). The reason why not all violated constraints are added to the model at each iteration is because numerical results show how only a fraction of them are actually needed (as seen in [28;29]).

Once Γ' and Γ'' are updated, the MILP is solved again and a new solution is found. The pre- and post-contingency flows are computed, and the flow limits are checked for those lines and timesteps that are not yet present in Γ' and Γ''. If any violation occurs, constraints are added again (as mentioned above) up until a solution with no violation is obtained. It is important to highlight that the final solution is optimal for both M1 and the original SCUC model (comprising all the SSNCs). This is true since the solution in M1 checks the optial solution against all the SSNCs belonging to the original problem (see [13,14] for more).

Regarding the MILP model and the sets Γ' and Γ'', as the algorithm progresses and new constraints are appended to the sets, the size of the MILP model increases and become more difficult to solve. However, as it will be shown Section 4 and 5, the size of the MILP at the last iteration features a number of constraints greatly smaller than the original one,

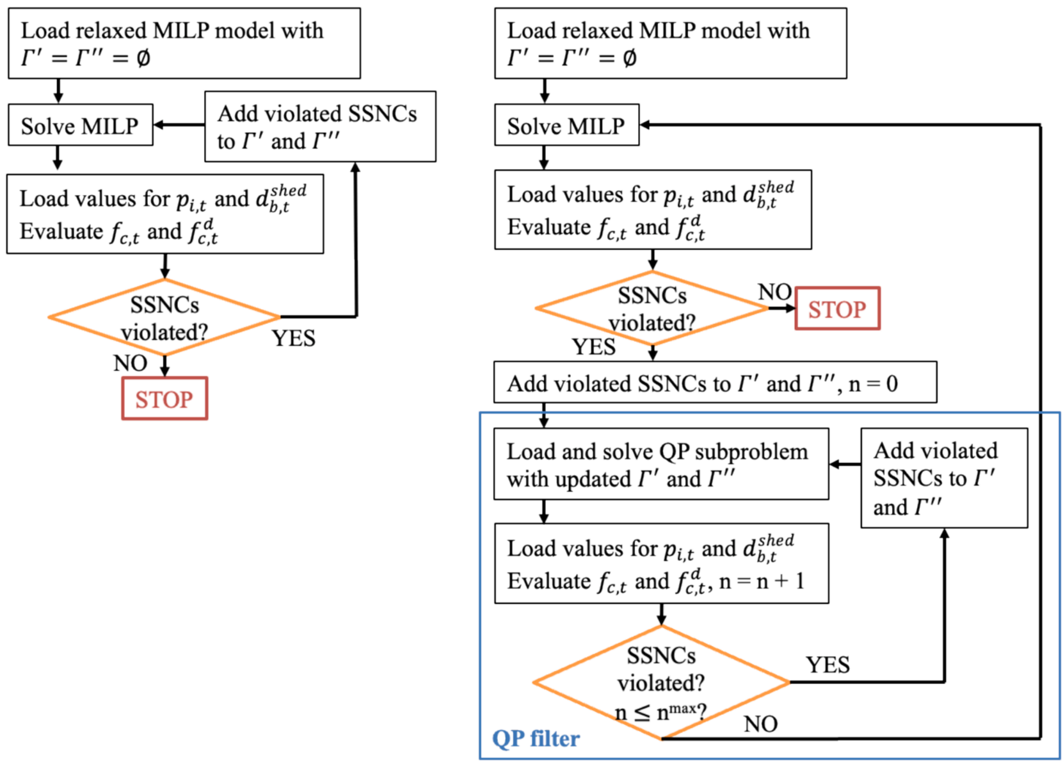

Fig. 1. On the left: flowchart representing the solution approach M1. On the right: flowchart representing the solution approach M2. SSNCs are added to the sets of active constraints Γ′ and Γ″ based on how the current solution violate them.

thus hugely easing the computational burden.

3.2.2. Iterative method based on QP filter (M2)

The approach presented here shares many of the features seen in the previous one, such as its iterative nature and the progressive addition of SSNCs to the MILP model. The main difference lies in the way these constraints are added. While in the previous approach the SSNCs are selected according to the outcome of just the MILP problem, here their evaluation depends also on the results of an operational subproblem modelled as a Quadratic Program (QP).

The flowchart of the algorithm can be seen in Fig. 1 (right). As for M1, a SCUC model with no SSNCs constraints (from here on called "master" problem) is defined, as well as the sets Γ′ and Γ″. At each iteration, the master problem is solved, power flows are evaluated, checked against the branch flow limits and added to Γ′ and Γ″ based on violation. Then, instead of just moving to the next iteration, a further update of Γ′ and Γ″ is performed by repeatedly solving a QP subproblem. Please note that in this study 20 iterations are considered given the preliminary computational outcomes. Regarding the constraint addition strategy, k was set to 15 for both the master problem (as of M1) and for each iteration when the subproblem is solved.

Before getting into the details on why this QP subproblem was considered, its mathematical formulation is presented.

$$min \sum_{i\in G, t\in T} \left(p_{i,t}^{QP} - \widetilde{p}_{i,t}\right)^2 \tag{3}$$

s.t.

$$\sum\nolimits_{i\in G} p_{i,t}^{QP} = \sum\nolimits_{b\in B} \left(D_{b,t} - d_{b,t}^{shed,QP}\right), \forall t \in T \tag{4}$$

$$0 \leq p_{i,t}^{QP} \leq \overline{P}_i, \forall i \in G, \forall t \in T \tag{5}$$

$$\begin{aligned} f_{c,t}^{QP} = & \sum_{b\in B, i\in G_b^B, G_b^B\neq\emptyset} ISF_{m,n}^{b} \cdot \frac{p_{i,t}^{QP}}{dt} \\ & - \sum_{b\in B} ISF_{m,n}^{b} \cdot \left(\frac{D_{b,t} - d_{b,t}^{shed,QP}}{dt}\right), m,n = y(c), \forall c, t \in \Gamma' \end{aligned} \tag{6}$$

$$f_{c,t}^{d,QP} = f_{c,t}^{QP} + LODF_{m,n}^{j,k} \cdot f_{d,t}^{QP}, m,n = y(c), j,k = y(d), \forall c,d,t \in \Gamma'' \tag{7}$$

$$-\overline{F}_c \leq f_{c,t}^{QP} \leq \overline{F}_c, \forall c, t \in \Gamma' \tag{8}$$

$$-\overline{F}_c \leq f_{c,t}^{d,QP} \leq \overline{F}_c, \forall c, d, t \in \Gamma'' \tag{9}$$

with $\widetilde{p}_{i,t}$ the power generation of unit i at time t and $\widetilde{d}_{b,t}^{shed}$ the shed demand at node b at time t coming from the solution of master SCUC problem (they are parameters), $p_{i,t}^{QP}$ the power generation variable of the QP subproblem related to unit i at time t, $d_{b,t}^{shed,QP}$ the shed demand variable related to the subproblem for bus b at time t, $\overline{P}_i$ the generation upper bound for unit i, $f_{c,t}^{QP}$ the pre-contingency power flow variable on branch c evaluated by means of the Injection Shifting Factors $ISF_{m,n}^{b}$ extracted from the PTDF matrix (see Appendix), $f_{c,t}^{d,QP}$ the post-contingency power flow variable on branch c due to an outage on branch d at time t and computed by means of the LODF matrtix (see Appendix), and $\overline{F}_c$ the flow limit on branch c. Please note that $y(c)$ and $y(d)$ refer to the maps linking the lines c and d with the respective nodes m,n and j,k.

By looking at the QP subproblem, the very first thing to note is the absence of binary variables. Then, the only constraints related to generation and demand are the global power balance (Eq. (4)) and the units' generation upper bound (Eq.(5)). No complicating constraints (e.g., ramping limits and duration constraints) are present, making the model significantly smaller in size and easier to solve. Pre- and post-contingency power flow limits are imposed by Eqs. (6)-(9) and depend on Γ′ and Γ″ (thus the constraints added in the previous iterations and those ones identified by solving the master problem at the current iteration). By looking at the objective function, the model aims at finding the closest generation schedule to the one of the master problem. This is done by minimizing the squared Euclidean distance $\sum_{i\in G,t\in T}\left(p_{i,t}^{QP}-\widetilde{p}_{i,t}\right)^2$ between the master and subproblem solution. In particular, $p_{i,t}^{QP}$ is the very first generation schedule compliant with the most updated set of active SSNCs cutting off $\widetilde{p}_{i,t}$ from the feasible region.

Given what already mentioned, it is now possible to understand why the QP subproblem is added to the algorithm, and what is the relationship between it, the master problem and the Γ′ and Γ″ sets. At first, it is important to highlight that both the master MILP and QP models share the same set of SSNCs in Γ′ and Γ″. Therefore, each time new constraints are added to these sets (either by performing a power flow analysis on the master or subproblem), both the models are updated with the same exact active SSNCs. By repeatedly solving the QP subproblem (from here on called “QP filter”) a portion of the active SSNCs related to the neighboring solutions of the current master problem’s one are found. Therefore, the QP filter allows to find a larger pool of SSNCs, providing a prediction of potentially active constraints for the following iterations. This should allow to reduce the total number of iteration and execution time.

As for M1, the solution found by M2 is optimal too. In fact, the additional SSNCs added by the QP filter at each iteration belong to the pool of network constraints of the original SCUC model. Since the final solution of M2 checks all the SSNCs, the solution is optimal for the original problem.

Finally, there are two main drawbacks with this approach. At first, if the SSNCs violated by the master MILP model are sufficient to find a new feasible integer solution that is substantially different than the previous one, the effort spent in solving multiple times the QP subproblem may not be worthwhile. Then, the solution found by the QP subproblem might be infeasible when applied to the SCUC model since the QP model does not include any of the operational constraints of the original one. That is why the QP model solved for a limited number of times: to avoid adding useless SSNCs linked to unfeasible solutions.

3.2.3. Solver callback-based method (M3)

The third method developed in this paper is an alternative implementation of M1. A different software approach was considered with the aim of investigating its computational impact.

In M3, which finds similarities with what presented in refs. [22] and [30], SSNCs are checked and dynamically added to the model by a solver callback function [31] as new integer solutions are found by the solver’s Branch-and-Cut algorithm. This allows to launch the solver just once thus avoiding all the memory allocations linked to running an iterative method (e.g., loading the model, solving it, loading the solution and checking it, etc. – per each iteration). The reason why M1 was chosen over M2 is because the latter requires to solve an inner QP subproblem that cannot be included in a callback function of any commercially-available solver at this time.

The way the method and callback function work can be seen in Fig. 2 (left). As for M1, a relaxed version of the SCUC model (MILP with constraints defined by Eqs. (A.1)-(A.23) – see Appendix) is considered with no SSNCs (empty sets Γ′ and Γ″ – see method M1). The model is loaded by the solver, and the Branch-and-Cut algorithm is started. Each time a new incumbent solution is found, the callback function is invoked by the solver. Thus, the numerical values of each variable of the new solution are loaded, the pre- and post-contingency flows are evaluated and the SSNCs are checked. The ones that are violated the most are added to Γ′ and Γ″ as for M1 (here too k is set to 15). Differently from the M1 and M2, constraints are added to the MILP as “Lazy Constraints” (which are constraints that are dynamically added to the model during the optimization process based on the violation of the current solution, e.g., [32]), with the aim of reducing the memory requirements. A solution is found when the MIP gap is below the given limit and no SSNCs are violated.

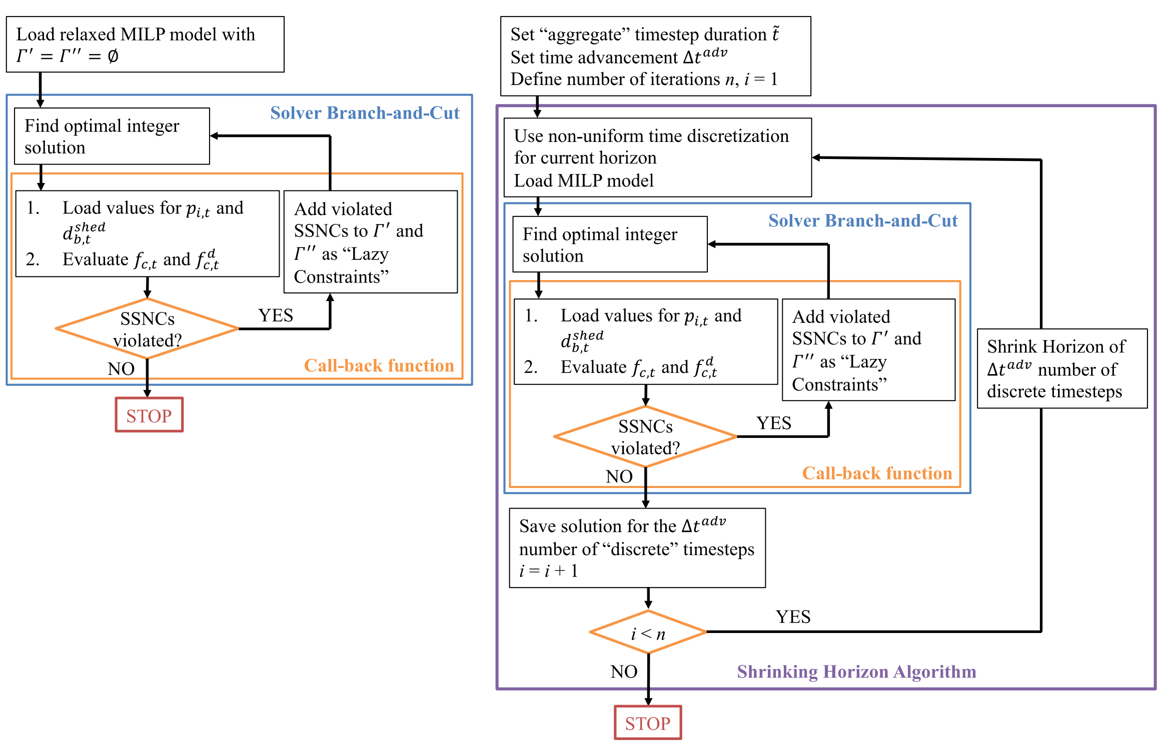

Fig. 2. On the left: flowchart representing the solution approach M3. On the right: flowchart representing the solution approach M4. In both M3 and M4 the solution is found by launching the solver just once. The SSNCs are added to the MILP model by means of a call-back function as solutions are found during the execution of the branch-and-cut algorithm.

3.2.4. Shrinking-horizon based method with solver callback (M4)

As previously mentioned, considering all SSNCs in the SCUC model makes it hard to solve, requiring a long computational time that is not in line with the requirements of most TSOs.

Even if custom approaches are considered (as those presented in the previous sections), the total solution time could be longer than the desired one (e.g., 15–30 min). For this reason, a novel temporal decomposition approach inspired by the Shrinking Horizon algorithm (Balasubramanian and Grossmann [24]) and based on non-uniform time discretization is considered.

These two concepts are here described separately. At first, the Shrinking Horizon is a solution approach based on solving the same horizon in multiple runs. Starting from the first iteration, the original horizon is considered and a solution if found. Then, the solution belonging to the first Δt^{adv} periods (advancement periods) is stored, and a new horizon shrunk by those Δt^{adv} ones is considered for the next run. In addition, all variables not belonging to the Δt^{adv} timesteps can be relaxed to ease calculation.

Despite its ability to solve the "shrunk" problems faster (thanks to their reduce dimension), in the SCUC problem it is important to retain the integer nature of the variables due to the importance of time-linking constraints (e.g. duration constraints). Therefore, as seen in other works (e.g., [33]), at the first iteration the problem will feature its full horizon (same complexity as the original problem). To solve this issue, non-uniform time discretization is introduced. The idea is to have a coarser time discretization for part of the horizon, reducing the number of timesteps and therefore the size of the problem (both variables and constraints). This approach is motivated by assuming that decisions at the beginning of the horizon are slightly influenced by the ones at the end of it. It is important to point out how the binary nature of some variables is kept in all timesteps (whether they are aggregated or not) such that the model does not degenerate (although some accuracy is lost). The block-flow diagram of algorithm M4 is shown in Fig. 2 (right).

In this work, the non-uniform discretization was implemented in the following way. At first, the original timestep discretization dt is considered and the values $\Delta t^{adv}, \widetilde{t} \in \mathbb{N}$ are defined. At each iteration, the time discretization of current horizon is modified. By considering T the number of timesteps of original duration dt, the first Δt^{adv} periods are kept as-is while the remaining $T - \Delta t^{adv}$ are grouped into blocks of $\widetilde{t}$ timesteps. Of course, $T - \Delta t^{adv}$ must be divisible by $\widetilde{t}$, which must be a multiple of Δt^{adv}. Therefore, the horizon with uniform discretization dt goes from having T timesteps to $\Delta t^{adv} + \left(T - \Delta t^{adv}\right)/\widetilde{t}$, of which Δt^{adv} with discretization dt and $T - \Delta t^{adv}$ with discretization $\widetilde{t}$. This non-uniform discretization requires to modify the time series of the parameters by either summing or averaging (e.g., demand of $\widetilde{t}$ timesteps is summed, fuel cost if averaged).

Regarding the model, at each iteration the relaxed model with constraints defined by Eqs. (A.1)-(A.23) (see Appendix) and no SSNCs is considered (as for the previous methods, Γ' and Γ'' are empty). Given the new time discretization, time-linking constraints needs to be remodeled with the aim of being consistent with the original formulation (e.g., time duration and ramping constraints) and keeping the integer nature of variables. More details on this are introduced in the following subsection and a more detailed description on how operational constraints were evaluated can be found in Appendix.

Then, SSCNs are added by means of the same callback function presented in method M3 (here too k is set to 15): this ensure that the solution found is feasible according to system and N-1 constraints.

A schematic representation of the implemented decomposition can be seen in Fig. 3, where the values used for the shrinking horizon advancement and modified discretization are used in this study ($\Delta t^{adv} = 4, \widetilde{t} = 4$ over an original time horizon of 24 one-hour timesteps).

At each run, the first Δt^{adv} original timesteps are kept, while the remaining ones are reorganized by means of aggregation in blocks of $\widetilde{t}$. This results in a significant reduction in the number of considered timesteps (and thus problem size). For example, at the first iteration only nine timesteps are considered (against the 24 of the original problem). In addition, given these values of Δt^{adv} and $\widetilde{t}$, only six iterations are needed to evaluate the 24-timestep solution.

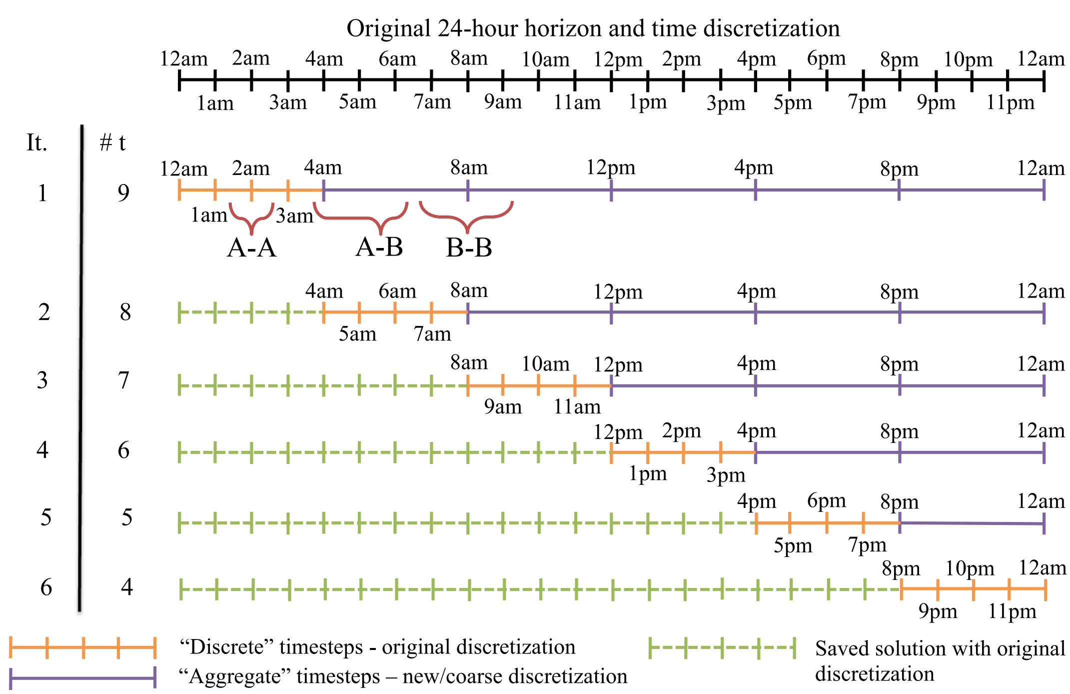

Fig. 3. Schematic representation of the Shrinking Horizon algorithm presented in this study. The horizon considered in each iteration is the one comprising orange and purple timesteps: the former retain the same time discretization of the original time series within the considered horizon, while the latter consider a more coarse resolution (aggregating block of four intervals).The green dashed line represent the solution saved in the previous iteration. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Given the abovementioned description, it can be understood how combining a shrinking horizon approach with non-uniform time discretization can help finding a good solution in a reduced amount of time.

However, it is important to mention that two drawbacks are present. At first, the solution evaluated by M4 has no guarantee of optimality. In fact, the solution found by the algorithm is obtained by combining the decisional outcome of the first Δt^{adv} timesteps of each sub-horizon. Due to the re-discretization happening at each of these sub-horizon, a loss of information is present and the relative solution might be sub-optimal. Secondly, the problem reduction requires a careful model reformulation and parameter evaluation. However, thorough understanding of the problem is sufficient to overcome this problem.

- Modelling of ramping constraints

In this subsection the reformulated ramping limits for method M4 are presented. All mathematical steps to get the following equations have been skipped for the sake of conciseness but can be found in the Appendix.

Ramping limits represent the maximum increase and decrease of a units' generation between adjacent timesteps. By looking at Fig. 3, three different couples of adjacent timesteps can be identified: A-A, A-B and B-B. A-A refers to a couple of discrete timesteps (retaining the original timestep duration), A-B to a couple where an aggregate timestep follows a discrete one, and B-B to a couple of aggregate timesteps. Since the aggregate and discrete timestep are characterized by a different timestep duration, also the maximum allowed increase/decrease in generation among adjacent time intervals will change.

The formulation of the ramping constraints considered for the general couple of adjacent timesteps of type X and Y is the following:

$$p_{i,t} - p_{i,t-1} \leq RU_i^{X-Y} z_{i,t-1} + SU_i^{X-Y} \delta_{i,t}^{on}, \forall i \in G, \forall t \in T^*, t \in T^{**}, t-1 \in T^{***} \tag{10}$$

$$p_{i,t-1} - p_{i,t} \leq RD_i^{X-Y} z_{i,t} + SD_i^{X-Y} \delta_{i,t}^{off}, \forall t \in T^*, t \in T^{**}, t-1 \in T^{***} \tag{11}$$

with $p_{i,t}$ the power generated by unit i at time t, and $z_{i,t}$ the binary on/off variable.

The values related to the parameters $RU_i^{X-Y}, RD_i^{X-Y}, SU_i^{X-Y}, SD_i^{X-Y}$ and the sets T^*, T^{**} and T^{***} depend on the whether a pair of timestep A-A, A-B or B-B is considered. Before describing these values, the following parameters needs to be defined:

$$\#RU_i = \left\lfloor \frac{\overline{P}_i - \underline{P}_i}{RU_i} \right\rfloor \tag{12}$$

$$\#SU_i = \left\lfloor \frac{\overline{P}_i - SU_i}{RU_i} \right\rfloor \tag{13}$$

$$\#RD_i = \left\lfloor \frac{\overline{P}_i - \underline{P}_i}{RD_i} \right\rfloor \tag{14}$$

$$\#SD_i = \left\lfloor \frac{\overline{P}_i - SD_i}{RD_i} \right\rfloor \tag{15}$$

with $\underline{P}_i$ and $\overline{P}_i$ the generation lower and upper bounds, RU_i and SU_i the upward ramping limits while operating and during the start-up, RD_i and SD_i the downward ramping limits while operating and during shut-down unit i. $\#RU_i$ is the minimum number of discrete timesteps needed to augment the unit's output from minimum to maximum; $\#SU_i$ is the minimum number of timesteps to increase the output from SU_i to $\overline{P}_i$; $\#RD_i$ the minimum number of discrete timesteps to reduce the unit's output from maximum to minimum load; $\#SD_i$ the minimum number of timesteps to decrease the output from its maximum value to SD_i.

Given this, the RU_i^{X-Y} and SU_i^{X-Y} can be defined for each pair for timesteps. For the pair A-A, $RU_i^{A-A} = RU_i$ and $SU_i^{A-A} = SU_i$. For A-B and B-B:

$$RU_i^{A-B} = \underline{P}_i\left(\min\left(\tilde{t}, \#RU_i\right) - 1\right) + \overline{P}_i \max\left(0, \tilde{t} - \#RU_i\right) + \sum_{n=1}^{\min\left(\tilde{t}, \#RU_i\right)} n \cdot RU_i \tag{16}$$

$$RU_i^{B-B} = \underline{P}_i\left(\min\left(\tilde{t}, \#RU_i\right) - \tilde{t}\right) + \overline{P}_i \max\left(0, \tilde{t} - \#RU_i\right) + \sum_{n=1}^{\min\left(\tilde{t}, \#RU_i\right)} n \cdot RU_i \tag{17}$$

$$SU_i^{A-B} = SU_i^{B-B} = SU_i\left[1 + min\left(\tilde{t} - 1, \#SU_i\right)\right] + \sum_{n=1}^{min\left(\tilde{t}-1, \#SU_i\right)} n \cdot RU_i + \overline{P}_i\left[\tilde{t} - min\left(\tilde{t} - 1, \#SU_i\right) - 1\right] \tag{18}$$

The same can be done for RD_i^{X-Y} and SD_i^{X-Y}. For A-A, $RD_i^{A-A} = RD_i$ and $SD_i^{A-A} = SD_i$, while for A-B and B-B:

$$RD_i^{A-B} = \overline{P}_i\left(1 - \min\left(\tilde{t}, \#RD_i\right)\right) - \underline{P}_i \max\left(0, \tilde{t} - \#RD_i\right) + \sum_{n=1}^{\min\left(\tilde{t}, \#RD_i\right)} n \cdot RD_i \tag{19}$$

$$RD_i^{B-B} = \overline{P}_i\left(\tilde{t} - \min\left(\tilde{t}, \#RD_i\right)\right) - \underline{P}_i \max\left(0, \tilde{t} - \#RD_i\right) + \sum_{n=1}^{\min\left(\tilde{t}, \#RD_i\right)} n \cdot RD_i \tag{20}$$

$$SD_i^{A-B} = SD_i \tag{21}$$

$$SD_i^{B-B} = SD_i + \overline{P}_i\left[\tilde{t} - min\left(\tilde{t} - 1, \#SD_i\right) - 1\right] + \sum_{n=1}^{min\left(\tilde{t}-1, \#SD_i\right)} \left(\overline{P}_i - n \cdot RD_i\right) \tag{22}$$

Regarding the sets T^*, T^{**} and T^{***}, these are defined in Table 1 (with T^{disc} the set of timesteps retaining the original duration and. T^{agg} the set of timestepd that have been re-discretized).

- Modelling of minimum up- and down-time constraints

As for ramping limits, here just the equations are presented, and further details are found in the Appendix. The reformulated minimum up- and down-time constraints considered are the following:

$$\sum_{\tau=t-\widetilde{UT}_{i,t}+1}^{t} \delta_{i,\tau}^{on} \leq z_{i,t}, \forall i \in G \backslash G^{on}, \forall t \in \left[\widetilde{UT}_i^{agg}, \cdots, |T|\right] \tag{23}$$

$$\sum_{\tau=t-\widetilde{DT}_{i,t}+1}^{t} \delta_{i,\tau}^{off} \leq 1 - z_{i,t}, \forall i \in G \backslash G^{on}, \forall t \in \left[\widetilde{DT}_i^{agg}, \cdots, |T|\right] \tag{24}$$

with $\delta_{i,\tau}^{on}$ and $\delta_{i,\tau}^{off}$ the start-up and shut-down binary flaggers for unit i at time τ, and $T = T^{disc} \cup T^{agg}$ the set of all timesteps in the horizon.

Four new parameters are introduced: $\widetilde{UT}_{i,t}$, $\widetilde{DT}_{i,t}$, $\widetilde{UT}_i^{agg}$ and $\widetilde{DT}_i^{agg}$. At first, $\widetilde{UT}_i^{agg}$ and $\widetilde{DT}_i^{agg}$ are defined as the minimum number of discrete and aggregate timesteps corresponding to the original UT_i and DT_i. As a consequence, if $UT_i \leq \Delta t^{adv}$ and $DT_i \leq \Delta t^{adv}$, then $\widetilde{UT}_i^{agg} = UT_i$ and $\widetilde{DT}_i^{agg} = DT_i$. On the other hand, if $UT_i > \Delta t^{adv}$ and $DT_i > \Delta t^{adv}$ then the following expressions hold:

$$\widetilde{UT}_i^{agg} = \Delta t^{adv} + \left\lceil \frac{UT_i - \Delta t^{adv}}{\tilde{t}} \right\rceil \tag{25}$$

Table 1
Definition of T^*, T^{**} and T^{***} for the different pairs of timesteps.

	A-A	A-B	B-B
T^*	$\forall t = 2, \cdots, \lvert T^{disc} \rvert$	$\{min(t) \in T^{agg}\}$	$t : t > min(t) \in T^{agg}$
T^{**}	T^{disc}	T^{agg}	T^{agg}
T^{***}	T^{disc}	T^{disc}	T^{agg}

$$\widetilde{DT}_i^{agg} = \Delta t^{adv} + \left\lceil \frac{DT_i - \Delta t^{adv}}{\widetilde{t}} \right\rceil \quad (26)$$

The parameter $\widetilde{UT}_{i,t}$ is not defined for $t < \widetilde{UT}_i^{agg}$, $\widetilde{UT}_{i,t} = \widetilde{UT}_i^{agg}$ for $t = \widetilde{UT}_i^{agg}$, while the following expression holds for $t > \widetilde{UT}_i^{agg}$:

$$\widetilde{UT}_{i,t} = max\left(0, UT_i - max\left(0, min\left(t - \Delta t^{adv}, \left\lceil \frac{UT_i}{\widetilde{t}} \right\rceil\right)\right)\right) \cdot \widetilde{t} + max\left(0, min\left(t - \Delta t^{adv}, \left\lceil \frac{UT_i}{\widetilde{t}} \right\rceil\right)\right) \quad (27)$$

The same applies for $\widetilde{DT}_{i,t}$: it is not defined for $t < \widetilde{DT}_i^{agg}$, $\widetilde{DT}_{i,t} = \widetilde{DT}_i^{agg}$ for $t = \widetilde{DT}_i^{agg}$, and the following expression holds for $t > \widetilde{DT}_i^{agg}$:

$$\widetilde{DT}_{i,t} = max\left(0, DT_i - max\left(0, min\left(t - \Delta t^{adv}, \left\lceil \frac{DT_i}{\widetilde{t}} \right\rceil\right)\right)\right) \cdot \widetilde{t} + max\left(0, min\left(t - \Delta t^{adv}, \left\lceil \frac{DT_i}{\widetilde{t}} \right\rceil\right)\right) \quad (28)$$

The parameters $\widetilde{UT}_{i,t}$ and $\widetilde{DT}_{i,t}$ are defined for only $t \geq \widetilde{UT}_i^{agg}$ and $t \geq \widetilde{DT}_i^{agg}$ respectively.

4. Case studies

The different solution approaches presented in this work were tested and compared on five test cases. These are listed in Table 2 where the number on buses, generation units and branches are shown. Table 3 presents the size of the MILP models in terms of number of variables and constraints when SSNCs are not considered. This is the size of the problems at the beginning of each solution method, since SSNCs are added during the execution of the algorithms. Please note that without the SSNCs the problems are small (featuring very low integrality gap) and become even smaller thanks to the solver's pre-solve action. However, Table 3 also shows that the actual number of SSNCs present in the original formulation is huge (we remind that the pre-contingency flow constraints are equal to $t*L$ and the post-contingency ones are equal to $t*L^2$ – with t the number of timesteps and L the number of network branches). Therefore, despite the small size and tightness of the models without SSNCs, the addition of these makes the problem very large in size, dramatically increasing the memory requirements and making the problem extremely hard to solve if not intractable. For this reason special algorithmic approaches are developed to reduce the computational time. This is needed especially for those cases when multiple, intra-day runs are executed during the day (e.g., in case multiple market sessions are present or the weather/price/demand forecast is frequently updated).

The test cases consider random parameter profiles (e.g., demand at each node) generated from publicly available data. In addition, with the aim of simulating an intraday scheduling optimization, the 24-hour profiles may start from any hour of the day (e.g., from 4:00 pm to 3:00 pm of the day after). Each test case was generated according to the methodology proposed in [34], that is based on the information provided by PGLIB [35] for the grid definition, and by PJM [36] for the bids, load data at each node and generator units parameters. Parameters include variable O&M and fuel costs, start-up and shutdown costs, minimum up- and down-time (hours), upwards and downwards ramping limits, minimum and maximum allowed load from year 2019.

The generated test cases (and the corresponding MILP models) have a time discretization of one hour. Therefore, each test instance is characterized by a 24-timestep time horizon.

All tests were run on a workstation featuring a six-core Intel® i7 processor with 16 GB of RAM. The solver used was Gurobi 9.5 [37], with default settings except for the MIP gap that was set to 1 % and the Time Limit to 8 h for all cases except for "case6468rte". This one, being the largest, was set to 30 h. Regarding the LP relaxation, the choice between Simplex and Barrier was made automatically by the solver. However, for the test cases considered, Simplex was chosen just for case ieee118. Finally, the programming language used was Julia, with the package JuMP for the creation of the optimization model.

5. Results

In this section the results obtained on the selected test cases are shown for the four methodologies presented in this work: the reference iterative approach based on constraint violation (M1, section 3.2.1), the enhanced iterative approach based on M1 with a "Quadratic Programming" filter (M2, section 3.2.2), a call-back based method replicating the reference method M1 (M3, section 3.2.3), and a Shrinking Horizon algorithm with solver callback functions (M4, section 3.2.4). The computational time, together with the operational costs, cost of electricity, and percentage of served demand are shown with the aim of providing a comprehensive and exhaustive view.

Results are divided into two subsections: the first one showing the optimization outcomes when the objective function $OF1$ (Eq. (1)) is considered in the model; the second related to $OF2$ (Eq. (2)). In this way

Table 2
Case studies considered in this study and their most representative features.

Name	First hour of the horizon	Generators	Buses	Lines	Transformers	System demand • Max/Min [GW] • Average [GW] • Total (horizon) [GWh]
ieee118	13	17	118	177	9	• 4.1/2.4 • 3.3 • 79.1
wp2383	13	293	2383	2726	170	• 14.0/11.98 • 12.9 • 309.1
sp3120	18	452	3120	3487	206	• 15.0/10.77 • 13.46 • 323.1
pegase2869	24	487	2869	4086	496	• 96.22/74.96 • 86.94 • 2086.48
case6468rte	20	1295	6468	7681	1319	• 79.59/60.70 • 71.93 • 1726.41

Table 3
Main mathematical features of the MILP model for the different case studies.

Name	Problem dimension (no pre-solve, OF1 and OF2) • Variables (Continuous/Binary) • Constraints (non-SSNC)	Problem dimension (after pre-solve) • Variables (Continuous/Binary) • Constraints (non-SSNC)		SSNC constraints in full model	Integrality Gap for model without SSNCs	
		OF1	OF2		OF1	OF2
ieee118	• 3697/1224 • 2928	• 1248/1010 • 1,790	• 1248/1010 • 1,138	1,669,536	2.76 %	0.03 %
wp2383	• 64,801/21,096 • 49,296	• 7872/19426 • 24,030	• 7872/19426 • 24,030	402,706,176	0.20 %	0.07 %
sp3120	• 87,289/32,544 • 76,008	• 14,640/30,214 • 38,989	• 14,640/30,214 • 38,989	654,813,216	0.13 %	0.01 %
pegase2869	• 66,481/35,064 • 81,888	• 26,160/29,555 • 45,130	• 26,160/29,555 • 45,130	1,007,966,688	0.11 %	0.01 %
case6468rte	• 164,665/85,032 • 198,480	• 39,408/77,826 • 101,212	• 39,408/77,826 • 101,212	3,888,432,000	0.16 %	0.04 %

it is possible to assess the effects on the solution and computational requirements (e.g., time) related to considering the maximization of social welfare as objective function over the typical total operational costs. Please note that only the operational costs are presented so to have a fair comparison between the operational decisions. In this way the high cost of the shed demand in *OF*1 and the revenues related to meeting the system demand in *OF*2 are filtered out. Comparing the plain results from considering the two objective functions would have been meaningless (i.e., given the same solution, the objective functions would be with opposite signs in most cases).

5.1. Model with objective function*OF*1

By considering Eq. (1) as the objective function, the model is the typical one found in SCUC problems where the energy balance constraint is relaxed by the addition of a shed demand penalty variable $d_{b,t}^{shed}$. The penalty cost c^{shed} was set to $\$10^7$ to minimize the shed amount.

Starting from Fig. 4, the total solution times for each approach are shown. Considering the two iterative approaches M1 and M2, a negligible solution time difference is seen for the smallest test case "ieee118", while for the other ones M2 is significantly faster (between −58 % and −74 % reduction). The approach M1 was unable to find a solution for the case "sp3120" and the method timed out. When looking at the approach M3 (tackling the full model with callback function), the time required to get to the optimal solution is significantly decreased, especially for cases "sp3120" and "wp2383" (–33 % and −66 % with respect to M2), while a modest improvement was seen for "pegase2869". Since M3 is in fact a different implementation of M1 (it follows the same algorithmic steps regarding the discovery and addition of new SSNCs), when compared to it the reduction of time obtained is very significant. This shows how different paths can be taken when developing algorithms, with significantly different computational outcomes. By "converting" M1 into a callback function, it is shown the potential of embedding some algorithmic procedures in the solver's Branch-and-Cut algorithm. Finally, when a solution approach based on the Shrinking Horizon is considered (method M4), a further reduction in computational time is achieved. This is very evident in the cases "sp3120" and "pegase2869", where M4 was 15.5 and 55.4 times faster than M3, respectively. This reduction in time is related to the significantly smaller size of the subproblems tackled by M4, together to the efficient use of callbacks for SSNCs discovery.

It is important to point out that for the case "case6468rte" each solution method timed out (solver Time Out parameter for this case was set to 30 h of CPU time). In fact, M1, M2 and M3 were not able to find a solution that did not violate any SSNCs within the given time, while M4 was not able to find a solution covering the entire horizon (time out

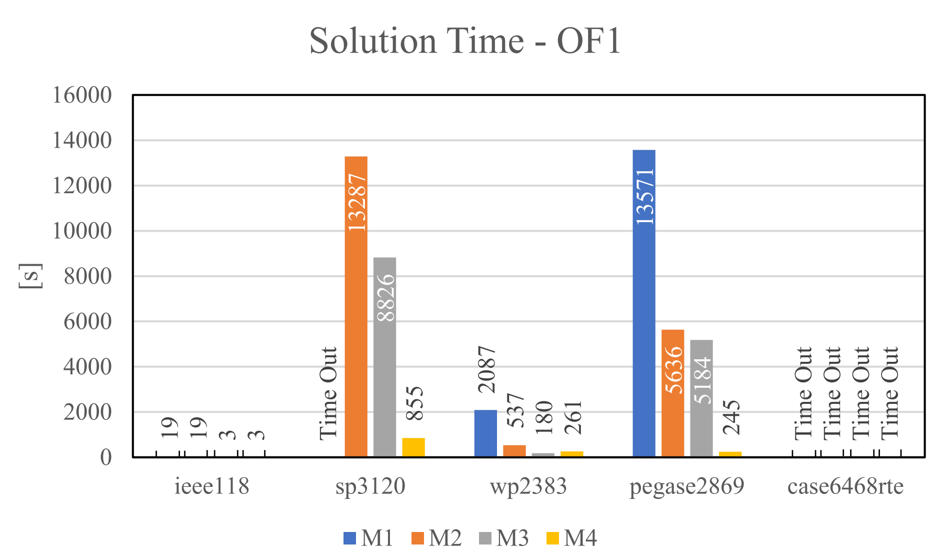

Fig. 4. Solution time for the different method tested when model considers *OF*1 as objective function.

Table 4
Number of iterations, total time spent in evaluating the SSNCs and how many of them are added for each case using method M1 and M2 and considering objective function OF1 and OF2.

	Iterations			
	M1 (OF1)	M2 (OF1)	M1 (OF2)	M2 (OF2)
ieee118	16	6	14	4
sp3120	Time out	31	42	31
wp2383	11	8	11	7
pegase2869	45	11	46	11
case6468rte	Time out	Time out	48	10
	Time spent in SSNCs evaluation [s]			
	M1 (OF1)	M2 (OF1)	M1 (OF2)	M2 (OF2)
ieee118	0.2	8.8	0.1	6.4
sp3120	Time out	1205.3	23.5	568.9
wp2383	4.0	60.2	4.3	46.6
pegase2869	17.3	311.9	16.1	160.7
case6468rte	Time out	Time out	72.9	1021.7
	Total SSNCs added			
	M1 (OF1)	M2 (OF1)	M1 (OF2)	M2 (OF2)
ieee118	360	897	311	724
sp3120	Time out	2713	1014	2430
wp2383	244	630	232	532
pegase2869	1099	2001	1085	1686
case6468rte	Time out	Time out	1152	1612

before finishing all iterations). Similar comments can be made for case “sp3120” since M1 did not find a solution within the maximum allowed time.

An analysis on the impact of considering the QP subproblem in M2 with respect to the more traditional M1 approach can be seen in Table 4. This table shows the number of iterations needed to obtain the optimal solution, as well as the total computational time spent for evaluating the SSNCs and how many of them are added. By looking at the numbers, it can be said that M2 requires fewer iterations, it spends more time in evaluating the SSNCs and it adds more of them in the model. The reason why M2 spends more time in evaluating SSNCs is because solving a QP subproblem for each timestep of the model takes significantly longer than just checking violated constraints and adding them to the set of active ones. At each iteration, M2 evaluates the security and network constraints violated by the current solution and the ones that are violated by the solutions of the QP subproblems. As a consequence, the number of added constraints per iteration is greater. This, however, further reduces the feasible region of the problem at each iteration and the SSNCs added by the QP subproblems prove to predict well those ones that would be added in the following iterations. This results in fewer iterations and an overall shorter solution time.

Finally, it is important to note that the number of SSNCs added with M2 is almost 3 times greater than with M1 (due to the QP filter), meaning that many of the added constraints are not active when the optimal solution if found. This offers some important insights about why special algorithmic methods were developed instead of solving the full model directly. In fact, the total number of constraints added to the model is dramatically lower than one of the original problem (see Table 3). This is evident already by looking at the total number network constraints violated by the solution of the model without SSNCs (from which just a fraction are added to model – Table 6). Therefore, to limit memory consumption and improve the computational time, a clever solution is to check the active SSNCs as new solutions are found. It is important to note that when the final solution is found, just a subset of the constraint shown in Table 4 are active since some of them are linked to solutions of previous iterations. Finally, the reason why it was not possible to include in Table 4 the number of SSNCs added and the time

Table 6
Number of constraints violated by the solution of the model without SSNCs.

	Number of violated SSNCs
ieee118	379
sp3120	395
wp2383	1563
pegase2869	2118
case6468rte	1909

Table 5
Operational results for each test case and solution method considered.

	Operational cost [k$]							
	OF1				**OF2**			
	M1	M2	M3	M4	M1	M2	M3	M4
ieee118	2,486	2,406	2,413	2,423	2,243	2,223	2,255	2,230
sp3120	Time out	6,736	5,981	6,074	5,454	5,735	5,876	5,464
wp2383	3,332	3,361	3,415	3,344	3,386	3,800	3,395	3,324
pegase2869	21,957	21,999	21,914	37,041	18,279	18,346	18,514	20,186
case6468rte	Time out	Time out	Time out	Time out	11,616	11,763	12,717	11,654
	Demand served							
	OF1				**OF2**			
	M1	M2	M3	M4	M1	M2	M3	M4
ieee118	100 %	100 %	100 %	100 %	99.96 %	99.95 %	99.98 %	99.97 %
sp3120	Time out	100 %	100 %	100 %	99.94 %	99.95 %	99.97 %	99.08 %
wp2383	100 %	100 %	100 %	100 %	99.98 %	99.94 %	100 %	99.99 %
pegase2869	100 %	100 %	100 %	99.60 %	99.98 %	99.98 %	100 %	99.58 %
case6468rte	Time out	Time out	Time out	Time out	99.67 %	99.09 %	99.62 %	99.65 %
	Cost of Electricity [$/MWh]							
	OF1				**OF2**			
	M1	M2	M3	M4	M1	M2	M3	M4
ieee118	31.25	30.99	31.08	31.21	28.89	28.64	29.05	28.73
sp3120	Time out	18.75	18.58	18.87	16.95	17.83	18.26	17.13
wp2383	10.78	10.87	10.98	10.82	10.96	12.30	10.98	10.76
pegase2869	10.51	10.53	10.50	17.81	8.76	8.79	8.87	9.70
case6468rte	Time out	Time out	Time out	Time out	6.75	6.88	7.39	6.78

spent in doing so for M3 and M4 is because the Gurobi callback functions implemented in Julia/JuMP did not allow to retrieve and save this information.

The operational results are presented in Table 5, where the actual total operational cost over the considered horizon and the percentage of the total served demand are shown. At first it can be noticed how all cases that did not time out meet all the system demand. Just for one case this did not happen: the test case "pegase2869". Here the solution obtained with the method M4 showed a 0.4 % shed demand, corresponding to 6.47 GWh. The reason why this happens can be understood by looking at how the method M4 works and Fig. 5. At the first iteration just the first four timesteps of the horizon retain the original parameters (discrete timesteps) while the other ones are aggregated. The optimal solution is the one that avoids any shed demand at the minimum operational cost both in the discrete and aggregate timesteps. Once the optimal solution for the first iteration is evaluated and the first four timesteps are saved, they are used to define the starting conditions for the subsequent shrunk horizon. However, these boundary conditions strongly reduce the generator's decision space, potentially making them unable to operate to serve the demand when the hourly (discrete) values are unveiled (due to the time-linking constraints related to ramping limits and minimum up/down time). This is what actually happens in case "pegase2869", where the optimal decisions of Iteration 1 force the system to shed demand in the first four hours of Iteration 2 and 3.

Table 5 also shows the total operational cost of the different solutions found. By looking at the solutions obtained with the methods M1, M2 and M3 (which are exact methods), the difference is motivated by the chosen 1 % MIP gap. When M4 is considered, instead, the solution achieved for cases "ieee118", "sp3120" and "wp2383" resulted optimal, while a suboptimal solution is obtained for the other cases. In particular, the solution of "pegase2869" features a total operational cost + 69 % higher than the optimal one. Thus, not only the solution is not able to fully meet the demand, but its cost is also significantly higher. The reason why this happens is related to what was previously mentioned: as the horizon is shrunk and new, more precise discrete parameters are unveiled, the solution tries to adapt to minimize the shed demand, becoming suboptimal because of the previous decisions. Therefore, the commitments of Iteration 1 propagate to the next ones, shifting the decision process from finding the minimum operational cost to finding the solution that minimizes the shed demand.

Despite having no guarantees on optimality, M4 showed to find good quality solutions that are compliant with SSNCs. If optimality is a necessity, M4 could be used for warm-starting the SCUC solution (both in terms of initial active constraints and variable values).

5.2. Model with objective function*OF*2

When Eq. (2) is considered as the objective function, the solution will maximize the portion of the expected demand that maximizes the profit. This depends on the electricity price $c^d_{b,t}$, which must be set to a value higher than the generation cost. In this study $c^d_{b,t}$ is time dependent and equal to the sum of the median and the 99th percentile of the system generation cost. Then, per each timestep the same value of $c^d_{b,t}$ is shared across all buses. In this way the electricity prices are set to reasonable values simulating what happens in different regions of the world.

Different results are obtained with respect to considering *OF*1. At first, the total run time for the different test cases can be seen in Fig. 6. Here it is evident how the computational time is greatly reduced with respect to the model featuring *OF*1. The overall reduction is between −11 % and −93 % when the same cases and methods are compared, and up to −99.2 % when the fastest method with *OF*2 is compared with the slowest with *OF*1. On the other hand, cases "ieee118" and "sp3120" showed an increase of + 8.6 % and + 14.8 % with M3, respectively.

The reason behind the significant reduction in computational time across all cases is a complex task to perform. However, by looking at the model with *OF*2, some things can be noted. The first one is that the shed demand variable $d^{shed}_{b,t}$ is used for the evaluation of the actual served demand $\left(D_{b,t} - d^{shed}_{b,t}\right)$ at bus b and time t. In particular, $d^{shed}_{b,t}$ is not penalized and instead $\left(D_{b,t} - d^{shed}_{b,t}\right)$ is incentivized by the price $c^d_{b,t}$. Then, while the balance equation is a hard constraint to meet when *OF*1 is used (unless a high penalty is paid because of $d^{shed}_{b,t}$ – see Eq. (A.32) in Appendix), here the equality implicitly becomes an inequality since $d^{shed}_{b,t} \leq D_{b,t}$. This makes the LP relaxation of the model with *OF*2 stronger, as it can be seen already by looking at the integrality gap when no SSNCs are considered (Table 3). The main consequence of this is that a less expensive search in the solution tree needs to be done by the solver, reducing the overall solution time (similarly to what noted in [38]).

While in the *OF*1 cases a constant decrease in computational time was shown going from method M1 to M4, with *OF*2 this trend is not as evident. Considering the iterative methods, M2 does not guarantee a clear computational advantage in all cases. Despite being significantly faster for case "pegase2869" (−60 %) and "case6468rte" (−93 %), similar solution times are seen in cases "ieee118" and "sp3120". Regarding the case "ieee118", being a small instance, the additional

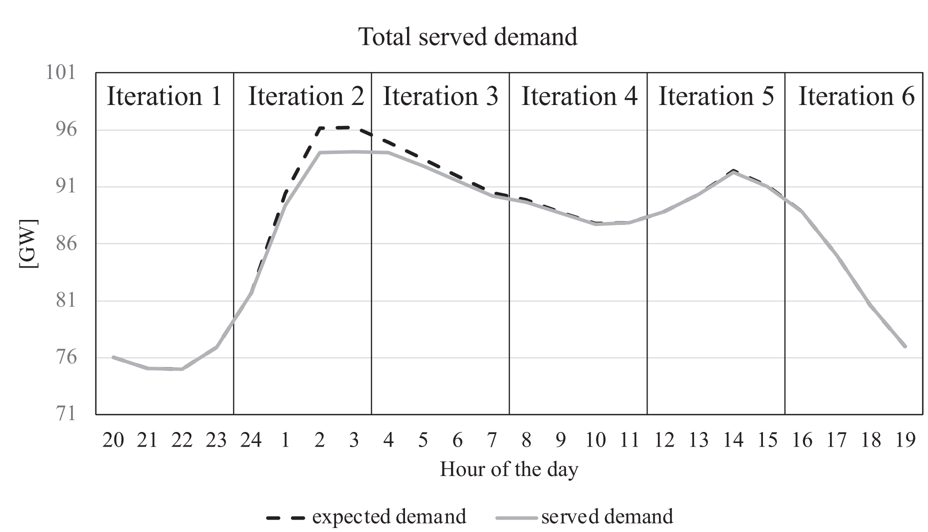

Fig. 5. Solution obtained with the shrinking horizon approach M4 for pegase2869 at each iteration. Only the solution related to the first four discrete timestep is saved at each subsequent run.

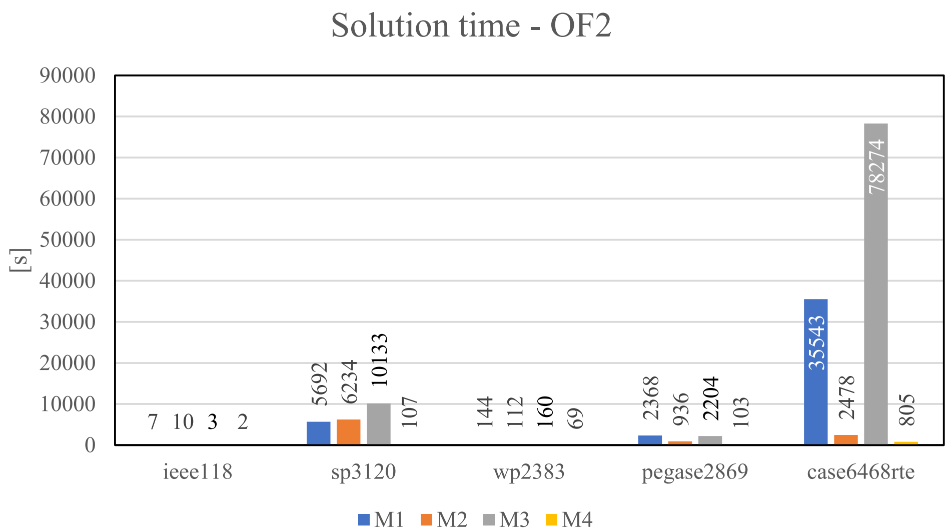

Fig. 6. Solution time for the different method tested when model considers *OF*2 as objective function.

time spent in evaluating the SSNCs does not translate in an overall solution time reduction. For what concerns the case "sp3120", the absence of run time reduction could be related to the solver logics, which are very hard to investigate.

Method M3 based on callback functions did not show any significant improvement over M2, resulting sometimes to be much slower. This is evident in cases "sp3120", "pegase2869" and "case6468rte". Finally, as for the previous cases where *OF*1 was considered, M4 showed faster solution times across all test cases, with significant reductions.

When looking at Table 4 a comparison between the results obtained with M1 and M2 is shown. Here similar comments to the *OF*1 can be made: M2 requires fewer iterations, it spends more time in evaluating SSNCs, and adds a greater number of them to the model. In general, by adopting *OF*2, a very similar number of iterations is required to obtain the optimal solution. On the other hand, fewer SSNCs are added in both M1 and M2, despite the numbers remaining similar. The lower number of SSNCs also results into a reduction of the time spent in evaluating them.

Finally, the operational results are presented in Table 5. The first thing that can be noted is how in almost all cases and methods the served demand is below 100 %. However, the largest percentage of shed demand is 0.92 % ("case6468rt3", M2), with an average value equal to 0.19 %. Therefore, the amount of shed demand is very small. When the total operation costs are considered, they result to be lower than most cases evaluated with *OF*1, with a maximum reduction recorded of −45.5 % ("pegase2869", M4), and an average reduction across all cases of −11.8 %. The reduction in operational cost also reflects in lower Costs of Electricity across most cases.

These results suggest that by changing the objective function of the model, better solutions can be found in a shorter time. The reason why different solutions are found can be understood by analyzing *OF*1 and *OF*2. At first, given $c^{d}_{b,t}$ large enough and a MIP gap set to 0 %, solving the model with either *OF*1 or *OF*2 should bring the same solution, or at least solutions with the same operational cost. However, this does not occur due to three reasons. At first, by solving the problem with *OF*2, just the fraction of the demand that is profitable the most is met (while assessing that SSNCs are met). As a consequence, it might happen that higher profits are achieved by serving less demand than with *OF*1. Then, given the high value of c^{shed} (set to 10^7) in *OF*1, numerical issues can occur, and thus affecting the final solution (e.g., a total shed demand of 10^{-5} MWh across the whole horizon has a cost of 100 k$). However, it is important to note that during the test phase, lower values of c^{shed} were considered and slower solution time were recorded (motivating therefore the choice). Finally, having set a MIP gap to 1 % might results in finding different solutions as well. This is more evident when instead of analyzing the operational costs of the solutions, the expected profit is considered (which is what *OF*2 evaluates). By looking at Table 7, it can be noted how the difference in profits between *OF*1 and *OF*2 are very limited. The largest differences are recorded when using method M4, which is expected since suboptimal decisions at the first iteration propagates across the other ones.

In conclusion, considering *OF*2 resulted in a significant improvement in computational time, while showing very limited shed demand. Even for those cases where achieving zero shed demand is mandatory, the solution obtained with *OF*2 can still be used as a warm start or for using heuristics. This could be particularly beneficial for cases where reducing the solution time is more important than guaranteeing optimality.

6. Conclusions

This work presents three novel solution approaches for the SCUC problem. The solution approaches were tested over 5 test cases with two

Table 7
Expected profits evaluated per each test case.

	Profits [k$]							
	OF1				OF2			
	M1	M2	M3	M4	M1	M2	M3	M4
ieee118	300,271	300,192	300,265	300,230	300,312 (+0.01 %)	300,304 (+0.04 %)	300,360 (+0.03 %)	300,363 (+0.04 %)
sp3120	Time out	857,294	857,574	857,452	857,418 (n.c.*)	857,616 (+0.04 %)	857,456 (−0.01 %)	850,158 (−0.85 %)
wp2383	1,821,952	1,821,981	1,821,898	1,821,945	1,820,466 (−0.08 %)	1,821,505 (−0.03 %)	1,821,916 (+0.00 %)	1,821,742 (−0.01 %)
pegase2869	10,705,909	10,705,867	10,705,952	10,647,914	10,707,099 (+0.01 %)	10,707,125 (+0.01 %)	10,706,864 (+0.01 %)	10,660,145 (+0.11 %)
case6468rte	Time out	Time out	Time out	Time out	8,833,581 (n.c.*)	8,781,962 (n.c.*)	8,828,043 (n.c.*)	8,831,768 (n.c.*)

* : non computable since timed out and no feasible solution was found.

different objective functions and compared with a well-known iterative approach in literature.

When total operational cost ($OF1$) is considered as objective function, the comparison between the methods M1-M4 shows how the fastest is the one based on the Shrinking Horizon approach (M4), followed by M3 and M2. For the test cases considered, a reduction in computational time with respect to M1 of up to −74.2 %, −91.36 % and −98.19 % was recorded for M2, M3 and M4 respectively. Regarding the solutions found, all methods presented similar solution with the only exception of M4 for case "pegase2869" (here 0.4 % shed demand was present).

When considering $OF2$ as objective function of the model (that includes virtual revenues for serving the demand) the first thing that can be noted is how the computational time is significantly reduced compared with $OF1$. By comparing each solution approach, the reduction in time is up to −93 %, −83.4 %, −57.5 % and −87.44 % for M1, M2, M3 and M4 respectively. In this case too M4 is overall the fastest. However, it is important to point out that the solutions obtained with $OF2$ are slightly different from the ones found with $OF1$, since they aim at maximizing the overall profit (up to + 0.11 % higher than $OF1$) at the cost of shedding part of the demand (up to 1 %).

In general, by looking at the results for $OF1$ and $OF2$ it is not possible to provide a single answer on which method is the best. Computational results show how M4 is much faster in finding good quality solutions. However, this method does not guarantee optimality. On the other hand, among the methods that do guarantee optimality, M3 is the fastest for $OF1$ but not for $OF2$. Thus, this work provides with new approaches to enrich the choice of available algorithms for the solution of the SCUC problem.

Future work should focus on improving the computational efficiency of callback functions on different solvers, finding better approaches to predict SSNCs, evaluate the impact of using the solution found with $OF2$ as a warm-start of the problem with $OF1$, and test the presented methods to solve Robust and Stochastic version of SCUC.

CRediT authorship contribution statement

Alessandro Francesco Castelli: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation. **Iiro Harjunkoski:** Writing – review & editing, Validation, Resources, Methodology, Investigation, Formal analysis. **Jan Poland:** Validation, Software, Methodology, Data curation. **Marco Giuntoli:** Validation, Software, Methodology, Formal analysis, Data curation. **Emanuele Martelli:** Supervision, Resources, Methodology, Formal analysis, Writing – review & editing. **Ignacio E. Grossmann:** Writing – review & editing, Supervision, Methodology, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix

A.1. Nomenclature

Abbreviations	
ISF	Injection Shifting Factor
LODF	Line Outage Distribution Factor
LSF	Linear Sensitivity Factor
PTDF	Power Transfer Distribution Factor
SSNC	System and Security Network Constraint
Sets	
B	Set of the buses in the network
G	Set of the available generators
G^{on}	$G^{on} \subseteq G$, set of the must-run generators
G^{off}	$G^{off} \subseteq G$, set of generators currently not available
G_r^R	$G_r^R \subseteq G$, set of the generators belonging to reserve area r
G_b^B	$G_b^B \subseteq G$, set of the generators connected directly to bus b
L	Set of the network branches
L_b^{in}, L_b^{out}	$L_b^B \subseteq L$, set of the transmission lines connected to bus b (in- and out-going)
L^{off}	$L^{off} \subseteq L$, set of the transmission lines currently not available
R	Network reserve areas
T	Set of the timesteps within the horizon considered
Binary Variables	
$\delta_{i,t}^{on}, \delta_{i,t}^{off}$	Start-up/shut-down flagger for unit i at time t. 1 if unit i started-up/shut-down at t, 0 otherwise
$z_{i,t}$	On/off status of unit i at time t. 1 if unit i is on at time t, 0 otherwise
Continuous Variables	
$d_{b,t}^{shed}$	Demand shed at bus b at time t [MWh]
$f_{c,t}$	Pre-outage power flowing through line c at time t [MW]
$f_{c,t}^{d}$	Post-outage power flowing through line c at time t resulting from outage of line d [MW]
$p_{i,t}$	Energy generated by unit i at time t [MWh]
$r_{i,t}^{up}, r_{i,t}^{down}$	Upward and downward reserve quantities made available by unit i at time t [MWh]
Parameters	
$\#RU_i, \#RD_i$	Minimum number of timesteps of length dt required by unit i to go from minimum/maximum to maximum/minimum generation according to the ramping limit RU_i/RD_i
$\#SU_i, \#SD_i$	Minimum number of timesteps needed by unit i to ramp up from SU_i to the maximum generation / to ramp down from maximum generation to SD_i
Δt^{adv}	Time advancement of the shrinking horizon algorithm (equal to the number of discrete timesteps considered in each shrunk horizon)
$D_{b,t}$	Demand related to a timestep duration dt to meet at bus b at time t [MWh]
$D_{b,t}^{agg}$	Demand related to the aggregate timestep t (of duration $\widetilde{t}\cdot dt$) to meet at bus b [MWh]

(continued on next page)

(*continued*)

dt	Timestep duration [h]
$\overline{F}_c$	Maximum power flow allowed through line c [MW]
H_i^U, H_i^D	Number of time periods for completing the minimum up\down time at the beginning of the considered time horizon, for unit i
$ISF_{m,n}^b$	ISF element related to power flowing on branch between nodes m, n due to an injection/withdrawal at node b
$LODF_{m,n}^{j,k}$	LODF element related to the redistribution on branch between nodes m, n of the power flowing on the branche between nodes j, k as a consequence of its outage
$\underline{P}_i, \overline{P}_i$	Minimum and maximum allowed generation unit i within dt [MWh]
RU_i, RD_i	Ramp-up\down limits for unit i between timesteps of duration dt [MWh]
$Res_{r,t}^{up}, Res_{r,t}^{down}$	Upward and downward reserve requirements of area r at time t [MWh]
SU_i, SD_i	Ramping limits at start-up\shutdown for unit i for timestep duration dt [MWh]
$\widetilde{t}$	Number of original timesteps considered in the aggregated one.
UT_i, DT_i	Minimum required number of timesteps of duration dt that unit i is required to stay on/off before being shutdown/started-up
$\widetilde{UT}_{i,t}, \widetilde{DT}_{i,t},$	Minimum required number of timesteps (aggregate and/or discrete) that unit i is required to stay on/off in the timesteps before t (t included), if on/off at t
$\widetilde{UT}_i^{agg}, \widetilde{DT}_i^{agg}$	Minimum required number of timesteps (aggregate and/or discrete) that unit i is required to stay on/off starting up/shutting down at the beginning of the horizon
y_c	$y : c \rightarrow (m, n)$, map defining the link between line c and the connected buses m and n

A.2. Mixed integer linear programming model

- Generation units' operational constraints

The operational constraints related to each unit available in the considered grid are described in this section. At first, each unit is characterized by its maximum and minimum generation limits:

$$\underline{P}_i z_{i,t} \leq p_{i,t} \leq \overline{P}_i z_{i,t}, \forall i \in G, \forall t \in T \tag{A.1}$$

where $\underline{P}_i$ and $\overline{P}_i$ are the minimum and maximum allowed generation levels of unit i respectively, $p_{i,t}$ the energy (MWh) generated by unit i at time t, and $z_{i,t}$ the on–off status of unit i at time t.

The start-up and shutdown flags $\delta_{u,t}^{on}$ and $\delta_{u,t}^{off}$ are introduced to define when unit i is switched on or off. The logical relationship between the on–off status and the already mentioned flags is defined by the following constraint for all timesteps except the first one:

$$z_{i,t-1} - z_{i,t} + \delta_{i,t}^{on} - \delta_{i,t}^{off} = 0, \forall i \in G \backslash G^{on}, \forall t = 2, \cdots, |T| \tag{A.2}$$

Only for the first timestep of the horizon, the relationship takes the following form:

$$z_i^{ini} - z_{i,1} + \delta_{i,1}^{on} - \delta_{i,1}^{off} = 0, \forall i \in G \backslash G^{on} \tag{A.3}$$

with z_i^{ini} a parameter being either equal to 0 or 1, defining the inherited initial on–off status of the considered unit.

Other time-linking constraints are those describing the minimum up- and down-times for each unit. These are taken from Morales-España et al. [39] since they correspond to the tightest formulation found in the literature (as also stated in Knueven at al. [40]). The minimum up-time is enforced by the following expression:

$$\sum\nolimits_{\tau=t-UT_i+1}^{t} \delta_{i,\tau}^{on} \leq z_{i,t}, \forall i \in G \backslash G^{on}, \forall t \in [UT_i, \cdots, |T|] \tag{A.4}$$

while the minimum down-time constraints are the following:

$$\sum\nolimits_{\tau=t-DT_i+1}^{t} \delta_{i,\tau}^{off} \leq 1 - z_{i,t}, \forall i \in G \backslash G^{on}, \forall t \in [DT_i, \cdots, |T|] \tag{A.5}$$

Given the initial on–off status of the unit z_i^{ini} (a parameter either equal to 0 or 1), the following inequalities are added to ensure that the unit completes the minimum up- or down-time in the current planning horizon if switched on or off in the previous one:

$$z_i^{ini} \leq z_{i,t}, \forall i \in G \backslash G^{on}, t = 1, \cdots, min(|T|, H_i^U) \tag{A.6}$$

$$z_i^{ini} \geq z_{i,t}, \forall i \in G \backslash G^{on}, t = 1, \cdots, min(|T|, H_i^D) \tag{A.7}$$

with H_i^U and H_i^D being the number of timesteps required to complete the minimum up- and down-time in the current horizon, respectively.

Finally, upward and downward ramping constraints are considered in the model:

$$p_{i,t} - p_{i,t-1} \leq RU_i z_{i,t-1} + SU_i \delta_{i,t}^{on}, \forall i \in G, \forall t = 2, \cdots, |T| \tag{A.8}$$

$$p_{i,t-1} - p_{i,t} \leq RD_i z_{i,t} + SD_i \delta_{i,t}^{off}, \forall i \in G, \forall t = 2, \cdots, |T| \tag{A.9}$$

where RU_i and RD_i are the unit's ramp-up and −down limits during nominal operation, SU_i and SD_i the ramping limits during start-up and shut-down. If the unit is already in operation at the beginning of the considered timestep (namely $z_i^{ini} = 1$), then for $t = 1$ the constraints take the following form:

$$p_{i,1} - p_i^{ini} \leq RU_i z_i^{ini} + SU_i \delta_{i,1}^{on}, \forall i \in G \tag{A.10}$$

$$p_i^{ini} - p_{i,1} \leq RD_i z_{i,1} + SD_i \delta_{i,1}^{off}, \forall i \in G \tag{A.11}$$

Please note that for those units that must run for all the time steps of the considered horizon (hence the ones belonging to the set G^{on}), Eqs. (A.8)-

(A.11) do not consider the terms $SU_i\delta^{on}_{i,t}$ and $SD_i\delta^{off}_{i,t}$. In addition, the following constraints hold for ensuring that must-run units are in operation for the entire considered horizon, and the unavailable units are kept shut down at each time period:

$$z_{i,t} = 1 \forall i \in G^{on}, \forall t \in T \tag{A.12}$$

$$z_{i,t} = 0 \forall i \in G^{off}, \forall t \in T \tag{A.13}$$

- Reserve constraints

Reserve constraints are needed to ensure that whichever solution is evaluated, limited upwards and downwards fluctuation in the load demand at each node can be withstood. As a consequence, a sufficient number of generation units must make available part of their generation capacity for reserve purposes. This is enforced by defining variables related to the amount of generation power allocated as upward and downward reserve ($r^{up}_{i,t}$, $r^{down}_{i,t}$) for each generation unit. At first, the generation reserve of each unit is limited by its maximum generation capacity and ramping capabilities:

$$0 \le r^{up}_{i,t} \le \min(\overline{P}_i - \underline{P}_i, RU_i)\cdot\left(1 - \delta^{on}_{i,t}\right) + \left(SU_i - \underline{P}_i\right)\delta^{on}_{i,t}, \forall i \in G, \forall t \in T \tag{A.14}$$

$$0 \le r^{down}_{i,t} \le \min(\overline{P}_i - \underline{P}_i, RD_i)\cdot\left(1 - \delta^{off}_{i,t+1}\right) + \left(SD_i - \underline{P}_i\right)\delta^{off}_{i,t+1}, \forall i \in G, \forall t \in T \tag{A.15}$$

Then, at each timestep the maximum reserve that can be provided is limited by the quantities committed to serve the demand:

$$r^{up}_{i,t} \le z_{i,t}\overline{P}_i - p_{i,t}, \forall i \in G, \forall t \in T \tag{A.16}$$

$$r^{down}_{i,t} \le p_{i,t} - z_{i,t}\underline{P}_i, \forall i \in G, \forall t \in T \tag{A.17}$$

Although Eqs. (A.14) and (A.15) consider the ramping capabilities of the unit, they do not capture the relationship between $r^{up}_{i,t}$, $r^{down}_{i,t}$ and the status of the unit among contiguous timesteps. For this reason, the following constraints are added:

$$r^{up}_{i,t} \le RU_i z_{i,t-1} - \left(p_{i,t} - \left(p_{i,t-1} - r^{down}_{i,t-1}\right)\right) + SU_i\delta^{on}_{i,t}, \forall i \in G, \forall t = 2, \cdots, |T| \tag{A.18}$$

$$r^{down}_{i,t} \le RD_i z_{i,t} - \left(p_{i,t-1} + r^{up}_{i,t-1} - p_{i,t}\right) + SD_i\delta^{off}_{i,t} + \left(SD_i - \underline{P}_i\right)\delta^{off}_{i,t+1}, \forall i \in G, \forall t = 2, \cdots, |T| \tag{A.19}$$

Starting from Eq. (A.18), it defines the upper bound of $r^{up}_{i,t}$ depending on the operational condition. If the unit starts up at time t, then the upwards reserve is bounded by the ramping limit during startup SU_i (namely the available generation given by $SU_i\delta^{on}_{i,t} - p_{i,t}$. On the other hand, if the unit was already running at $t-1$, then the maximum quantity that can be offered as reserve is $RU_i z_{i,t-1} - \left(p_{i,t} - \left(p_{i,t-1} - r^{down}_{i,t-1}\right)\right)$. The term $p_{i,t-1} - r^{down}_{i,t-1}$ represents the power at the beginning of t (the downwards reserve is allocated to avoid any overestimate).

Regarding Eq. (A.19), the reserve $r^{down}_{i,t}$ is dependent on the generation at the previous timestep and the downward ramping limit RD_i. If the unit is operating, the upper bound is defined by the term $RD_i z_{i,t} - \left(p_{i,t-1} + r^{up}_{i,t-1} - p_{i,t}\right)$, with $p_{i,t-1} + r^{up}_{i,t-1}$ the actual generation at the end of $t-1$ in case reserve is provided. In the timestep prior shutdown ($\delta^{off}_{i,t+1} = 1$) the unit can ramp down from SD_i to $\underline{P}_i$, allowing a larger value than RD_i depending on the unit. Therefore the $(SD_i - \underline{P}_i)\delta^{off}_{i,t+1}$ is added such that the binding constraints become Eq. (A.15) and (A.17). Finally, when the unit is shutdown the only non zeros are $p_{i,t-1}$ and $SD_i\delta^{off}_{i,t}$, whose difference is always positive due to Eq. (A.9).

For the first timestep of the horizon, given the initial condition inherited from the previous horizon, the following constraints hold:

$$r^{up}_{i,1} \le p^{ini}_i - p_{i,1} + RU_i z^{ini}_i + SU_i\delta^{on}_{i,1}, \forall i \in G \tag{A.20}$$

$$r^{down}_{i,1} \le p^{ini}_i - p_{i,1} + RD_i z_{i,1} + SD_i\delta^{off}_{i,1}, \forall i \in G \tag{A.21}$$

Also here, must-run units share the same Eqs. (A.14)-(A.21), but the terms $SU_i\delta^{on}_{i,t}$ and $SU_i\delta^{off}_{i,t}$ are not present. Please note that in case the computational complexity allows to update the generators schedule frequently (e.g., every 10 min), and for applications where load fluctuation is limited, it is unlikely that most of the units are required to share their maximum upwards and downwards reserve quantities among contiguous time periods (and vice-versa). Therefore, $r^{down}_{i,t-1}$ and $r^{up}_{i,t-1}$ can be dropped from Eqs. (A.18) and (A.19), respectively (although resulting in a less conservative solution).

Finally, the following constraints are needed to ensure that the reserve requirements per each area of the network are met:

$$\sum_{i \in G_r} r^{up}_{i,t} \ge Res^{up}_{r,t}, \forall r \in R, \forall t \in T \tag{A.22}$$

$$\sum_{i \in G_r} r^{down}_{i,t} \ge Res^{down}_{r,t}, \forall r \in R, \forall t \in T \tag{A.23}$$

- System and security network constraints

Linearized system and security (N-1) constraints are considered in the MILP model in order to ensure feasible and reliable operation. These are needed to make sure that the generation and transmission schedule does not produce overload on any line both under nominal condition and in case the load is diverted because of a contingency in which at most one line fails.

Given $x_{m,n}$, $y_{m,n}$ and $z_{m,n}$ the reactance, admittance and impedance of the line connecting bus m and n, the susceptance $b_{m,n}$ can be evaluated. However, since the DC power flow model holds under the assumption of negligible line resistance with respect to its reactance [41], $z_{m,n} \cong x_{m,n}$ and therefore $b_{m,n} = -1/x_{m,n}$. The Injection Shift Factors (ISFs) can be defined as follows [42]:

$$ISF^{i}_{m,n} = \frac{x_{m,i} - x_{n,i}}{x_{m,n}},\ m,n = y(c), m,i = y(a),\ n,i = y(e),\ \forall c, a,\ e \in L, \forall i \in B \tag{A.24}$$

where $y(c), y(a)$ and $y(e)$ are the functions mapping the buses connecting lines c, a and e.

These network sensitivity parameters represent the fraction of additional 1 MW injection at bus i that flows through the line connecting bus m and n. Once defined, the Power Transfer Distribution Factors (PTDFs) and Line Outage Distribution Factors (LODFs) according to [9] can be computed for each bus and line as follows:

$$PTDF^{i,j}_{m,n} = ISF^{i}_{m,n} - ISF^{j}_{m,n}, m,n = y(c), \forall c \in L, \forall i,j \in B \tag{A.25}$$

$$LODF^{i,j}_{m,n} = \frac{PTDF^{i,j}_{m,n}}{1 - PTDF^{i,j}_{i,j}}, m,n = y(c), i,j = y(a), \forall a, c \in L \tag{A.26}$$

In particular, $PTDF^{i,j}_{m,n}$ represents the variation of the power flowing on line c (connecting m and n) resulting from the power transfer of 1 MW injected in bus i and consumed at bus j. On the other hand, $LODF^{i,j}_{m,n}$ represents the fraction of pre-contingency power flowing from bus i to j (line a) that is redistributed on the line connecting m and n (line c) consequently the outage of the line linking i and j (line a). Please note that $ISF^{i}_{m,n}$ and $LODF^{i,j}_{m,n}$ are the elements of the ISF and $LODF$ matrices, which are needed for the definition of the following network constraints.

Under nominal operating conditions (no contingencies) the power flowing through each line must always be within the rated limits. This is imposed by the following constraint:

$$-\overline{F}_c \le f_{c,t} \le \overline{F}_c, \forall c \in L\, \forall t \in T \tag{A.27}$$

where $\overline{F}_c$ is the maximum allowed power flowing through line c in each direction ($\overline{F}_c$ referred to flow limit going from bus m to n, $-\overline{F}_c$ for the one from n to m). In particular, the power flow $f_{c,t}$ is defined by the matrix ISF, the generated quantity $p_{i,t}$ and total served demand $\left(D_{b,t} - d^{shed}_{b,t}\right)$ as follows:

$$f_{c,t} = \sum_{b \in B, i \in G^B_b, G^B_b \neq \varnothing} ISF^{b}_{m,n} \cdot \frac{p_{i,t}}{dt} - \sum_{b \in B} ISF^{b}_{m,n} \cdot \left(\frac{D_{b,t} - d^{shed}_{b,t}}{dt}\right), m,n = y(c), \forall t \in T \tag{A.28}$$

with $\sum_{b \in B, i \in G^B_b, G^B_b \neq \varnothing} ISF^{b}_{m,n} \frac{p_{i,t}}{dt}$ the timestep average power flowing through line c due to the total power injected by the generators, and $\sum_{b \in B} ISF^{b}_{m,n} \left(\frac{D_{b,t} - d^{shed}_{b,t}}{dt}\right)$ the timestep average power flowing on line c due to the total power consumption. Please note that since $p_{i,t}$ and $D_{b,t} - d^{shed}_{b,t}$ are expressed as energy, $\frac{p_{i,t}}{dt}$ and $\frac{D_{b,t} - d^{shed}_{b,t}}{dt}$ represent the average generation and consumption power within a given timestep.

Overload on any line must be avoided also under any contingency in which the outage of one line happens (N-1 contingency). At time t, the post-outage power $f^{d}_{c,t}$ flowing through line c because of the outage of line d is defined as follows:

$$f^{d}_{c,t} = f_{c,t} + LODF^{j,k}_{m,n} \cdot f_{d,t}, m,n = y(c), j,k = y(d), \forall t \in T \tag{A.29}$$

where $LODF^{j,k}_{m,n} \cdot f_{d,t}$ is the fraction of pre-outage power flowing through d that is diverted onto c.

To find a solution where no post-contingency flow exceeds the line limits, the following N-1 security constraints are added:

$$-\overline{F}_c \le f^{d}_{c,t} \le \overline{F}_c, \forall c, d \in L, \forall t \in T \tag{A.30}$$

In this way the solution is robust against any line outage.

The constraints described by Eq. (A.27) and (A.30) are the System and Security Network Constraints (SSNCs) of the problem.

- Energy balance constraints

At each timestep of the considered horizon the energy demand at each node must be met. This is enforced by the following equality:

$$\sum\nolimits_{i \in G^B_b} p_{i,t} + \sum\nolimits_{c \in L^{in}_b} f_{c,t} dt - \sum\nolimits_{c \in L^{out}_b} f_{c,t} dt = D_{b,t} - d^{shed}_{b,t}, \forall b \in B, \forall t \in T \tag{A.31}$$

where $\sum_{i \in G^B_b} p_{i,t}$ is the sum of the quantities generated by the units connected to bus b, $\sum_{c \in L^{in}_b} p^{N}_{c,t}$ and $\sum_{c \in L^{out}_b} p^{N}_{c,t}$ the energy transmitted from bus b to other ones connected and vice versa, $D_{b,t}$ and $d^{shed}_{b,t}$ the requested and shed demand at bus b at time t.

For the linearized DC power flow model, the ISF matrix can be used to allocate the power flowing on each line of the network given the generation and consumption at each node. In addition, the presence of system and security network constraints (Eq. (A.27) and (A.30)) ensures that the generated power is routed on each line so as to avoid any overload under nominal and N-1 contingency conditions (therefore taking care of the balance at each node). Thus, Eq. (A.31) can be substituted by the following network balance:

$$\sum_{i\in G} p_{i,t} = \sum_{b\in B} (D_{b,t} - d^{shed}_{b,t}),\forall t \in T \tag{A.32}$$

A.3. MILP model reformulation for method M4

In this section the modelling of the operational constraints valid for the "aggregate" timesteps introduced with method M4 are presented.

- Modelling of generation bounds, energy balance, system and security network constraints, and objective function

At each timestep, the amount of electrical energy that the generators can produce is limited by the upper and lower bounds. Since the parameters $\underline{P}_i$ and $\overline{P}_i$ are related to the minimum and maximum generation allowed for the original time step duration dt (one hour in this work), Eq. (A.1) holds $\forall t \in T^{disc}$. On the other hand, when dealing with aggregated timesteps, the constraint translates to the following:

$$\underline{P}_i z_{i,t} \le p_{i,t} \le (\tilde{t}\cdot\overline{P}_i) z_{i,t},\forall i \in G \forall t \in T^{agg} \tag{A.33}$$

with $\tilde{t}$ being the number of original timesteps with length dt contained in an aggregate one. The reason $\underline{P}_i$ is the lower bound in the aggregate timesteps is because the units could be switched on only for some of the $\tilde{t}$ aggregated timesteps.

Regarding the energy balance, Eq. (A.32) holds $\forall t \in T^{disc}$. For the aggregate timesteps, the constraint becomes:

$$\sum_{i\in G} p_{i,t} = \sum_{b\in B} \left(D^{agg}_{b,t} - d^{shed}_{b,t}\right),\forall t \in T^{agg} \tag{A.34}$$

where $D^{agg}_{b,t}$ is the sum of the demand of the $\tilde{t}$ timesteps belonging to the aggregate one.

The SSNCs described by the Eq. (A.27), (A.29) and (A.30) do not change for any of the discrete and aggregate timestep. What changes is the definition of the power flowing through line c:

$$f_{c,t} = \sum_{b\in B, i\in G^B_b, G^B_b \neq \emptyset} ISF^b_{m,n}\cdot\frac{p_{i,t}}{\tilde{t}\cdot dt} - \sum_{b\in B} ISF^b_{m,n}\cdot\left(\frac{D^{agg}_{b,t} - d^{shed}_{b,t}}{\tilde{t}\cdot dt}\right), m,n = y(c),\forall t \in T^{agg} \tag{A.35}$$

In this way, for $t \in T^{agg}$ the term $f_{c,t}$ refers to the average power flow in the aggregate timestep.

The objective function keeps being either Eq. (1) and (2). The only difference is that any time-dependent cost parameters will have to be modified such that it is referred to a discrete or aggregate timestep. In particular, cost parameters related to aggregate timesteps are evaluated as the average of the elements belonging to them.

- Modelling of ramping constraints

Time linking constraints considered in the model are the ones related to the ramping limits (Eq. (A.8) and (A.9)), minimum up- and down-time (Eqs. (A.2)-(A.7)) and reserve (Eqs. (A.14)-(A.23)).

Starting from the ramping limits, they represent the maximum increase and decrease of units' generation between adjacent timesteps. By looking at Fig. 3, three different couples of adjacent timesteps can be identified: A-A, A-B and B-B. A-A refers to a couple of discrete timesteps, A-B to a couple where an aggregate timestep follows a discrete one, and B-B to a couple of aggregate timesteps. Since the aggregate and discrete timestep are characterized by different timestep duration, also the maximum allowed increase/decrease in generation among adjacent time intervals will change.

The general formulation of the constraints considered for the general couple of adjacent timesteps X and Y is the following:

$$p_{i,t} - p_{i,t-1} \le RU^{X-Y}_i z_{i,t-1} + SU^{X-Y}_i \delta^{on}_{i,t},\forall i \in G,\forall t \in T^{*}, t \in T^{**}, t-1 \in T^{***} \tag{A.36}$$

$$p_{i,t-1} - p_{i,t} \le RD^{X-Y}_i z_{i,t} + SD^{X-Y}_i \delta^{off}_{i,t},\forall i \in G,\forall t \in T^{*}, t \in T^{**}, t-1 \in T^{***} \tag{A.37}$$

The values related to the parameters $RU^{X-Y}_i, RD^{X-Y}_i, SU^{X-Y}_i, SD^{X-Y}_i$ and the sets T^{*}, T^{**} and T^{***} can be found in Table A.1 per each case.

Table A1
How parameters and sets in Eq. (A.36) and (A.37) are defined depending on the timestep couples considered in the reduced MILP with aggregate timesteps.

	A-A	A-B	B-B
RU^{X-Y}_i	RU_i	Eq. (A.42)	Eq. (A.43)
RD^{X-Y}_i	RD_i	Eq. (A.48)	Eq. (A.49)
SU^{X-Y}_i	SU_i	Eq. (A.51)	
SD^{X-Y}_i	SD_i	Eq. (A.53)	Eq. (A.54)
T^{*}	$\forall t = 2,\cdots,\lvert T^{disc}\rvert$	$\{min(t) \in T^{agg}\}$	$t : t > min(t) \in T^{agg}$
T^{**}	T^{disc}	T^{agg}	T^{agg}
T^{***}	T^{disc}	T^{disc}	T^{agg}

It is now presented how the different parameters are evaluated for each timestep couple in the model.

The operational *ramp-up* limit is computed as the difference between the maximum generation achievable in an aggregate timestep starting from the minimum allowed generation in the previous timestep. Given RU_i the maximum allowed generation increase between adjacent timesteps for unit i, the minimum number of discrete timesteps needed to augment the unit's output from minimum to maximum is defined as follows:

$$\#RU_i = \left\lfloor \frac{\overline{P}_i - \underline{P}_i}{RU_i} \right\rfloor \tag{A.38}$$

The reason why $\#RU_i$ is evaluated as the floor of $(\overline{P}_i - \underline{P}_i)/RU_i$ can be seen in Figure A.1 (dashed red line above the maximum generation line). Given, for example, $(\overline{P}_i - \underline{P}_i)/RU_i = 2.6$ and the discrete nature of the timesteps, the decimal part must be truncated. By rounding up, $\#RU_i$ would be equal to 3, therefore resulting higher than the maximum allowed. On the contrary, by rounding down, $\#RU_i$ is equal to 2 (within the bounds) and the remaining part of the ramp corresponds to a load variation of $0.6 \cdot RU_i$, which can be allocated in the following timestep.

To evaluate the upward ramping limit either between a discrete and aggregate timesteps or two aggregate timesteps, it is necessary to calculate the difference among the minimum allowed generation in the previous interval and the maximum that can be achieved. For a A-B couple, the energy generated by the unit i in the previous discrete timestep is equal to the minimum load:

$$p^A_{t-1} = \underline{P}_i \tag{A.39}$$

On the other hand, the unit's output for the previous aggregate timestep (thus related to the timestep couple B-B) is:

$$p^B_{t-1} = \tilde{t} \cdot \underline{P}_i \tag{A.40}$$

The maximum generation achievable in one aggregate timestep starting from the minimum load condition is defined by the following expression (a graphical representation of p_t can be seen in Figure A.1):

$$p^B_t = \underline{P}_i \min(\tilde{t}, \#RU_i) + \overline{P}_i \max(0, \tilde{t} - \#RU_i) + \sum_{n=1}^{\max(\tilde{t}, \#RU_i)} n \cdot RU_i \tag{A.41}$$

Therefore, the RU_i^{A-B} and RU_i^{B-B} can be evaluated as follows:

$$RU_i^{A-B} = \underline{P}_i \left(\min(\tilde{t}, \#RU_i) - 1\right) + \overline{P}_i \max(0, \tilde{t} - \#RU_i) + \sum_{n=1}^{\min(\tilde{t}, \#RU_i)} n \cdot RU_i \tag{A.42}$$

$$RU_i^{B-B} = \underline{P}_i \left(\min(\tilde{t}, \#RU_i) - \tilde{t}\right) + \overline{P}_i \max(0, \tilde{t} - \#RU_i) + \sum_{n=1}^{\min(\tilde{t}, \#RU_i)} n \cdot RU_i \tag{A.43}$$

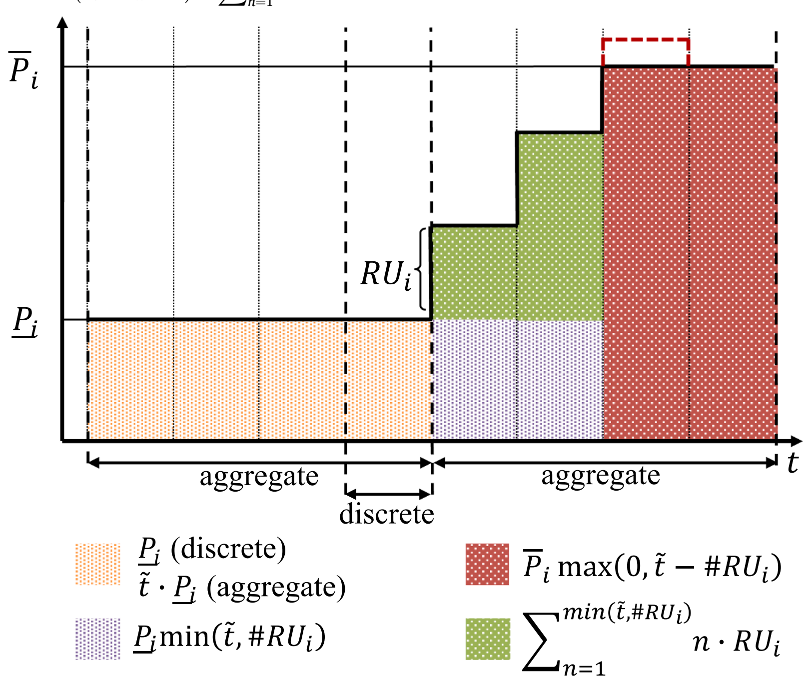

Fig. A1. Schematic representation of the maximum allowed generation increase between adjacent timesteps (A-B and B-B couples).

In a similar fashion, for the timestep couples A-B and B-B the *ramp-down* limit is evaluated as the difference between the generation related to ramping from maximum to minimum load, and the minimum allowed generation in the subsequent timestep. Given RD_i the maximum allowed generation decrease between subsequent timesteps for unit i, the minimum number of discrete timesteps needed to reduce the unit's output from maximum to minimum is defined as follows:

$$\#RD_i = \left\lfloor \frac{\overline{P}_i - \underline{P}_i}{RD_i} \right\rfloor \tag{A.44}$$

Rounding down $(\overline{P}_i - \underline{P}_i)/RD_i$ is motivated by the same reason presented for upwards ramping limits case and can be seen in Figure A.2 (red dashed line): in case $\#RD_i$ would be rounded up, the resulting generation at the end of the ramp would be lower than $\underline{P}_i$.

The maximum allowed generation for the discrete timestep (related to the timestep couple A-B) is equal to the maximum load:

$$p^A_{t-1} = \overline{P}_i \tag{A.45}$$

On the other hand, the unit's output for the previous aggregate timestep (thus related to the timestep couple A-B) is:

$$p^B_{t-1} = \tilde{t}\cdot\overline{P}_i \tag{A.46}$$

The minimum generation achievable in one aggregate timestep starting from the maximum load condition is defined by the following expression (a graphical representation of p_t can be seen in Figure A.2):

$$p^B_t = \underline{P}_i \max(0, \tilde{t} - \#RD_i) + \sum_{n=1}^{\min(\tilde{t}, \#RD_i)} (\overline{P}_i - n\cdot RD_i) \tag{A.47}$$

Therefore, the RD^{A-B}_i and RD^{B-B}_i can be evaluated as follows:

$$RD^{A-B}_i = \overline{P}_i(1 - \min(\tilde{t}, \#RD_i)) - \underline{P}_i \max(0, \tilde{t} - \#RD_i) + \sum_{n=1}^{\min(\tilde{t}, \#RD_i)} n\cdot RD_i \tag{A.48}$$

$$RD^{B-B}_i = \overline{P}_i(\tilde{t} - \min(\tilde{t}, \#RD_i)) - \underline{P}_i \max(0, \tilde{t} - \#RD_i) + \sum_{n=1}^{\min(\tilde{t}, \#RD_i)} n\cdot RD_i \tag{A.49}$$

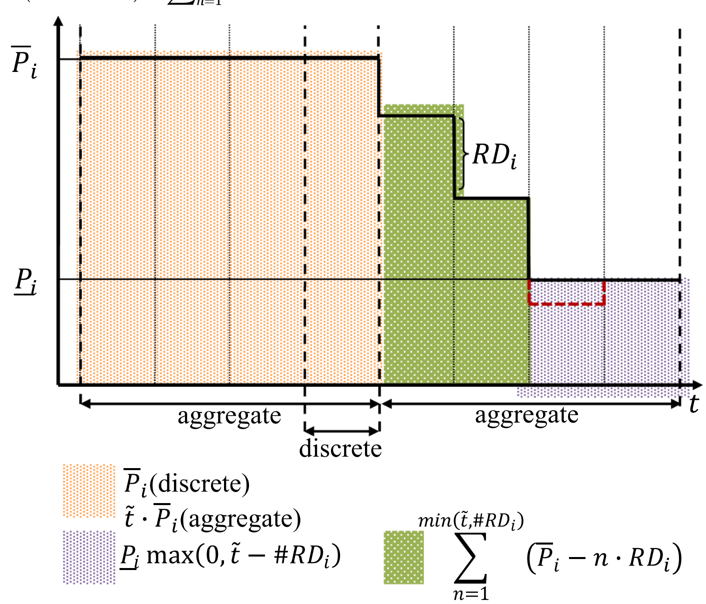

Fig. A2. Schematic representation of the maximum allowed generation decrease between adjacent timesteps (A-B and B-B couples).

Regarding the ramping limits related to *start-up* and *shut down*, they are equal to the maximum generation achievable before starting up and shutting down respectively.

Given SU_i the maximum allowed generation during the time-step of duration dt when the unit i switches on, the following term defines the minimum number of timesteps required to increase the output from SU_i to its maximum value.

$$\#SU_i = \left\lfloor \frac{\overline{P}_i - SU_i}{RU_i} \right\rfloor \tag{A.50}$$

The value comes from rounding down $(\overline{P}_i - SU_i)/RU_i$ for the same reasons seen above. Whatever the timestep prior the start-up is discrete or aggregate, the unit's output is null. Therefore $p^A_{t-1} = p^B_{t-1} = 0$. The maximum output achievable in one aggregate timestep at start up is limited by RU_i. It follows this expression, which also defines SU^{A-B}_i and SU^{B-B}_i:

$$\begin{aligned} SU^{A-B}_i &= SU^{B-B}_i \\ &= SU_i[1 + min(\tilde{t} - 1, \#SU_i)] + \sum_{n=1}^{min(\tilde{t}-1, \#SU_i)} n\cdot RU_i + \overline{P}_i[\tilde{t} - min(\tilde{t} - 1, \#SU_i) - 1] \end{aligned} \tag{A.51}$$

The abovementioned equation is graphically represented in Figure A.3.

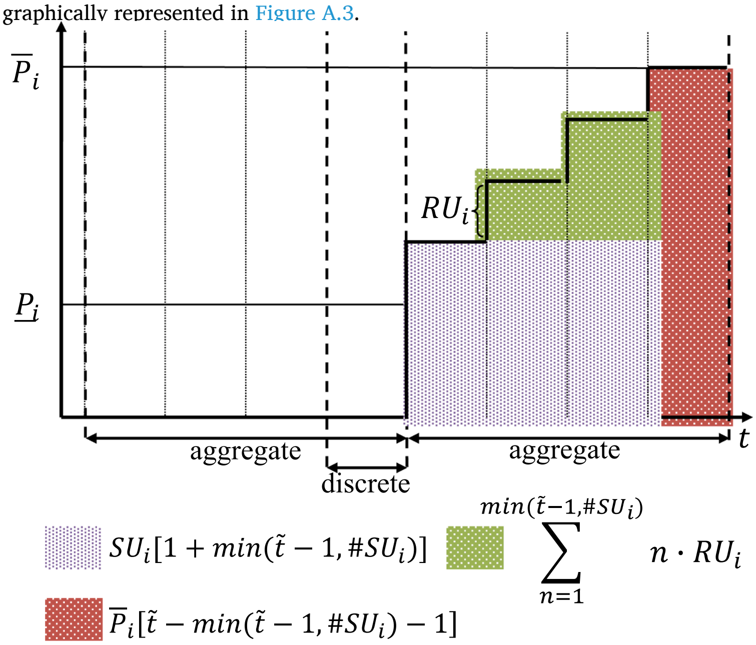

Fig. A3. Schematic representation of the maximum allowed generation at start up (A-B and B-B couples).

As for the start-up, also the ramping limit prior a shutdown is evaluated as the maximum allowed generation before turning the unit off. Given SD_i the maximum allowed generation in the timestep of duration dt prior shutdown, the following term defines the minimum number of timesteps required to decrease the output from its maximum value to SD_i.

$$\#SD_i = \left\lfloor \frac{\overline{P}_i - SD_i}{RD_i} \right\rfloor \tag{A.52}$$

The value comes from rounding down $(\overline{P}_i - SD_i)/RD_i$ for the same reasons seen in the subsections above. Whatever the timestep is discrete or aggregate, the unit's output is null if shutdown. Therefore $p_t^B = 0$. The maximum output allowed in one discrete timestep prior shutdown is the following:

$$SD_i^{A-B} = p_{t-1}^A = SD_i \tag{A.53}$$

When an aggregate timestep is considered, then the following expression holds:

$$SD_i^{B-B} = SD_i + \overline{P}_i\left[\tilde{t} - min\left(\tilde{t}-1, \#SD_i\right) - 1\right] + \sum_{n=1}^{min\left(\tilde{t}-1,\#SD_i\right)} \left(\overline{P}_i - n \cdot RD_i\right) \tag{A.54}$$

The above-mentioned equation is graphically represented in Figure A.4.

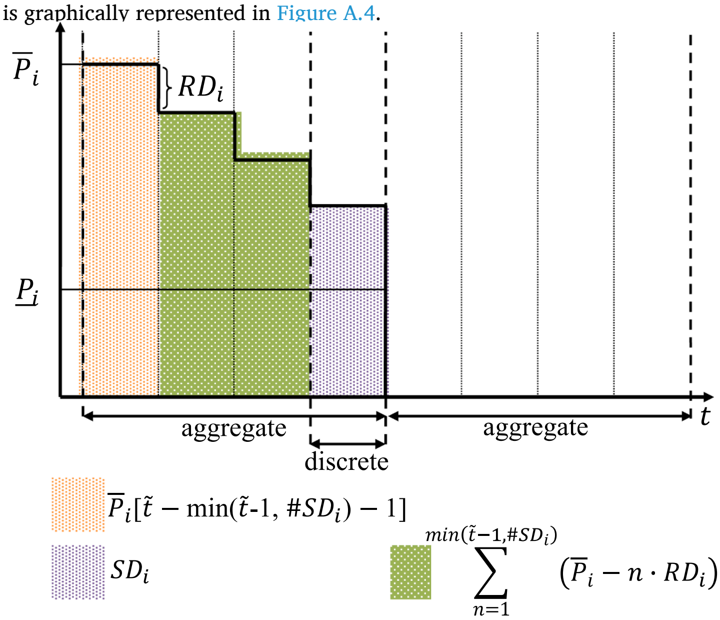

Fig. A4. Schematic representation of the maximum allowed generation before shut down (A-B and B-B couples).

- Modelling of minimum up- and down-time constraints

The minimum up- and down-time constraints considered are the ones described by Eq. (A.4) and (A.5). As for the ramping limits, the parameters related to these constraints must be evaluated again in light of the presence of aggregate timesteps in the reduced problem. In fact, UT_i and DT_i are related to discrete timesteps with original length of dt. Four new parameters are introduced: $\widetilde{UT}_{i,t}$, $\widetilde{DT}_{i,t}$,$\widetilde{UT}_i^{agg}$ and $\widetilde{DT}_i^{agg}$. They take part of constraints as follows:

$$\sum\nolimits_{\tau=t-\widetilde{UT}_{i,t}+1}^{t}\delta^{on}_{i,\tau}\leq z_{i,t}\,,\forall i\in G\backslash G^{on},\forall t\in\left[\widetilde{UT}_i^{agg},\cdots,|T|\right] \tag{A.55}$$

$$\sum\nolimits_{\tau=t-\widetilde{DT}_{i,t}+1}^{t}\delta^{off}_{i,\tau}\leq 1-z_{i,t}\,,\forall i\in G\backslash G^{on},\forall t\in\left[\widetilde{DT}_i^{agg},\cdots,|T|\right] \tag{A.56}$$

with $T = T^{disc}\cup T^{agg}$. The parameters $\widetilde{UT}_{i,t}$ and $\widetilde{DT}_{i,t}$ are defined for only $t\geq\widetilde{UT}_i^{agg}$ and $t\geq\widetilde{DT}_i^{agg}$ respectively.

At first, $\widetilde{UT}_i^{agg}$ and $\widetilde{DT}_i^{agg}$ are defined as the minimum number of discrete and aggregate timesteps corresponding to the original UT_i and DT_i. As a consequence, if $UT_i\leq\Delta t^{adv}$ and $DT_i\leq\Delta t^{adv}$, then $\widetilde{UT}_i^{agg} = UT_i$ and $\widetilde{DT}_i^{agg} = DT_i$. On the other hand, if $UT_i>\Delta t^{adv}$ and $DT_i>\Delta t^{adv}$ then the following expressions hold:

$$\widetilde{UT}_i^{agg}=\Delta t^{adv}+\left\lceil\frac{UT_i-\Delta t^{adv}}{\widetilde{t}}\right\rceil \tag{A.57}$$

$$\widetilde{DT}_i^{agg}=\Delta t^{adv}+\left\lceil\frac{DT_i-\Delta t^{adv}}{\widetilde{t}}\right\rceil \tag{A.58}$$

The parameter $\widetilde{UT}_{i,t}$ is not defined for $t<\widetilde{UT}_i^{agg}$, $\widetilde{UT}_{i,t}=\widetilde{UT}_i^{agg}$ for $t = \widetilde{UT}_i^{agg}$, while the following expression holds for $t>\widetilde{UT}_i^{agg}$:

$$\widetilde{UT}_{i,t} = max\left(0, UT_i-max\left(0,min\left(t-\Delta t^{adv},\left\lceil\frac{UT_i}{\widetilde{t}}\right\rceil\right)\right)\cdot\widetilde{t}+max\left(0,min\left(t-\Delta t^{adv},\left\lceil\frac{UT_i}{\widetilde{t}}\right\rceil\right)\right)\right) \tag{A.59}$$

The same applies for $\widetilde{DT}_{i,t}$: it is not defined for $t<\widetilde{DT}_i^{agg}$, $\widetilde{DT}_{i,t}=\widetilde{DT}_i^{agg}$ for $t = \widetilde{DT}_i^{agg}$, and the following expression holds for $t>\widetilde{DT}_i^{agg}$:

$$\widetilde{DT}_{i,t} = max\left(0, DT_i-max\left(0,min\left(t-\Delta t^{adv},\left\lceil\frac{DT_i}{\widetilde{t}}\right\rceil\right)\right)\cdot\widetilde{t}+max\left(0,min\left(t-\Delta t^{adv},\left\lceil\frac{DT_i}{\widetilde{t}}\right\rceil\right)\right)\right) \tag{A.60}$$

The reason why the Eq. (A.57) and (A.58) rounds up the value in the ratio is to ensure that the minimum up- and down-time are respected (this might not happen if rounding down). However, this comes at the cost that decisions could potentially be suboptimal. An example of how the constraints are modelled can be seen in Figure A.5.

$$UT_i = 10,\quad \widetilde{UT}_i^{agg}=6,\quad \widetilde{UT}_{i,t}=[-,-,-,-,-,6,\ 3,3,3]$$

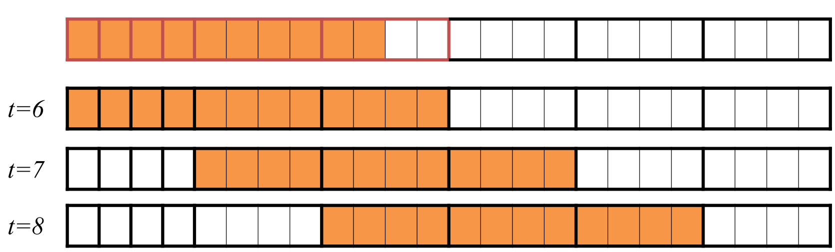

Fig. A5. Schematic representation of how the minimum up-time constraints are modelled in the reduced MILP with aggregate timesteps.

- Modelling of reserve constraints

The constraints defined by Eqs. (A.14)-(A.17), and (A.22) and (A.23) remain the same in the reduced problem, with the exception that the parameters are multiplied by $\widetilde{t}$ for those timesteps belonging to T^{agg}.

On the other hand, the constraints in Eqs. (A.18) and (A.19) become the following:

$$r^{up}_{i,t}\leq\left(p_{i,t-1}-r^{down}_{i,t-1}\right)-p_{i,t}+RU_i^{X-Y}z_{i,t-1}+SU_i^{X-Y}\delta^{on}_{i,t}\,,\forall i\in G,\forall t=T^{*},\ t\in T^{**},t-1\in T^{***} \tag{A.61}$$

$$r^{down}_{i,t}\leq p_{i,t}-\left(p_{i,t-1}+r^{up}_{i,t-1}\right)-p_{i,t}+RD_i^{X-Y}z_{i,t}+SD_i^{X-Y}\delta^{off}_{i,t}\,,\forall i\in G,\forall t=T^{*},\ t\in T^{**},t-1\in T^{***} \tag{A.62}$$

With RU_i^{X-Y}, RD_i^{X-Y}, SU_i^{X-Y}, SD_i^{X-Y}, T^{*}, T^{**} and T^{***} defined as in Table A.1.

References

[1] Carrión M, Arroyo JM. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans Power Syst Aug. 2006;21(3):1371–8. https://doi.org/10.1109/TPWRS.2006.876672.

[2] Fu Y, Shahidehpour M, Li Z. Security-constrained unit commitment with AC constraints. IEEE Trans Power Syst May 2005;20(2):1001–13. https://doi.org/10.1109/TPWRS.2005.846076.

[3] P. P. Gupta, P. Jain, S. Sharma, and R. Bhakar, "Reliability-Security Constrained Unit Commitment based on benders decomposition and Mixed Integer Non-Linear Programming," *2017 Int. Conf. Comput. Commun. Electron. COMPTELIX 2017*, pp. 328–333, Aug. 2017, doi: 10.1109/COMPTELIX.2017.8003988.

[4] B. Dipan Biswas, S. Kamalasadan, and S. Paudyal, "A two-stage combined UC-OPF model using mixed integer and semi-definite programming," *2021 IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. ISGT 2021*, Feb. 2021, doi: 10.1109/ISGT49243.2021.9372168.
[5] M. Jiang, H. Ge, Q. Guo, and H. Sun, "Network-Constrained AC Unit Commitment Based on Linear Approximation Techniques," *5th IEEE Conf. Energy Internet Energy Syst. Integr. Energy Internet Carbon Neutrality, EI2 2021*, pp. 2980–2985, 2021, doi: 10.1109/EI252483.2021.9713630.
[6] Ho YC. An explanation of ordinal optimization: Soft computing for hard problems. Inf Sci (Ny) Feb. 1999;113(3–4):169–92. https://doi.org/10.1016/S0020-0255(98)10056-7.
[7] Nan Y, Di Y, Zheng Z, Jiazhan C, Daojun C, Xiaoming W. Research on modelling and solution of stochastic SCUC under AC power flow constraints. IET Gener Transm Distrib 2018;12:3618–25. https://doi.org/10.1049/iet-gtd.2017.1845.
[8] Amjady N, Dehghan S, Attarha A, Conejo AJ. Adaptive robust network-constrained AC unit commitment. IEEE Trans Power Syst Jan. 2017;32(1):672–83. https://doi.org/10.1109/TPWRS.2016.2562141.
[9] Song CS, Park CH, Yoon M, Jang G. Implementation of PTDFs and LODFs for power system security. J Int Counc Electr Eng 2011;1(1):49–53. https://doi.org/10.5370/jicee.2011.1.1.049.
[10] Guo J, Fu Y, Li Z, Shahidehpour M. Direct calculation of line outage distribution factors. IEEE Trans Power Syst 2009;24(3):1633. https://doi.org/10.1109/TPWRS.2009.2023273.
[11] H. Ronellenfitsch, D. Manik, J. Hörsch, T. Brown, and D. Witthaut, "Dual theory of transmission line outages," 2017.
[12] M. Giuntoli, V. Biagini, and K. Schönleber, "Novel Formulation of PTDF and LODF Matrices for Security Constrained Optimal Power Flow for Hybrid AC and DC Grids," *Proc. 2019 IEEE PES Innov. Smart Grid Technol. Eur. ISGT-Europe 2019*, 2019, doi: 10.1109/ISGTEurope.2019.8905672.
[13] Xavier ÁS, Qiu F, Wang F, Thimmapuram PR. Transmission constraint filtering in large-scale security-constrained unit commitment. IEEE Trans Power Syst May 2019;34(3):2457–60. https://doi.org/10.1109/TPWRS.2019.2892620.
[14] A. S. Xavier, F. Qiu, and S. Ahmed, "Learning to Solve Large-Scale Security-Constrained Unit Commitment Problems," *INFORMS J. Comput.*, vol. 33, no. 2, pp. 739–756, Feb. 2019, Accessed: Oct. 11, 2021. [Online]. Available: https://arxiv.org/abs/1902.01697v2.
[15] Yang Y, Yang Z, Yu J, Xie K, Jin L. Fast economic dispatch in smart grids using deep learning: An active constraint screening approach. IEEE Internet Things J Nov. 2020;7(11):11030–40. https://doi.org/10.1109/JIOT.2020.2993567.
[16] Yang N, et al. Intelligent data-driven decision-making method for dynamic multisequence: An E-Seq2Seq-based SCUC expert system. IEEE Trans Ind Informatics May 2022;18(5):3126–37. https://doi.org/10.1109/TII.2021.3107406.
[17] Porras A, Pineda S, Morales JM, Jimenez-Cordero A. Cost-driven screening of network constraints for the unit commitment problem. IEEE Trans Power Syst Jan. 2023;38(1):42–51. https://doi.org/10.1109/TPWRS.2022.3160016.
[18] Wang J, Shahidehpour M, Li Z. Security-constrained unit commitment with volatile wind power generation. IEEE Trans Power Syst 2008;23(3):1319–27. https://doi.org/10.1109/TPWRS.2008.926719.
[19] Bertsimas D, Litvinov E, Sun XA, Zhao J, Zheng T. Adaptive robust optimization for the security constrained unit commitment problem. IEEE Trans Power Syst 2013;28(1):52–63. https://doi.org/10.1109/TPWRS.2012.2205021.
[20] Wang C, Fu Y. Fully parallel stochastic security-constrained unit commitment. IEEE Trans Power Syst Sep. 2016;31(5):3561–71. https://doi.org/10.1109/TPWRS.2015.2494590.
[21] Wu L, Shahidehpour M, Li T. Stochastic security-constrained unit commitment. IEEE Trans Power Syst May 2007;22(2):800–11. https://doi.org/10.1109/TPWRS.2007.894843.
[22] P. Liu, Z. Wu, W. Gu, P. Yu, J. Du, and X. Luo, "A novel acceleration strategy for N-l contingency screening in distribution system," *IEEE Power Energy Soc. Gen. Meet.*, vol. 2020-August, Aug. 2020, doi: 10.1109/PESGM41954.2020.9281445.
[23] Cominesi SR, Farina M, Giulioni L, Picasso B, Scattolini R. A two-layer stochastic model predictive control scheme for microgrids. IEEE Trans Control Syst Technol Jan. 2018;26(1):1–13. https://doi.org/10.1109/TCST.2017.2657606.
[24] Balasubramanian J, Grossmann IE. Approximation to multistage stochastic optimization in multiperiod batch plant scheduling under demand uncertainty. Ind Eng Chem Res Jul. 2004;43(14):3695–713. https://doi.org/10.1021/ie030308+.
[25] Morales-España G. Tight MIP formulations of the power-based unit commitment problem. Oper Res 2015.
[26] Constante-Flores GE, Conejo AJ, Qiu F. AC network-constrained unit commitment via conic relaxation and convex programming. Int J Electr Power Energy Syst 2022;134. https://doi.org/10.1016/j.ijepes.2021.107364.
[27] Tejada-Arango DA, Sanchez-Martin P, Ramos A. Security constrained unit commitment using line outage distribution factors. IEEE Trans Power Syst Jan. 2018;33(1):329–37. https://doi.org/10.1109/TPWRS.2017.2686701.
[28] Ardakani AJ, Bouffard F. "Prediction of umbrella constraints". 20th Power Syst Comput Conf PSCC Aug. 2018;2018. https://doi.org/10.23919/PSCC.2018.8450586.
[29] A. J. Ardakani and F. Bouffard, "Identification of umbrella constraints in DC-based security-constrained optimal power flow," pp. 1–1, Nov. 2014, doi: 10.1109/PESGM.2014.6938802.
[30] Holzer JT, Chen Y, Wu Z, Pan F, Veeramany A. Fast simultaneous feasibility test for security constrained unit commitment. IEEE Trans Power Syst 2023:1–10. https://doi.org/10.1109/TPWRS.2023.3265269.
[31] Lodi A. The heuristic (dark) side of MIP solvers. Stud Comput Intell 2013;434:273–84. https://doi.org/10.1007/978-3-642-30671-6_10/COVER.
[32] "How do I implement lazy constraints in Gurobi? – Gurobi Help Center." https://support.gurobi.com/hc/en-us/articles/360013197972 (accessed Dec. 31, 2022).
[33] Constantinescu EM, Anitescu M, Lee S, Rocklin M, Zavala VM. A computational framework for uncertainty quantification and stochastic optimization in unit commitment with wind power generation. IEEE Trans Power Syst 2011;26(1):431–41. https://doi.org/10.1109/TPWRS.2010.2048133.
[34] S. Schmitt, I. Harjunkoski, M. Giuntoli, J. Poland, and X. Feng, "Fast Solution of Unit Commitment Using Machine Learning Approaches," *ENERGYCON 2022 - 2022 IEEE 7th Int. Energy Conf. Proc.*, 2022, doi: 10.1109/ENERGYCON53164.2022.9830191.
[35] S. Babaeinejadsarookolaee *et al.*, "The Power Grid Library for Benchmarking AC Optimal Power Flow Algorithms," Aug. 2019, doi: 10.48550/arxiv.1908.02788.
[36] "Data Miner 2." https://dataminer2.pjm.com (accessed Dec. 31, 2022).
[37] Optimization Gurobi LLC, "Gurobi Optimizer Reference Manual," 2022.
[38] Castelli AF, Pilotti L, Monchieri A, Martelli E. Optimal design of aggregated energy systems with (N-1) reliability: MILP models and decomposition algorithms. Appl Energy Feb. 2024;356:122002. https://doi.org/10.1016/J.APENERGY.2023.122002.
[39] Morales-España G, Gentile C, Ramos A. Tight MIP formulations of the power-based unit commitment problem. OR Spectr Oct. 2015;37(4):929–50. https://doi.org/10.1007/s00291-015-0400-4.
[40] Knueven B, Ostrowski J, Watson J-P. On mixed integer programming formulations for the unit commitment problem. INFORMS J Comput 2020;34(4):857–76. https://doi.org/10.1287/ijoc.2019.0944.
[41] Van Den Bergh K, Delarue E. An improved method to calculate injection shift keys. Electr Power Syst Res May 2016;134:197–204. https://doi.org/10.1016/J.EPSR.2016.01.020.
[42] Shahidehpour M, Yamin H, Li Z. Market Operations in Electric Power Systems: Forecasting, Scheduling, and Risk Management. Hoboken, New Jersey: John Wiley & Sons; 2002.