

Tensor Convolution-Based Aggregated Flexibility Estimation in Active Distribution Systems

Chrysostomou, Demetris; Torres, Jose Luis Rueda; Cremer, Jochen Lorenz

DOI

[10.1109/TSG.2024.3453667](https://doi.org/10.1109/TSG.2024.3453667)

Publication date

2024

Document Version

Final published version

Published in

IEEE Transactions on Smart Grid

Citation (APA)

Chrysostomou, D., Torres, J. L. R., & Cremer, J. L. (2024). Tensor Convolution-Based Aggregated Flexibility Estimation in Active Distribution Systems. *IEEE Transactions on Smart Grid*, 16(1), 87 - 100. <https://doi.org/10.1109/TSG.2024.3453667>

Important note

To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.

Tensor Convolution-Based Aggregated Flexibility Estimation in Active Distribution Systems

Demetris Chrysostomou^{1b}, *Graduate Student Member, IEEE*, José Luis Rueda Torres^{2b}, *Senior Member, IEEE*,
and Jochen Lorenz Cremer^{3b}, *Member, IEEE*

Abstract—Power system operators require advanced applications in the control centers to tackle increasingly variable power transfers effectively. One urgently needed application concerns estimating the feasible available aggregated flexibility from a power system network, which can be effectively deployed to mitigate issues in interconnected networks. This paper proposes the TensorConvolution+ algorithm to address the above application. Unlike related literature approaches, TensorConvolution+ estimates the density of feasible flexibility combinations to reach a new operating point within the p-q flexibility area. This density can improve the decision-making of system operators for efficient and safe flexibility deployment. The proposed algorithm applies to radial and meshed networks, is adaptable to new operational conditions, and can consider scenarios with disconnected flexibility areas. Using convolutions and tensors, the algorithm efficiently aggregates the combinations of flexibility providers' adjustable power output that can occur for each flexibility area set point. Simulations on the meshed Oberrhein and radial CIGRE test networks illustrate the effectiveness of TensorConvolution+ for flexibility estimation with high numerical confidence and a minor computing effort. Additional simulations highlight how system operators can interpret the estimated density of feasible flexibility combinations for decision-making purposes, the algorithm's capability to estimate disconnected flexibility areas, and adapt to new operating conditions.

Index Terms—Convolution, distribution networks, flexibility aggregation, TSO-DSO coordination, tensors.

NOMENCLATURE

Abbreviation

<i>DFC</i>	Density of feasible combinations.
<i>DN</i>	Distribution network.
<i>DSO</i>	Distribution system operator.
<i>FA</i>	Flexibility area.
<i>FSP</i>	Flexibility service provider.
<i>OB0</i>	Oberrhein network of substation 0.
<i>OB1</i>	Oberrhein network of substation 1.

Manuscript received 28 February 2024; revised 15 July 2024; accepted 26 August 2024. Date of publication 2 September 2024; date of current version 26 December 2024. This work was supported in part by the research program “MegaMind - Measuring, Gathering, Mining and Integrating Data for Self-Management in the Edge of the Electricity System,” and in part by the Dutch Research Council (NWO) through the Perspectief Program under Grant P19-25. Paper no. TSG-00335-2024. (*Corresponding author: Demetris Chrysostomou.*)

The authors are with the Department of Electrical Sustainable Energy, Delft University of Technology, 2628CD Delft, The Netherlands (e-mail: D.Chrysostomou@tudelft.nl; j.l.ruedatorres@tudelft.nl; j.l.cremer@tudelft.nl).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TSG.2024.3453667>.

Digital Object Identifier 10.1109/TSG.2024.3453667

<i>OC</i>	Operating condition of the DN.
<i>OP</i>	Operating point as p, q of the PCC.
<i>OPF</i>	Optimal power flow.
<i>PCC</i>	Point of common coupling.
<i>PF</i>	Power flow.
<i>TSO</i>	Transmission system operator.

Parameters

δp	Resolution of FA in active power [MW].
δq	Resolution of FA in reactive power [MVAR].
c_v	Voltage sensitivity threshold [p.u.].
l_v	Loading sensitivity threshold [%].
l_{\max}	Maximum loading network constraint [%].
v_{\max}	Maximum voltage network constraint [p.u.].
v_{\min}	Minimum voltage network constraint [p.u.].

Operations

*	Convolution.
o	Hadamard product.
⊗	Tensor convolution.
⊥	Variable independence.
×	Set cartesian product.

Indices and Sets

o	Index for origin, the PCC.
Ω^S	Infinite set of flexibility shifts.
Ω^{S^o}	Set of flexibility shifts at the PCC from all FSP combinations.
$\tilde{\Omega}_i^S$	Set of flexibility shifts offered by FSP i .
Ω_i^S	Set of flexibility shifts offered by FSP i as observed at the PCC.
$\Omega_{\mathcal{K}}^S$	Set of flexibility shifts at the PCC achievable by combinations from the FSP sequence \mathcal{K} .
Ω^{FS^o}	Set of feasible flexibility shifts at the PCC from all FSP combinations.
Ω^C	Set of all flexibility shift combinations at the PCC.
$\Omega_{s^o}^C$	Set of flexibility shift combinations at the PCC achieving s^o .
Ω_{γ}^C	Set of flexibility shift combinations at the PCC causing impact at network component γ .
Ω_{γ, s^o}^C	Set of flexibility shift combinations at the PCC achieving s^o and causing impact at network component γ .
$\Omega_{\mathcal{K}}^C$	Set of flexibility shift combinations at the PCC from the FSP sequence \mathcal{K} .

$\Omega_{s^o}^{FC}$	Set of feasible flexibility shift combinations at the PCC achieving s^o .
Ω^{FSP}	Set of FSPs.
Ω_{γ}^{FSP}	Set of FSPs whose flexibility impacts γ .
$\Omega_{\gamma}^{FSP'}$	Set of FSPs whose flexibility does not impact γ .
$\Omega^{B'}$	Set of network buses that cannot reach their constraints from the offered FSP shifts.
Ω^B	Set of network buses.
$\Omega^{L'}$	Set of network lines and transformers that cannot reach their constraints from the offered FSP shifts.
Ω^L	Set of network lines and transformers.

Variables and Functions

α_f	Percentage of pixels correctly assigned as feasible over the total number of feasible pixels [%].
α_r	Percentage of pixels correctly assigned as reachable over the total number of reachable pixels [%].
$\Delta l_{z,s}$	Loading impact on z , from the flexibility shift s [p.u.].
Δp	Shift in active power [MW].
Δq	Shift in reactive power [MVAR].
$\Delta v_{b,s}$	Voltage impact on b , from the flexibility shift s [p.u.].
γ	A network bus, line or transformer.
Γ_b	Tensor of all voltage shifts from flexibility combinations from FSPs in Ω_b^{FSP} .
\mathcal{K}	Sequence of k FSPs.
$\Phi_{i,b}(\cdot)$	Function of voltage impacts caused on b by FSP i .
$\Phi_{j,z}(\cdot)$	Function of loading impacts caused on z by FSP j .
π	A combination of FSP shifts as a tuple.
$\Psi(\cdot)$	Function describing a sequence of tensor convolutions.
θ	Weight for the OPF-based algorithm [–].
$\tilde{F}_{\mathcal{K}}(\cdot)$	Function describing the convolution of a sequence of indicator functions.
Υ_{γ}	Matrix with population of combinations feasible for γ from all FSPs.
Ξ_b	Binary tensor classifying feasible combinations for b .
Ξ_z	Binary tensor classifying feasible combinations for z .
A_{γ}	Matrix with population of combinations feasible for γ from impactful FSPs.
Ac_f	Average α_f over different FA estimations [%].
Ac_r	Average α_r over different FA estimations [%].
b	A bus.
$C(\cdot)$	Function describing summation of all dimensions of a tensor except the first 2.
$F_i(\cdot)$	Indicator function of FSP i .
$H(\cdot)$	Function returning a set of individual FSP shifts participating in the combination π .
i, j, r	An FSP.
j_d	FSP offering discrete setpoints.
$l_{z,0}$	Loading of z under the initial OC [%].
$l_{z,\pi}$	Loading of z if π is applied [%].
$m_{\mathcal{K}}(s^o)$	Number of shift combinations of FSP sequence \mathcal{K} reaching s^o [–].

p	Active power [MW].
q	Reactive power [MVAR].
s	Shift in active and reactive power.
$T_{i,b}$	Tensor of impact of i on b .
$v_{b,0}$	Voltage of b under the initial OC [p.u.].
$v_{b,\pi}$	Voltage of b if π is applied [p.u.].
z	A line or transformer.

I. INTRODUCTION

THE COORDINATION between transmission system operators (TSOs) and distribution system operators (DSOs) faces challenges in data exchange [1] as the flexibility services from providers connected to the distribution networks (DNs) become important for services such as balancing and congestion management. TSOs need to anticipate the available DN flexibility, to effectively use this flexibility in their operation. DSOs must ensure that said flexibility services respect the DN operational constraints and use these services in distribution-level markets [2]. Between existing approaches for TSO-DSO coordination, the DSO-managed approach requires exchanging non-sensitive information from the DSOs to the TSOs. This information is flexibility areas (FAs); areas in the active and reactive power plane designating the extreme values of flexibility that the DN can “offer” to the transmission network at their point of common coupling (PCC) [3]. The DSO-managed coordination process involves two tasks for DSOs. In the first task, the DSO obtains offers from flexibility service providers (FSPs), estimates the theoretically feasible FA as in [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], and informs the TSO of that FA. The TSO can request an operating point (OP) within the FA from the DSO. The DSO performs the second task, optimizing the individual FSP shifts achieving the TSO request while respecting the DN constraints and minimizing the costs as in [15], [16].

In existing FA estimation algorithms, the area surrounded by the FA curve’s extreme values is mainly assumed as equally reachable and feasible. However, a different set of flexibility combinations can reach each FA OP, and the algorithms cannot guarantee the FSP availability and actual desired response [16], [17]. Therefore, each FA OP should not be viewed as equal, but represent the density of feasible flexibility combinations through which the OP can be reached. This additional information on FA OP can influence the TSO to select OP based on expected delivery, reliability, and effectiveness. Currently, to the best of the authors’ knowledge, no algorithm provides information on the density of feasible flexibility combinations (DFC). Hence, TSOs may be unable to select safe and efficient shifts for FA OPs. This paper focuses on FA estimation, considering all combinations and providing the DFC for each FA OP efficiently.

Early distribution network FA estimation studies [7], [12] effectively proposed power flow (PF)-based algorithms. These algorithms provide simple and coherent methods for FA estimation but have limitations in exploring the flexibility area space [12] and high computational time [7]. Optimal power flow (OPF)-based approaches are faster and apply multi-objective optimization such as ϵ -constraint

method [4], [17], weighted sum method [5], [10], and radial reconstruction-based method [3], [6]. OPF-based approaches provide straightforward and efficient algorithms to identify the FA limits, improving the potential of system operators to include flexibility in their decision-making. However, OPF-based approximations may have limited validity in meshed networks, as highlighted by [10], [13], [14]. The vast majority of studies use radial networks [3], [4], [5], [6], [7], [8], [9], [12], [13], [14], [17].

The authors of [11] introduced an efficient chance-constrained FA on meshed networks, showcasing the benefits of power flow routers for FAs. The results of [11] showed relatively slower performance than the other OPF-based algorithms. A limitation of OPF-based algorithms, as identified by [5], [13], is the incapability to deal with FAs that are disconnected, which might be the case for FSPs with limited offered flexibility setpoints. This paper's proposed algorithm performs in radial and meshed network topologies and can estimate disconnected FAs.

The objective of existing FA estimation approaches is to identify the limits of the aggregated DN flexibility [3], [4], [5], [6], [7], [8], [9], [10], [11], [14]. Evaluating each possible FSP shift combination through existing approaches is intractable, as they are tailored to the above objective. The authors of [17] considered the reliability of each FSP in the algorithm as the confidence in delivering the offered flexibility. The approach of [17] does not evaluate all possible flexibility shift combinations to reach an OP between any set of FSPs. Nevertheless, the results of [17] showcased great insights on the inner area of FAs, but the task was computationally expensive with an average time of 970s. This paper's proposed FA estimation approach considers all possible flexibility shift combinations and reports the DFC to reach each FA OP.

Existing FA estimation approaches mainly require the OC of the DN as input data to perform OPF and PF simulations as [4], [5], [6], [9], [10], [12], [14]. However, DNs have typically limited real-time observability with measurement units on limited network components [18], [19], [20], [21]. To deal with uncertainties from renewable sources, [11] applied a chance-constrained method, and [3] applied robust optimization. Reference [7] included the probability distributions of forecast errors to determine the probability of feasibility. The proposed algorithm can use the FA estimated under expected DN OC (bus voltages, power injections, and line loading), to adapt and approximate the FA in correlated real-time OC with observability limited to a subset of network components.

Deep learning-based algorithms have been recently explored for the FA estimation problem [22], and in tasks related to FA estimation such as OPF [23], [24] and PF [25], [26]. Deep learning models can improve efficiency in performing these tasks. However, some limitations concerning low generalization to unseen network topologies and requiring large training datasets can challenge the application of deep learning in algorithms for FA estimation. The proposed algorithm does not employ deep learning models and does not require training.

The proposed TensorConvolution+ algorithm deals with issues on the exploration of inner FA, evaluates all FSP combinations and informs on the DFC to reach each FA OP.



Fig. 1. DSO-managed approach for TSO-DSO Coordination. Steps order (1).

Existing alternative algorithms explore the extreme limits that flexibility combinations can achieve. Evaluating all combinations informs the system operators on which PCC operating points have more feasible flexibility options. A higher DFC can correspond to safety regarding network constraints, less dependency on specific FSPs, and more options to optimize costs in algorithms such as [15], [16]. Alternative approaches cannot tractably evaluate all possible discrete combinations, whereas TensorConvolution+ is time efficient. In addition, TensorConvolution+ handles the limitation of existing FA estimation algorithms in dealing with disconnected FAs and discrete FSPs. This paper's contributions are:

- Developing the first method to evaluate all possible discrete combinations of flexibility, improving the information encompassed in FAs.
- Introducing the application of convolutions in flexibility estimation, and analyzing useful properties of convolution to simplify the FA problem's complexity.
- Introducing the application of tensors in flexibility estimation, to store and evaluate the feasibility of flexibility combinations.

Case studies on the 70-bus and 109-bus meshed medium voltage Oberrhein networks and the 15-bus CIGRE medium voltage radial network with photovoltaic and wind plant modules show the algorithm's capability to work for diverse network topologies, the need for DFC in FAs, the above contributions and the algorithm's adaptability to partially observable OCs.

The following sections are (Section II) flexibility estimation algorithm; introducing the problem formulation and objectives, (Section III) tensor convolutions; the application of tensors and convolutions in the algorithm, (Section IV) case studies; on the need for DFC and algorithm contributions, and (Section V) conclusion.

II. FLEXIBILITY ESTIMATION ALGORITHM

In DSO-managed coordination, as illustrated in Fig. 1 the first DSO task concerns the FA estimation. This task informs the TSO about the available shifts $s^o = [\Delta p^o, \Delta q^o]^T$ from the initial PCC OP $[p^o, q^o]^T$, considering the DN network constraints. The FA estimation impacts the TSO selection of an aggregated shift, and subsequently the DSO second task of FSP shift optimization. Therefore, the FA estimation problem requires (i) exploring possible FSP shift combinations and (ii) evaluating whether each combination results in a feasible OC for the network constraints.

The requirement (i) on FA estimation approaches determines which area can be reached using FSP shifts. The set of FSPs is Ω^{FSP} . The generic infinite set of shifts is $\Omega^S = \{s|s = [\Delta p, \Delta q]^T \in \mathbb{R}^2\}$. Each FSP $i \in \Omega^{FSP}$ offers a set of shifts as $\tilde{\Omega}_i^S \subset \Omega^S$. The function $\sigma : \Omega^S \rightarrow \Omega^S$ maps each shift of FSP i from its bus to the PCC. Thus, the set of shifts from each FSP, as observed at the PCC is $\Omega_i^S = \{s|s = \sigma(\tilde{s}), \forall \tilde{s} \in \tilde{\Omega}_i^S\} \subset \Omega^S$. The set of all shift combinations at the PCC between all FSPs is:

$$\Omega^C = \Omega_1^S \times \Omega_2^S \times \dots \times \Omega_{|\Omega^{FSP}|}^S, \in \mathbb{R}^{2|\Omega^{FSP}|}, \quad (1)$$

$$|\Omega^C| = \prod_{i \in FSP} |\Omega_i^S|, \quad (2)$$

where \times is a set cartesian product and $|\cdot|$ is a set cardinality. Each combination $\pi \in \Omega^C$ corresponds to one tuple of FSP-shifts, and $\pi \rightarrow \{s_1, s_2, \dots, s_{|\Omega^{FSP}|}\} := H(\pi)$. Thus, function $H(\pi)$ represents obtaining a set of individual FSP shifts participating in the combination π .

The issue fulfilling requirement (i) is that the number of flexibility combinations grows exponentially as FSPs and their offered shifts increase. Due to this issue, state-of-the-art approaches do not explore all possible FSP shift combinations. The proposed approach estimates all possible combinations, i.e., all $\pi \in \Omega^C$.

The requirement (ii) on FA estimation approaches determines whether each discrete FSP shift combination considers the network constraints. Let the reachable shifts at the PCC s^o expanded by the index π for each $\pi \in \Omega^C$ as:

$$s_\pi^o = \sum_{s \in H(\pi)} s, \quad (3)$$

as each combination π leads to one s^o . Using (3), the set of reachable shifts at the PCC from all possible combinations is:

$$\Omega^{S^o} = \{s_\pi^o, \forall \pi \in \Omega^C\} \subset \Omega^S. \quad (4)$$

As shown in Fig. 2, multiple combinations π can reach the same PCC OP shift s^o , i.e., $\exists s_\pi^o = s_{\hat{\pi}}^o$ s.t. $\pi, \hat{\pi} \in \Omega^C$. Therefore, the authors expand Ω^C by the index s^o , indicating the set of combinations leading to each s^o as:

$$\Omega_{s^o}^C = \{\pi | s_\pi^o = s^o \forall \pi \in \Omega^C\} \subset \Omega^C. \quad (5)$$

The cardinality $|\Omega_{s^o}^C|$ represents the number of possible combinations reaching s^o . For example in Fig. 2, for $s^o = [3, 3]^T$, the $|\Omega_{s^o}^C| = 2$ (from π_1, π_2). However, out of the 2 combinations, only one leads to a feasible voltage (π_2). Therefore, an FA estimation approach should account for the network constraints. Feasible shift combinations require network constraints to be fulfilled at the resulting shifted OCs. Every shift combination π when applied, can impact the voltage magnitude $v_{b,\pi}$ of every network bus $b \in \Omega^B$, and the loading $l_{z,\pi}$ of every line or transformer $z \in \Omega^L$. The set of feasible combinations reaching each s^o is:

$$\Omega_{s^o}^{FC} = \left\{ \pi | (v_{\min} \leq v_{b,\pi} \leq v_{\max} \forall b \in \Omega^B, |l_{z,\pi}| \leq l_{\max} \forall z \in \Omega^L), \forall \pi \in \Omega_{s^o}^C \right\} \subset \Omega_{s^o}^C. \quad (6)$$

v_{\max}, v_{\min} are the maximum and minimum allowed voltage for network buses, l_{\max} is the maximum allowed loading for

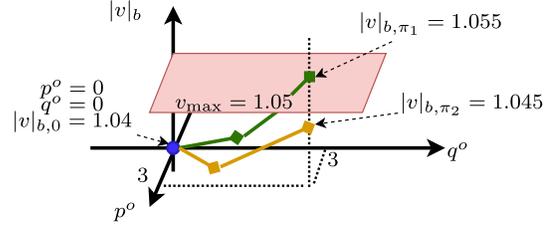


Fig. 2. Example for 2 shift combinations (π_1 as \rightarrow , π_2 as \rightarrow) from the initial OP (\bullet) to reach $p^o + \Delta p^o = 3\text{MW}$, $q^o + \Delta q^o = 3\text{MVAR}$, but only π_2 feasible for bus b due to the maximum voltage constraint (\square).

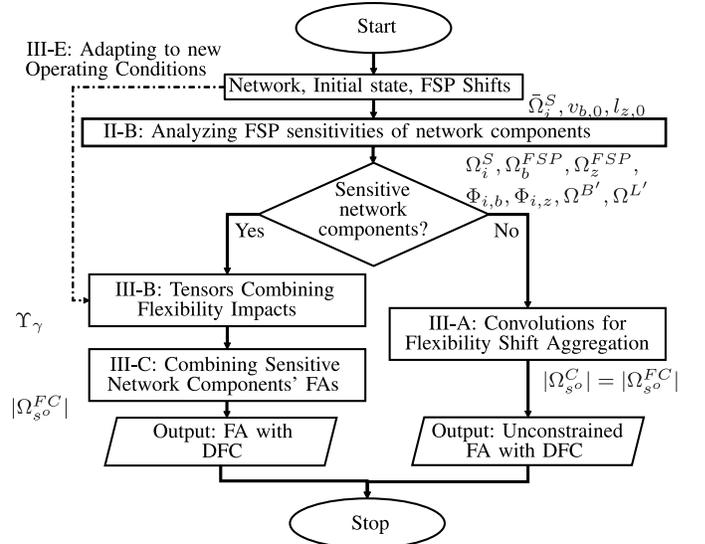


Fig. 3. Overview of the proposed flexibility estimation algorithm.

network lines and transformers. The set of all shifts at the PCC that are feasible is $\Omega^{FS^o} = \{s^o | 1 \leq |\Omega_{s^o}^{FC}| \forall s^o \in \Omega^{S^o}\} \subset \Omega^{S^o}$. For each s^o its DFC = $|\Omega_{s^o}^{FC}| / \max_{s^o \in \Omega^{S^o}} |\Omega_{s^o}^{FC}|$, the normalized cardinality of $|\Omega_{s^o}^{FC}|$.

The issue of fulfilling requirement (ii) is that evaluating the impact of all combinations π on all network components is computationally expensive. Hence, evaluating the impact of all possible combinations is intractable in existing approaches. State-of-the-art approaches simplify the issue to identify all s^o for which at least 1 feasible shift combination exists, i.e., Ω^{FS^o} . The proposed approach evaluates all possible combinations and estimates $|\Omega_{s^o}^{FC}|$ of all $s^o \in \Omega^{FS^o}$.

The proposed algorithm is TensorConvolution+ with input as the network topology, the initial OCs, and the FSPs. When adapting FAs for new OCs, the needed input is a subset of the network components' new voltage and loading magnitudes. Hyperparameters include the resolution $\delta p, \delta q$ of Ω^S , i.e., increments in Ω^S elements, and the sensitivity thresholds c_v, c_l . Fig. 3 shows the algorithm's steps. TensorConvolution+ initially runs PF simulations to generate samples for requirements (i) and (ii). The algorithm decomposes the flexibility constrained by each network component (e.g., bus, line), and uses tensors and convolutions to efficiently process the samples for requirement (ii). Then, TensorConvolution+ returns the FA constrained by all network components. If all combinations

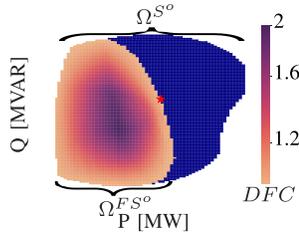


Fig. 4. FA from the proposed algorithm with the DFC of feasible FA points (red) and not feasible FSP shift combinations (blue) from the initial OP (star).

fulfill the network constraints, i.e., $\Omega_{s^o}^{FC} = \Omega_{s^o}^C \forall s^o \in \Omega^{S^o}$, then convolutions are applied to deal with requirement (i). Fig. 4 shows the output FA of the proposed algorithm. Each blue-colored pixel has a value of 1 and represents a reachable but not feasible OP for the network constraints. Each pixel with values between (1, 2] represents a reachable and feasible OP and its DFC, i.e., larger values have more feasible options than lower values. The area covered by all colored pixels represents the reachable set Ω^{S^o} from FSP offers, related to requirement (i). The feasible area represents Ω^{FS^o} and DFC, related to requirements (i), (ii).

A. Reducing Required Power Flow Simulations

Computing all inputs to (6) requires estimating $v_{b,\pi}, l_{z,\pi}$ as:

$$v_{b,\pi} = v_{b,0} + \sum_{s \in H(\pi)} \Delta v_{b,s} \quad \forall b \in \Omega^B, \forall \pi \in \Omega_{s^o}^C, \quad (7)$$

$$l_{z,\pi} = l_{z,0} + \sum_{s \in H(\pi)} \Delta l_{z,s} \quad \forall z \in \Omega^L, \forall \pi \in \Omega_{s^o}^C, \quad (8)$$

where $v_{b,0}, l_{z,0}$ are the OC's values for $b \in \Omega^B$ and $z \in \Omega^L$. Variables $\Delta v_{b,s}, \Delta l_{z,s}$ are the shift s impacts on b, z , estimated using PF simulations. The number of PF simulations needed to explore all possible FSP shift combinations is (2). To simplify this combination complexity, the authors apply:

Assumption 1: The impact of each FSP's output shift on a network component is not affected by other FSPs' output shifts, i.e., $\Delta v_{b,\hat{s}} \perp \Delta v_{b,\tilde{s}} \forall \hat{s} \in \Omega_{\hat{i}}^S, \forall \tilde{s} \in \Omega_{\tilde{i}}^S, \forall \hat{i} \neq \tilde{i} \in \Omega^{FSP}$, $\Delta l_{z,\hat{s}} \perp \Delta l_{z,\tilde{s}} \forall \hat{s} \in \Omega_{\hat{i}}^S, \forall \tilde{s} \in \Omega_{\tilde{i}}^S, \forall \hat{i} \neq \tilde{i} \in \Omega^{FSP}$.

When adopting assumption 1, the proposed algorithm requires one PF simulation for each possible FSP shift. Hence, the required PF simulations are decreased from (2) to $\sum_{i \in \Omega^{FSP}} |\Omega_i^S|$.

B. Analyzing FSP Sensitivities of Network Components

This section analyzes the sensitivities of network components to FSP shifts through the impacts $\Delta v_{b,s}, \Delta l_{z,s}$. This analysis further reduces the complexity $O((|\Omega^B| + |\Omega^L|) \cdot |\Omega_{s^o}^C|)$ in estimating (7) and (8). The analysis starts with:

- *Observation (a):* each network component's voltage or loading is not sensitive to all FSPs.
- *Observation (b):* not all network components can reach their voltage or loading limitations due to the FSP shifts.

Exploiting (a), the FSP sensitivity sets for each $b \in \Omega^B$ and $z \in \Omega^L$ are:

$$\Omega_b^{FSP} = \left\{ i | c_v \leq \max_{s \in \Omega_i^S} (|\Delta v_{b,s}|), i \in \Omega^{FSP} \right\}, \quad (9)$$

$$\Omega_z^{FSP} = \left\{ i | c_l \leq \max_{s \in \Omega_i^S} (|\Delta l_{z,s}|), i \in \Omega^{FSP} \right\}, \quad (10)$$

where c_v, c_l , are sensitivity thresholds. The FSPs that do not impact the constraints of a network component are $\Omega_{\gamma}^{FSP'} = \{i | i \in \Omega^{FSP} \setminus \Omega_{\gamma}^{FSP}\} \forall \gamma \in \Omega^B \cup \Omega^L$. For example, in a network with 2 feeders connected to the PCC, the components on the first feeder can be insensitive to shifts from FSPs connected to the second feeder. The sets in (9)-(10) replace Ω^C with Ω_{γ}^C , the set of combinations for which $\gamma \in \Omega^B \cup \Omega^L$ is sensitive to its constraints. This analysis reduces the constraint-evaluated combinations from $|\Omega_{s^o}^C|$ to $|\Omega_{\gamma,s^o}^C| \forall \gamma \in \Omega^B \cup \Omega^L$; the set of sensitive combinations per component as:

$$\Omega_{\gamma,s^o}^C = \left\{ \pi | \sum_{s \in H(\pi)} s = s^o \forall \pi \in \Omega_{\gamma}^C \right\} \subset \Omega_{\gamma}^C. \quad (11)$$

In DN OCs with higher margin from the network constraints, $|\Omega_{\gamma,s^o}^C| \ll |\Omega_{s^o}^C|$.

Exploiting (b) the sets of non-sensitive components are the ones whose voltage or loading cannot reach the constraints when accumulating the highest impact from the FSPs, as:

$$\Omega^{B'} = \left\{ b | \left(v_{b,0} + \sum_{i \in \Omega_b^{FSP}} \max_{s \in \Omega_i^S} (\Delta v_{b,s}) \leq v_{max} \right) \wedge \left(v_{min} \leq v_{b,0} + \sum_{i \in \Omega_b^{FSP}} \min_{s \in \Omega_i^S} (\Delta v_{b,s}) \right), \forall b \in \Omega^B \right\}, \quad (12)$$

$$\Omega^{L'} = \left\{ z | \left(l_{z,0} + \sum_{i \in \Omega_z^{FSP}} \max_{s \in \Omega_i^S} (\Delta l_{z,s}) \leq l_{max} \right) \wedge \left(l_{min} \leq l_{z,0} + \sum_{i \in \Omega_z^{FSP}} \min_{s \in \Omega_i^S} (\Delta l_{z,s}) \right), \forall z \in \Omega^L \right\}. \quad (13)$$

The analysis of observations (a) and (b), reduces the combinations evaluated in (7)–(8) by replacing $\Omega_{s^o}^C$ with Ω_{γ,s^o}^C , Ω^B with $\Omega^B \setminus \Omega^{B'}$ and Ω^L with $\Omega^L \setminus \Omega^{L'}$. Thus, the complexity becomes $O((|\Omega^B \setminus \Omega^{B'}| + |\Omega^L \setminus \Omega^{L'}|) \cdot |\Omega_{\gamma,s^o}^C|)$.

III. TENSOR CONVOLUTIONS

A. Convolutions for Flexibility Shift Aggregation

This section introduces convolutions, to aggregate shifts without considering network constraints. Convolutions can efficiently combine the flexibility sets, and accumulate the number of combinations leading to each s^o , i.e., $|\Omega_{s^o}^C|$. Relating, the Minkowski sum can be applied to efficiently combine flexibility sets without considering network constraints [27], e.g., combining feasible flexibility sets from multiple feeders connected to the PCC. However, the

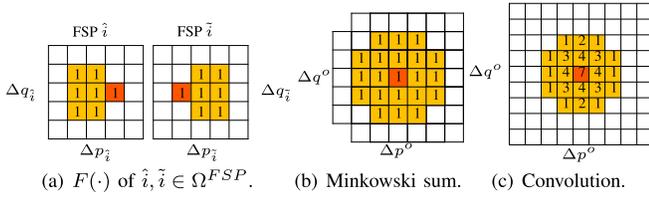


Fig. 5. FAs for Minkowski sum and Convolution of $F(\cdot)$ for $\hat{i}, \tilde{i} \in \Omega^{FSP}$. The FAs include squares of feasible shifts (■), and initial operating points (●).

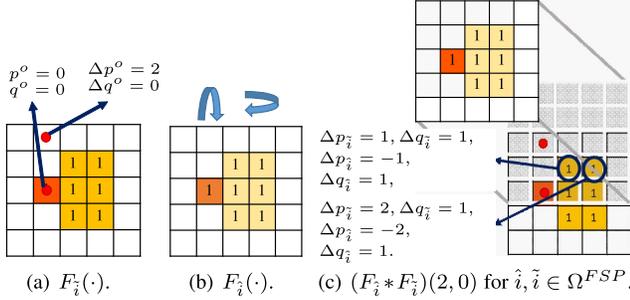


Fig. 6. Convolution of $F_{\hat{i}}(\cdot), F_{\tilde{i}}(\cdot)$ for $\hat{i}, \tilde{i} \in \Omega^{FSP}$ at $\Delta p^o = 2, \Delta q^o = 0$.

Minkowski sum does not consider how many combinations from the input sets lead to each s^o . In Fig. 5(b), the Minkowski sum, and in Fig. 5(c), the two-dimensional (2D) discrete convolution of two flexibility sets. In Fig. 5(c), each resulting point includes the number of combinations reaching it.

The proposed algorithm aggregates shifts without considering constraints in two cases. First, when $\Omega^{B'} = \Omega^B, \Omega^{L'} = \Omega^L$, thus $\Omega_{s^o}^C = \Omega_{s^o}^{FC}$. Second, for each component $\gamma \in \Omega^B \cup \Omega^L$ where $1 \leq |\Omega_{\gamma}^C| < |\Omega^C|$, the algorithm explores all Ω^C combinations; the cartesian product between Ω_{γ}^C and all FSPs in $\Omega^{FSP'}$, i.e., $\Omega^C = \Omega_1^S \times \dots \times \Omega_{|\Omega_{\gamma}^{FSP'}|}^S \times \Omega_{\gamma}^C$.

Let any 2 FSPs (\hat{i}, \tilde{i}) offering shifts $\Omega_{\hat{i}}^S$ and $\Omega_{\tilde{i}}^S$ respectively, as in Fig. 5(a). The 2D discrete convolution for each shift $s^o = [\Delta p^o, \Delta q^o]^T$ is defined as:

$$(F_{\hat{i}} * F_{\tilde{i}})(\Delta p^o, \Delta q^o) = \sum_{\Delta p = -\infty}^{\infty} \sum_{\Delta q = -\infty}^{\infty} (F_{\hat{i}}(\Delta p, \Delta q) \cdot F_{\tilde{i}}(\Delta p^o - \Delta p, \Delta q^o - \Delta q)), \quad (14)$$

where $(\cdot * \cdot)(x, y)$ is a convolution of two functions for the input x, y , and $F_i \forall i \in \Omega^{FSP}$ is an indicator function as:

$$F_i(s) = \begin{cases} 1, & \text{if } s \in \Omega_i^S, \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

Fig. 6 visualizes the convolution process for a single $\Delta p^o, \Delta q^o$. The circled overlapping pixels in Fig. 6(c) represent all possible combinations reaching $p^o + \Delta p^o, q^o + \Delta q^o$. Useful properties from the convolution [28] are applied as:

- 1) Associativity and commutativity: The order of FSPs \hat{i}, \tilde{i} (or more) does not affect the convolution output.
- 2) Impulse response: The convolution of an FA with a delta function results in the FA shifted by the delta offset.

The Appendix A proves a third property stating: *The discrete convolution of shifts between k FSPs considers all possible discrete combinations between these k FSPs for s^o . The*

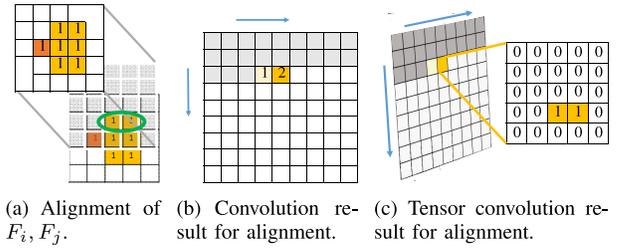


Fig. 7. Convolution of $F(\cdot)$ for $i, j \in \Omega^{FSP}$ for an s^o . Out of all reachable shifts (■), 2 combinations are possible for the alignment (○).

resulting value for each s^o is the sum of all combinations from the k FSPs that can reach it.

The proposed approach describes the combined FA function for a sequence \mathcal{K} of convolutions for k FSPs as:

$$\tilde{F}_{\mathcal{K}}(s^o) = \begin{cases} m_{\mathcal{K}}(s^o), & \text{if } s^o \in \Omega_{\mathcal{K}}^S, \\ 0, & \text{otherwise,} \end{cases} \quad (16)$$

where:

$$\Omega_{\mathcal{K}}^C = \Omega_1^S \times \Omega_2^S \times \dots \times \Omega_k^S, \quad (17)$$

$$\Omega_{\mathcal{K}}^S = \left\{ s_{\mathcal{K}} | s_{\mathcal{K}} = \sum_{s \in \pi} s, \forall \pi \in \Omega_{\mathcal{K}}^C \right\} \subset \Omega^{s^o}. \quad (18)$$

$m_{\mathcal{K}}(s^o) \in \mathbf{N}$ is the number of shift combinations of FSP sequence $\mathcal{K} = 1, 2, \dots, k$ reaching s^o . For the remainder of the paper, $F_{\hat{i}} * F_{\tilde{i}}$ corresponds to applying (14) for all $[\Delta p^o, \Delta q^o]^T \in \Omega_{\mathcal{K}}^S$, where $\mathcal{K} = \hat{i}, \tilde{i}$. This operation is efficient and widely available within computer vision and machine learning software libraries.

B. Tensors Combining Flexibility Impacts

This section describes the algorithm to estimate the FA of each constraint-sensitive network component, i.e., $b \in \Omega^B \setminus \Omega^{B'}, z \in \Omega^L \setminus \Omega^{L'}$. The proposed algorithm uses tensors to efficiently explore, represent, and process the information obtained through PF simulations.

Intuitively, as shown in Fig. 6, during a 2D discrete convolution, every FSP combination reaching s^o is accumulated. However, applying (7)-(8) before the accumulation is needed to check whether each combination is feasible or not for the network constraints. Hence, the authors propose avoiding the summation step of convolution and storing the alignment of each step in new dimensions. For example, using convolution, the 2 combinations of Fig. 7(a) result in an entry value of 2 for the matrix in Fig. 7(b). Through the proposed tensor-convolution as in Fig. 7(c), the entry for $\Delta p^o = 0, \Delta q^o = 2$ is a matrix of the element-wise multiplication between the F_j , and the shifted-flipped F_i . The resulting tensors store the information of which combinations π are available through this alteration.

After this tensor-convolution process, the combination of 2 FSPs results in a 4 dimensional tensor, and each additional FSP adds 2 dimensions. The function for this process is Ψ , and the tensor convolution operation is $\hat{*}$.

$$\Psi(F_1, F_2, \dots, F_k) = F_1 \hat{*} F_2, \dots, \hat{*} F_k, \in \mathbf{R}^{2k}. \quad (19)$$

As shown in Section II-B observations, the impactful FSPs for each component's constraints and the sensitive network components are limited, mitigating dimensionality issues. Nevertheless, if a limited RAM requires reduced tensor dimensionality, the algorithm can aggregate pairs of FSPs into one only for the component causing the issue. The algorithm determines which pair is closer through the electrical distance shown in [29]; the impedance of the lines between each FSP pair.

1) *Obtaining Combination Impacts*: For each restrictive network component $b \in \Omega^B \setminus \Omega^{B'}$, $z \in \Omega^L \setminus \Omega^{L'}$, for each of their impactful FSPs $i \in \Omega_b^{FSP}$, $j \in \Omega_z^{FSP}$, the impact functions $\Phi_{i,b}(s)$, $\Phi_{j,z}(s)$ are:

$$\Phi_{i,b}(s) = \begin{cases} \Delta v_{b,s}, & \text{if } s \in \Omega_i^S, \\ 0, & \text{otherwise,} \end{cases} \quad (20)$$

$$\Phi_{j,z}(s) = \begin{cases} \Delta l_{z,s}, & \text{if } s \in \Omega_j^S, \\ 0, & \text{otherwise.} \end{cases} \quad (21)$$

Estimating (7)-(8) is needed to check if the resulting $v_{b,\pi}$ are feasible for the network constraints. Thus, the algorithm needs to sum the associated Δv caused by every FSP within π for every π effective on b . The authors propose iteratively taking $i \in \Omega_b^{FSP}$ and applying:

$$T_{i,b} = \Psi(F_0, \dots, F_{i-1}) \hat{*} \Phi_{i,b} \hat{*} \Psi(F_{i+1}, \dots, F_{|\Omega_b^{FSP}|}), \in \mathbb{R}^{2|\Omega_b^{FSP}|}. \quad (22)$$

The above result, $T_{i,b}$, is the tensor of voltage impacts from i , on b , whose entries represent each π . $T_{i,b}$ does not include the voltage shifts caused by other FSPs. After obtaining $T_{i,b} \forall i \in \Omega_b^{FSP}$, the proposed algorithm performs element-wise tensor addition as:

$$\Gamma_b = \sum_{i \in \Omega_b^{FSP}} T_{i,b}, \in \mathbb{R}^{2|\Omega_b^{FSP}|}. \quad (23)$$

This addition aggregates the contribution to the voltage shift from all impactful FSPs for component b for each combination π , i.e., each Γ_b entry value is equal to $\sum_{s \in H(\pi)} \Delta v_{b,s}$ for a unique $\pi \in \Omega_b^C$. The boolean tensor $T_b^{bool} = F_0 \hat{*} \dots \hat{*} F_{|\Omega_b^{FSP}|}$ shows which combinations π belong to Ω_b^C , i.e., T_b^{bool} entries are 1 where a combination exists and 0 where not. The proposed algorithm applies the following to classify each combination as feasible or not:

$$\Xi_b = \Lambda^v(\Gamma_b + v_{b,0} \cdot \mathbf{1}_{\Gamma_b}) \circ T_b^{bool}, \in \mathbb{R}^{2|\Omega_b^{FSP}|}, \quad (24)$$

$$\Lambda^v(v_b) = \begin{cases} 1, & \text{if } v_{min} \leq v_b \leq v_{max}, \\ 0, & \text{otherwise.} \end{cases} \quad (25)$$

where $\mathbf{1}_{\Gamma_b}$ is a tensor of ones, with the shape of Γ_b . The operation \circ is the tensor Hadamard product. The addition $\Gamma_b + v_{b,0} \cdot \mathbf{1}_{\Gamma_b}$ estimates the resulting voltages at component b for every possible combination of flexibility shifts. The filter Λ^v returns 1 for all combinations within the network constraints. The Hadamard product sets 0 for all combinations not offered by the FSPs. The corresponding variables for loading components are Γ_z , T_z^{bool} , $\Lambda^l(l_z)$, and Ξ_z .

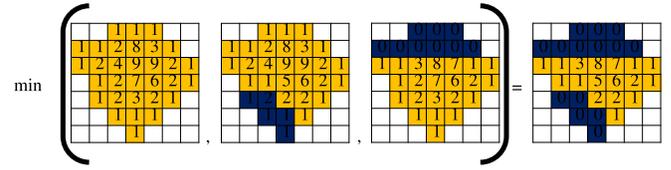


Fig. 8. Element-wise minimum of FAs.

2) *Tensor to Flexibility Mapping*: The Ξ_b , Ξ_z have dimensionalities of $2 \cdot |\Omega_b^{FSP}|$, $2 \cdot |\Omega_z^{FSP}|$ respectively. The first 2 dimensions correspond to the PCC s^o ; as in Fig. 7(c). The rest constitute the constraint-validated combinations reaching each PCC s^o . Therefore, summing all tensor entries except the first 2 dimensions returns the flexibility area constrained by the component b or z . Intuitively, this operation restores the accumulation step of convolution. Using the Einstein summation convention allows easy and efficient summing operations over multiple dimensions of tensors in Python [30]. The function $C : \xi \rightarrow 2$ represents the summation of all tensor's Ξ entries from any ξ dimensions to the first 2 dimensions:

$$A_\gamma = C(\Xi_\gamma), \in \mathbb{R}^2, \quad (26)$$

where $\gamma \in \Omega^B \cup \Omega^L$ is any network component. Each A_γ entry corresponds to a different s^o . Each entry value of A_γ is the number of combinations in Ω_γ^C , feasible for γ for an s^o .

3) *Adding FSPs Insensitive for Components*: A_γ excludes contributions from FSPs causing negligible voltage or loading impacts on $\gamma \in \Omega^B \cup \Omega^L$. The proposed algorithm performs 2D convolution between A_γ and the F of all elements in $\Omega_\gamma^{FSP'} = \{j'_0, j'_1, \dots\}$ to consider the feasibility of all possible combinations in Ω^C for component γ , as:

$$\Upsilon_\gamma = A_\gamma * F_{j'_0} * F_{j'_1}, \dots, \in \mathbb{R}^2. \quad (27)$$

Let μ, τ be the row and column indices of $\Upsilon_\gamma \forall \gamma \in \Omega^B \cup \Omega^L$. The bijective function λ maps each s^o to a unique μ, τ , i.e., $\lambda: s^o \rightarrow (\mu, \tau)$. Each entry value of Υ_γ is the number of combinations in Ω^C feasible for γ for a unique s^o .

C. Combining Sensitive Network Components' FAs

To combine the FA of all components into one, the authors apply:

Assumption 2: $|\Omega_{s^o}^{FC}| \approx \min_{\gamma \in \Omega^B \cup \Omega^L} (\Upsilon_{\gamma, \lambda(s^o)})$.

Using assumption 2 allows estimating the FA of each component before approximating the final FA at the PCC as their element-wise minimum. For instance, Fig. 8 illustrates a scenario with 3 network components, one non-restrictive and two restrictive due to different constraints.

D. Dirac Functions for Non-Linear FSP

The proposed algorithm can deal with disconnected areas and non-linear FSPs, exploring the impulse response property introduced in Section III-A. Non-linear FSPs can cause disconnected areas as in [5], [13]. Non-linear FSPs could be on-load tap changers [5], [13] or generators/loads offering specific shift set points rather than a range of shifts [31]. Depending

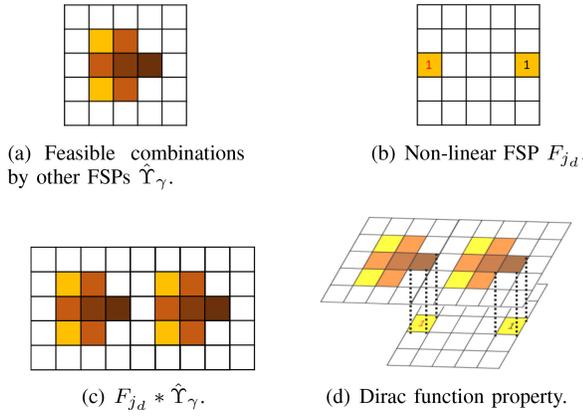


Fig. 9. Convolution of a discrete variable's $F_{j_d}(\cdot)$ with the feasible combinations from other FSPs. The convolution result of (c) is the same as displacing (a) by the (b) shifts as in (d).

on the network component sensitivity or insensitivity to the non-linear FSPs, the algorithm performs different estimations to combine the non-linear FSP flexibility with the FA from the rest of the FSPs.

If the component $\gamma \in \Omega^B \cup \Omega^L$ is insensitive to the non-linear FSP $j_d \in \Omega_\gamma^{FSP'}$, the proposed algorithm initially performs (27), excluding j_d to get $\hat{\Upsilon}_\gamma$, as in Fig. 9(a). To add the flexibility from non-linear FSPs one could convolve $\hat{\Upsilon}_\gamma$ with the indicator function of j_d , i.e., F_{j_d} of Fig. 9(b), as in Fig. 9(c). Alternatively, exploiting the Dirac function property of convolutions, the algorithm displaces $\hat{\Upsilon}_\gamma$ by each shift $\Delta p, \Delta q$ from j_d and sums these displaced results, as in Fig. 9(d). The summation results to Υ_γ .

If j_d is sensitive for $b \in \Omega^B$, the algorithm ignores the j_d impact on b until after (23). Then, for each j_d shift $s \in \Omega_{j_d}^S$ whose impact is $\Delta v_{b,s}$, the matrix $A_{b,s}^d$ is:

$$A_{b,s}^d = C \left(\Lambda^v (\Gamma_b + (v_{b,0} + \Delta v_{b,s}) \cdot \mathbf{1}_{\Gamma_b}) \circ T_b^{bool} \right). \quad (28)$$

The (28) encompasses adding the impacts of all linear FSP shifts to the impact of a non-linear FSP value, filtering the results based on the network constraints as in (24), (25), and summing feasible combinations for each s as in (26). Thus, each $A_{b,s}^d$ measures all feasible combinations between the linear FSPs and a non-linear FSP value s . Subsequently, the proposed algorithm sums $A_{b,s}^d \forall s \in \Omega_{j_d}^S$ displaced by the s offsets. The summation result describes A_γ , used in (27). The (28) also holds for $z \in \Omega^L$, but with loading variables instead.

E. Adaptability to Partially Observable Operating Conditions

Let FSPs that offer the same flexibility shifts for related OCs. In that case, the proposed algorithm assumes the shift impacts $\Delta v_{b,s} \forall s \in \Omega_i^S \forall i \in \Omega^{FSP}, \forall b \in \Omega^B, \Delta l_{z,s} \forall s \in \Omega_i^S \forall i \in \Omega^{FSP}, \forall z \in \Omega^L$ to be similar. Therefore, the tensors $\Gamma_b, \Gamma_z, T_b^{bool}, T_z^{bool}$ are also similar for these related OCs. The proposed algorithm can store the tensors $\Gamma_b, \Gamma_z, T_b^{bool}, T_z^{bool}$, and adapt the flexibility in subsequent FA estimations by applying Λ^v, Λ^l on the stored tensors and the new OCs $l_{z,0} \forall z \in \Omega^L \setminus \Omega^{L'}, v_{b,0} \forall b \in \Omega^B \setminus \Omega^{B'}$. The proposed algorithm

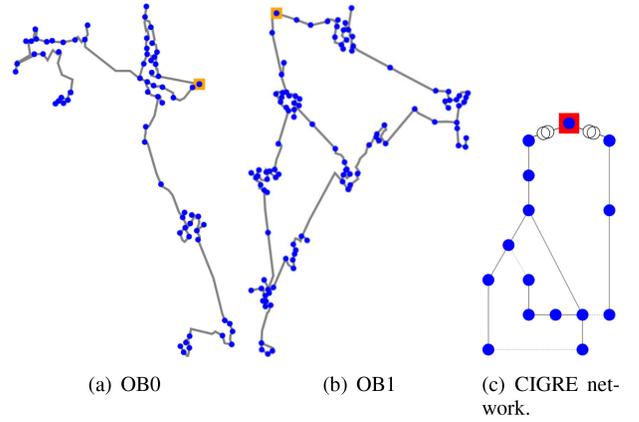


Fig. 10. Test network lines (—), buses (●), high to medium voltage transformer stations (■), transformers (⊙), external grid (■).

applies tensor train decomposition to reduce the space needed to save multiple high-dimensional tensors; a method that allows efficient representation of high-dimensional tensors with a small number of parameters without losing significant information [32]. For example, let an FA be estimated for expected (e.g., day ahead) OCs. Let real-time measurement units be placed on the DN components that are sensitive to constraints and flexibility shifts, i.e., $\Omega^B \setminus \Omega^{B'}, \Omega^L \setminus \Omega^{L'}$. TensorConvolution+ stores the tensors of the expected OCs FA and uses the real-time measurements from the sensitive components to approximate the partially observable OCs FA.

IV. CASE STUDIES

Fig. 10 illustrates the test networks; the meshed medium voltage Oberrhein network's substations 0 (OB0) (70 buses) and 1 (OB1) (109 buses), and the radial CIGRE medium voltage network (15 buses). The authors modified OB0 and OB1 to get meshed networks and provide more challenging scenarios for the algorithm. These modifications led to initial OCs with a minimum voltage in OB0 of $0.95p.u.$ and OB1 of $0.958p.u.$. These modifications increased the sensitivity of network components to FSPs. The algorithm's inputs were the network and the locations of load and generator FSPs. The algorithm's parameters were the $\delta p, \delta q, c_l, c_v, l_{max}, v_{max}, v_{min}$. In all case studies the loading constraint was $l_{max} = 100\%$ and the threshold parameters were $c_v = 0.001p.u., c_l = 1\%$ for the CIGRE network, $c_v = 0.0001p.u., c_l = 1\%$ for OB0, and $c_v = 0.005p.u., c_l = 1\%$ for OB1. The other inputs and parameters are referenced below and vary between the case studies. In all case studies, the FSPs were assumed to offer flexibility covering any p, q setpoint with apparent power less than their initial apparent power except in case study C where one FSP was non-linear and case study B.3.

Study A considered 3 FSPs in the CIGRE network, offering their full flexibility. The 3 FSPs were loads 3, 11 and generator 8. The FSP costs were 40 €/MW, 50 €/MW, and 60 €/MW. The exhaustive baseline approach performed PF simulations for all possible shift combinations between the FSPs for each $\delta p = 0.25MW, \delta q = 0.25MVAR$ increment. The voltage constraints were $v_{min} = 0.95p.u., v_{max} = 1.05p.u.$ The

simulations recorded the cost and feasibility for the network constraints per combination. Study B.1 considered 2 FSPs in radial OB0, load 57, and generator 29 to compare the TensorConvolution+ performance to the exhaustive PF baseline with a $\delta p = 0.025\text{MW}$, $\delta q = 0.025\text{MVAR}$ increment. The voltage constraints were $v_{\min} = 0.95p.u.$, $v_{\max} = 1.05p.u.$. Missing values were linearly imputed. Study B.2 considered 360 estimations of FAs for 2–15 FSPs on the OB0, and OB1. Each estimated FA had a random set of FSPs between the network loads and generators. For the cases with $|\Omega^{FSP}| \geq 10$, the random FSPs were sampled from 2 sets relating to different network regions. The 240 estimations had narrow voltage constraints with $c_{\min} = 0.95p.u.$, $c_{\max} = 1.05p.u.$, and 120 estimations wide voltage constraints with $c_{\min} = 0.9p.u.$, $c_{\max} = 1.1p.u.$. In narrow voltage constraints estimations, around 400 pixels (as in Fig. 14), and in wide voltage constraints around 670 pixels were explored. The number of pixels determined the values of δp , δq from the total capacity of flexibility offers per scenario. For each estimated FA, 1000 samples of flexibility shifts were generated with a Monte Carlo baseline. Each feasible sample generated from the baseline was assigned to a pixel on the proposed algorithm's estimated FA. If the pixel was estimated feasible by the TensorConvolution+, then the estimation was correct. The percentage between the correctly assigned feasible pixels over the total number of feasible pixels explored is a_f . The percentage between the explored and not explored pixels is a_r . The 1000-sample sets were sampled from 2 conventional (uniform, Kumaraswamy) and 1 harder distributions, amounting to 3000 samples. The CIGRE network shows the effects of radial topologies and higher resolutions with 11 FSPs, loads 9, 14, 16, 17, and generators 0, 1, 2, 3, 4, 5, 6 and narrow voltage constraints. The output for the CIGRE network was around 10000 pixels with $\delta p = 0.01\text{MW}$, $\delta q = 0.02\text{MVAR}$. Study B.3 compares 3 FAs estimated using an OPF-based algorithm and TensorConvolution+. The FAs included 5 and 6 FSPs on the CIGRE network. Study C considered 7 FSPs offering any setpoint in their flexibility range, loads 12, 14, 16 and generators 0, 1, 2, 3, 8. The wind plant FSP (generator 8) only offered full curtailment (2 setpoints) to produce a disconnected area. The voltage constraints were set at $0.94\text{--}1.06p.u.$, a challenging case where not-feasible shifts exist, but both areas in the disconnected FA include feasible shifts. Study D considered 9 FSPs, loads 3, 5, 6, 9, 17, and generators 4, 5, 6, 8 in the CIGRE network. These FSPs varied in offered capacity, S from 0.03MVA to 1.5MVA . The step-size of $\delta p = 0.1\text{MW}$, $\delta q = 0.2\text{MVAR}$ led to approximately 1500 pixels while neglecting 3 FSPs of 0.03 , 0.03 , and 0.04MVA . The voltage constraints were $v_{\min} = 0.95p.u.$, $v_{\max} = 1.05p.u.$. Study E scenarios considered visually different FAs for the same FSPs between initial and altered OCs. The FSPs in the CIGRE network case were loads 3, 5, 6, 9, 17 and generator 8, with $\delta p = 0.2\text{MW}$, $\delta q = 0.2\text{MVAR}$. For the altered OCs, the power factor of loads 0, 7, 8, 12 and 15 was reversed for the CIGRE network. In OB0, shifts were randomly sampled for all non-FSP generators and loads using normal distribution centered at the expected OC's values, with a standard deviation of 0.2. The FSPs in the OB0 case were loads 18, 22 and

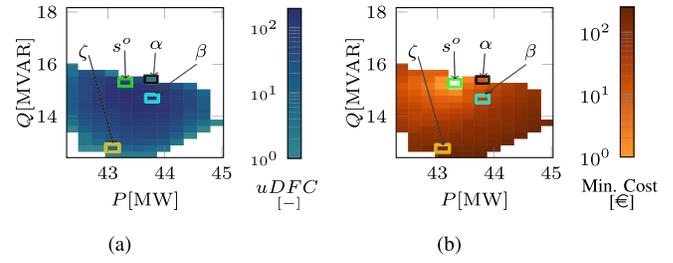


Fig. 11. Selecting safe and cheap shifts with the unnormalized DFC (uDFC) in (a). Additional information on shift minimum costs (Min. Cost) in (b). Possible shifts include α (\square); very cheap, but not safe, β (\square); cheap and safe, and ζ : (\square) expensive and not safe.

TABLE I
COMPARING SHIFTS α , β , ζ WITH UNNORMALIZED DFC (UDFC),
NUMBER OF NOT-FEASIBLE COMBINATIONS (NFC), FEASIBLE
COMBINATIONS PERCENTAGE (FCP), AND MINIMUM COST (MIN.COST)

Shift	uDFC [-]	NFC[-]	FCP[%]	Min. Cost [€]
α	2	174	1.1	37.9
β	193	0	100	66.5
ζ	5	0	100	177.5

generators 26, 50, with $\delta p = 0.068\text{MW}$, $\delta q = 0.068\text{MVAR}$. The voltage constraints were $v_{\min} = 0.95p.u.$, $v_{\max} = 1.05p.u.$

The algorithm's output is an FA Pandas data frame. The times referenced correspond to estimating the data frame. Simulations were performed on an Intel Core i7-1185G7 CPU with 16 GB RAM and an NVIDIA A100 GPU with 40GB VRAM. The GPU is available in Google Colab [33]. The algorithm's implementation in Python also included the PandaPower, SciPy, PyTorch, scikit-learn, and Numpy libraries.

A. DFC Improving TSOs Flexibility Shift Selection

This case study exemplifies the improvement in flexibility shift selection using DFC. Fig. 11 illustrates the simulation results on the CIGRE network, and Table I summarizes the results for shifts α , β , ζ . Fig. 12 illustrates the PF results for the cheapest flexibility shift combinations reaching α , β , ζ . The α shift's results are unsafe as multiple buses have approximately $0.95pu$ voltage magnitude, and only 1.1% of the available combinations for α are feasible. The few combinations available for ζ make this shift's combinations expensive, and less reliable (dependent on specific FSPs). The 193 feasible combinations for β make this selection less dependent on specific combinations. In addition, β has safer results for the network constraints than α and costs cheaper than ζ . Thus, using DFC, the TSOs can select safe and cost-efficient shifts.

B. Analysing TensorConvolution+ Performance

This case study visually and quantitatively evaluates the TensorConvolution+ performance in terms of FA DFC, range, and computational speed.

1) *Performance Analysis for DFC*: Fig. 13 illustrates the resulting FA of the proposed approach and the ground truth,

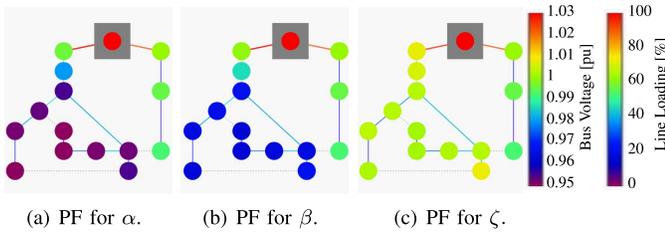


Fig. 12. PF results for cheapest flexibility combinations for shifts α , β , ζ , where network voltage constraints are $0.95 - 1.05p.u.$

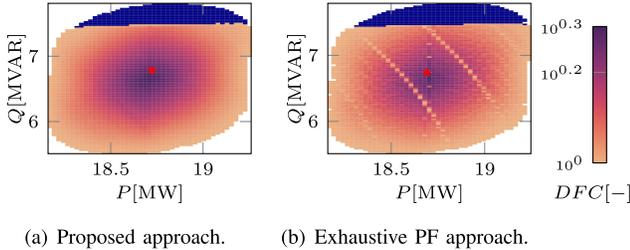


Fig. 13. The FA for TensorConvolution+ in (a) and PF-based approach in (b). The FAs include the DFC (■) for feasible shifts from the initial OP (★) and not-feasible shifts (■).

TABLE II
EVALUATION OF ALGORITHM FOR OB0, OB1 NETWORKS (NET.),
WITH NARROW AND WIDE VOLTAGE CONSTRAINTS (VC)

		Hard		Uniform		Kumaraswamy		
VC	Net.	$ \Omega^{\text{FSP}} $	$Ac_f\%$	$Ac_r\%$	$Ac_f\%$	$Ac_r\%$	$Ac_f\%$	$Ac_r\%$
Narrow	OB0	2-7	99	99	99	100	99	99
Narrow	OB1	2-7	99	99	100	100	100	100
Wide	OB0	10-15	99	99	99	99	98	98

an exhaustive PF-based approach on the radial version of OB0. The root mean squared error between all pairs of pixels between Fig. 13(a) and Fig. 13(b) is 0.13, validating the observable high similarity between the two FAs.

To analyze the computational times using the CPU, the exhaustive PF-based approach needed 3 hours, 39 minutes, and 5 s for 480702 PF simulations. Increasing the number of FSPs increases the number of PF simulations needed largely. For this case study, when adding 1 more FSP, the number of PF simulations needed for the exhaustive PF-based approach is 1300032. This number highlights the challenge the proposed approach addresses. The proposed algorithm needed 42.8s, a speed-up of over 300 times.

2) *Performance Analysis for Flexibility Range*: Table II summarizes the average evaluation metrics of the scenarios on meshed OB0 and OB1 for narrow and wide voltage constraints. The metrics Ac_f and Ac_r are the mean a_f and a_r over the estimated FAs. These results show a great performance of TensorConvolution+ for the FA range. Fig. 14 illustrates an example alignment of each distribution’s samples with the algorithm’s FA estimation, as used to estimate a_f and a_r for each of the 360 FAs. Multiple cases required more than the available RAM in the narrow voltage constraints OB0 estimations for 6, 7 FSPs. In those cases, TensorConvolution+

aggregated at least one pair of FSPs to one FSP for at least one network component to reduce the tensor dimensions.

Fig. 15 shows the computational time to estimate the FA for different FSP numbers. The authors multiplied the number of PF simulations needed for the exhaustive algorithm with the time the GPU spent for 1 PF simulation (0.038s for OB0 and 0.039s for OB1) to estimate the average time the baseline would need to estimate the 360 FAs. The exhaustive PF-based algorithm would be intractable for the majority of the FAs. The average time needed for the TensorConvolution estimations of Fig. 15 varied between 6–36s and the average time over the 360 estimations was 11s. For a high resolution of 10000 pixels and 11 FSPs in the CIGRE network, the proposed algorithm needed 50s to estimate the FA. The maximum number of FSPs impacting a component for the CIGRE network’s FA was 4 due to the radial network structure. All estimations were performed on the GPU.

3) *Comparison With OPF-Based Method*: An OPF-based algorithm was employed to compare with the TensorConvolution+ estimations. The OPF-based algorithm employed weighted-sum multi-objective optimization with objectives $\max(\theta\Delta p^o + (1 - \theta)\Delta q^o)$, $\max(-\theta\Delta p^o + (1 - \theta)\Delta q^o)$, $\max(\theta\Delta p^o - (1 - \theta)\Delta q^o)$, and $\max(-\theta\Delta p^o - (1 - \theta)\Delta q^o)$. For each objective, θ varied between 0–1 with step size as a hyper-parameter. Due to convergence issues, the transforming loading limitations were ignored in the OPF algorithm. The FSP flexibility in this section was considered square where the active and absolute reactive power were between 0 and the nominal power of the FSP.

TensorConvolution+ and the OPF-based algorithm were compared in three cases on the CIGRE network. The cases varied in resolutions and network sensitivity to constraints. First, a case with FSPs the loads 14, 16 and generators 2, 4, 6. The first case resolution was $\delta p = 0.01MW$, $\delta q = 0.02MVAR$ (≈ 690 pixels) and a θ with 0.1 increments for the OPF-based algorithm (44 OPFs). The second case included FSPs the loads 14, 16 and generators 2, 4, 5, 6. The resolution for the second case was $\delta p = 0.02MW$, $\delta q = 0.04MVAR$ (≈ 300 pixels), and a θ of 0.2 increments (24 OPFs). The third case included FSPs the loads 3, 5, 6, 17, and the generator 8. The third case resolution was $\delta p = 0.15MW$, $\delta q = 0.3MVAR$ (≈ 1050 pixels), and a θ with 0.1 increments (44 OPFs). The results of Fig. 16 illustrate the FAs obtained for the three cases. The two algorithms agree on the shape of the FAs. However, the TensorConvolution+ algorithm also explores the inner area feasibility and estimates the DFC. Regarding computational burden, using the GPU for cases 1, 2, 3, TensorConvolution+ required 9s, 5.4s, 23.3s, and the OPF-based algorithm required 37.6s, 21s, 36.3s respectively.

C. Estimating Disconnected Flexibility Areas

This case study showcases the proposed algorithm’s capability to estimate disconnected FAs and to deal with non-linear FSPs. Fig. 17(a) illustrates the proposed algorithm’s result compared to the result from 10000 samples of Fig. 17(b) using the Monte-Carlo-based algorithm with “Hard” distribution. The PF-based algorithm was not capable of effectively

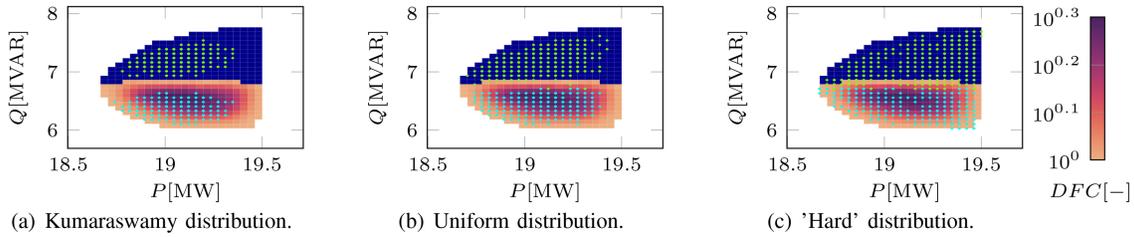


Fig. 14. Monte Carlo-based results from different distributions aligned with the TensorConvolution+ output. Feasible TensorConvolution+ output pixels (■). Not feasible TensorConvolution+ output pixels (■). Feasible Monte Carlo samples (●). Not feasible Monte Carlo samples (●).

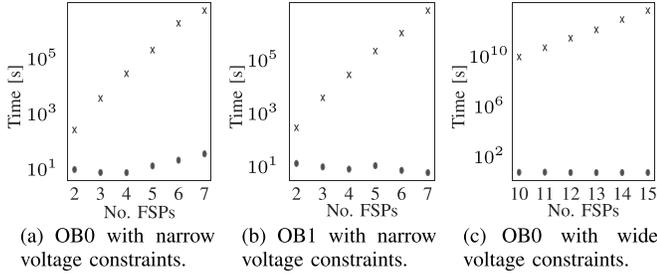


Fig. 15. Computational time of TensorConvolution+ (●) and estimated computational time for exhaustive PF-based approach (X) per FSP amount.

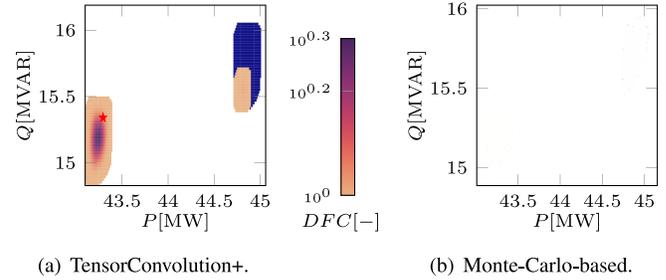


Fig. 17. Disconnected FA Predicted by TensorConvolution+ and the Monte-Carlo-Based algorithms.

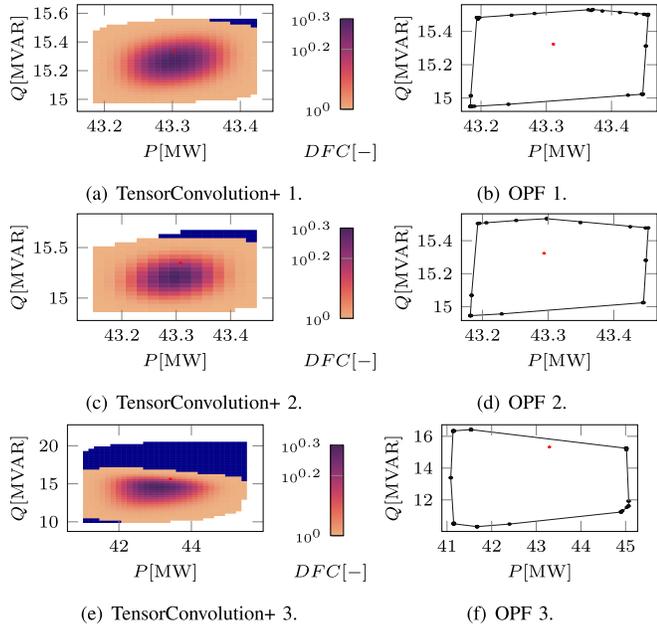


Fig. 16. Comparing FAs from TensorConvolution+ and OPF based algorithm.

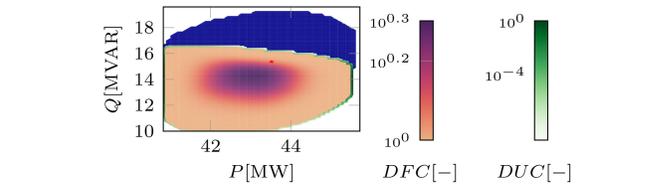


Fig. 18. FA with DFC for feasible combinations (■), density of uncertain combinations (DUC) for feasibility (■), and not feasible FSP shift combinations (■) from initial OP (★).

exploring the disconnected areas, as the exact limits of feasible areas are unclear. The proposed algorithm was capable of estimating the FA, the range, and DFC in 8.5s using the CPU or 7.4s using the GPU.

D. Including Flexibility From Small FSPs

This study shows the proposed algorithm’s capability to include contributions from small FSPs. The increments $\delta p, \delta q$ between FSP shifts determine the FA resolution. In cases with large differences in the flexibility between the larger and

smaller FSPs, reducing the resolution is impractical. Thus, the algorithm initially neglects all FSPs offering flexibility lower than $\delta p, \delta q$ and estimates the FA from the rest. Then, using bi-linear spline interpolation increases the resolution of the estimated FA to add the smaller FSPs. The algorithm convolutes the enhanced FA with the previously neglected FSPs’ shifts. The additional area obtained in the last result constitutes the uncertain FA (not tested for the network constraints). In this study’s results, after neglecting 3 small FSPs, the algorithm increased the pixels of the evaluated area 5 times and aggregated the previously neglected FSPs. Fig. 18 visualizes the resulting FA. The uncertain FA addition process caused 11.09s delays using the GPU, with the rest of the FA estimation process needing 53.2s.

E. Adapting FAs for Partially Observable OCs

This case study showcases the algorithm’s capability to adapt FAs for altered partially known OCs. For the CIGRE network, the algorithm used the tensors computed for Fig. 19(a), and the partial initial OCs of Fig. 19(c) to estimate the FA in Fig. 19(e). The OC change from Fig. 19(a) to Fig. 19(c) increased the CIGRE network’s buses’ sensitivity to the over-voltage constraints. For OBO, the algorithm used the

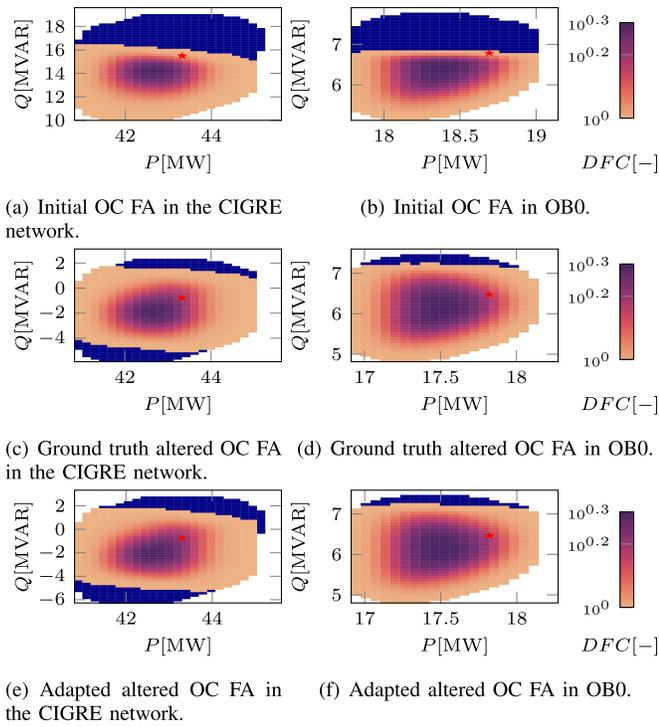


Fig. 19. Adaptability of FAs from initial OCs to altered OCs.

tensors computed for Fig. 19(b), and the partial initial OCs of Fig. 19(d) to estimate the FA in Fig. 19(f). The OC change from Fig. 19(b) to Fig. 19(d) reduced the OB0 component sensitivities to FSP shifts. The adapted areas of Fig. 19(e) and Fig. 19(f) approximate the ground truths of Fig. 19(c) and Fig. 19(d). However, the adapted FAs required the initial OCs from limited components; 13 components in the CIGRE network, and 11 in OB0.

Using the GPU, estimating Fig. 19(a) and Fig. 19(b) and storing the tensors required 286s and 636s. The ground truths of Fig. 19(c) and Fig. 19(d) required 13.2s and 17.6s respectively. The adapted areas of Fig. 19(e) and Fig. 19(f) required 6.8s and 8.3s, respectively. Adapting FAs using the computed values for prior FAs is a capability absent in existing algorithms. The required computing time for TensorConvolution+ can be reduced to approximately half through this capability.

F. Discussion

The proposed algorithm can evaluate the feasibility for the network constraints for all combinations between the FSP shifts. The studies show that this evaluation can benefit system operators in selecting shifts. Shifts with higher DFC offer more feasible flexibility options for system operators. Bigger DFC relates with a higher margin from the network constraints (Fig. 12) and lower costs (Table I). The results show that TensorConvolution+ is the most computationally efficient algorithm for FA estimation with DFC among the algorithms studied. Simulations between the exhaustive PF-based alternative showed that the proposed algorithm could improve the computational speed over 300 times for 2 FSPs.

The speed improvement increases with FSPs (Fig. 15) as the required PFs for the exhaustive approach are (2). Most existing algorithms do not evaluate all flexibility combinations but focus on estimating the FA range. The proposed algorithm has high confidence in estimating the FA range, with an average $A_{cf} = 99\%$. Identifying which FA regions are feasible and which are not is important for system operators to adopt FAs. Existing algorithms can have issues estimating disconnected FAs, causing estimation delays or incorrectly assuming in-between regions as feasible [5]. The proposed algorithm makes use of the impulse response property of convolution (Section III-A) to estimate disconnected areas with low computational burden. The results show that the algorithm can estimate disconnected FAs in 7.4s with distinct feasible regions (Fig. 17). The results demonstrated that the proposed algorithm perform for meshed and radial network topologies, which was a challenge for existing algorithms.

The parameters δp , δq , and c_v , c_l influence the proposed algorithm's performance. Lowering δp , δq , increases the FA resolution and computational burden. For a total offered flexibility P_{tot} , recommended $\delta p = 0.05 \cdot P_{tot}$, $\delta q = 0.1 \cdot Q_{tot}$ result to approximately 400-pixel FA. Lowering c_v , c_l increases the complexity of (7)-(8). Recommended values of $0.001 \geq c_v \geq 0.0001$, $1 \geq c_l \geq 0.1$ are based on the initial OC minimum margins from the constraints. The proposed algorithm's main limitation is the memory usage to store all flexibility combinations in tensors. As FSPs and constraint-sensitive DN components increase, this limitation becomes more notable. Aggregating FSPs for the components requiring unavailable memory can reduce this limitation. Alternatively, using memory-efficient software can mitigate the limitation. Another limitation concerns the need for complete network observability for (non-adapted) FA estimation.

V. CONCLUSION

This paper develops an approach to estimate the flexibility of distribution systems for TSO-DSO coordination. This approach has a near-term practical value for power system operators. This paper introduces applying tensors and convolutions for the flexibility estimation task, utilizing their useful properties. The proposed density of feasible combinations (DFC) improves deciding on operating points that guarantee higher flexibility. The proposed algorithm makes computing and identifying these operating points tractable. The tractability is realized by minimizing the required PF simulations, applying convolutions, and using tensors. Convolutions aggregate flexibility from FSPs, including DFC. Tensors store the impacts of FSP shift combinations to determine the combination feasibility. Moreover, the algorithm applies the Dirac function to represent discrete FSPs. This application allows estimating disconnected FAs with low computational burden.

Case studies on 15-, 70-, and 109-bus systems show the proposed algorithm's performance on meshed and radial networks for connected and disconnected FAs. The results show high numerical confidence in the FA range and DFC. The algorithm estimated the FA around 300 times faster than the alternative approach. The algorithm's average duration of

11s renders the approach promising for further development toward near-real-time TSO-DSO coordination.

Future work can investigate dealing with limited observability in distribution systems. The investigation will explore deep learning-based approaches or techniques from system identification that may advance this flexibility estimation approach. Subsequent work will pursue further improvement of the proposed approach's memory efficiency. Future research will also explore algorithms that can adapt to changing network topologies. Changing network topologies alters the impacts of FSPs on network components. Thus, approximating the impact alterations should allow adapting FAs for changing topologies.

APPENDIX A

PROOF: EXPLORING ALL FEASIBLE COMBINATIONS

The authors use a proof through induction to show that the convolution between any k FSPs will result in the total number of shift combinations between these k FSPs leading to the point $\Delta p^o, \Delta q^o$. Let's assume 2 FSPs, \hat{i} and \tilde{i} . Due to (15), the inner part of (14) can be described as:

$$\begin{aligned} & F_{\hat{i}}(\Delta p, \Delta q) F_{\tilde{i}}(\Delta p^o - \Delta p, \Delta q^o - \Delta q) \\ &= \begin{cases} 1, & \text{if } [\Delta p^o - \Delta p, \Delta q^o - \Delta q]^T \in \Omega_{\hat{i}}^S, [\Delta p, \Delta q]^T \in \Omega_{\tilde{i}}^S, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (29)$$

Therefore, each counted point is from shifts existing in the FSP \hat{i} and \tilde{i} capabilities. From (29), all points of $\Delta p, \Delta q$ that are counted, result in $\Delta p_{\hat{i}} + \Delta p_{\tilde{i}} = \Delta p^o, \Delta q_{\hat{i}} + \Delta q_{\tilde{i}} = \Delta q^o$. Thus, counted shifts only lead to $\Delta p^o, \Delta q^o$. As the summations in convolution are $\sum_{\Delta p=-\infty}^{\infty} \sum_{\Delta q=-\infty}^{\infty}$, all possible combinations of $\Delta p, \Delta q$ are explored. Hence, the convolution counts all available shifts from $\hat{i}, \tilde{i} \in \Omega^{FSP}$, that lead to the OP $p^o + \Delta p^o, q^o + \Delta q^o$. Thus, the result of (29) for all s^o , for $\mathcal{K} = \hat{i}, \tilde{i}$ can be expressed through:

$$\tilde{F}_{\mathcal{K}}(s^o) = \begin{cases} m_{\mathcal{K}}(s^o), & \text{if } s^o \in \Omega_{\mathcal{K}}^S, \\ 0, & \text{otherwise,} \end{cases} \quad (30)$$

$$\Omega_{\mathcal{K}}^S = \left\{ s_{\mathcal{K}} | s_{\mathcal{K}} = \sum_{s \in \pi} s, \forall \pi \in \Omega_{\hat{i}}^S \times \Omega_{\tilde{i}}^S \right\}. \quad (31)$$

$m_{\mathcal{K}}(s^o) \in \mathbf{N}$ is the number of combinations of \mathcal{K} for s^o .

Extending to more FSPs, let the convolution of $\mathcal{K} = 1, \dots, k$ FSPs be described by an extended version of (30), the relationship (16). Then, the convolution of $\tilde{F}_{\mathcal{K}}$ and F_{k+1} is:

$$\begin{aligned} & (\tilde{F}_{\mathcal{K}} * F_{k+1})(\Delta p^o, \Delta q^o) \\ &= \sum_{\Delta p=-\infty}^{\infty} \sum_{\Delta q=-\infty}^{\infty} F_{k+1}(\Delta p, \Delta q) F_{\mathcal{K}}(\Delta p^o - \Delta p, \Delta q^o - \Delta q). \end{aligned} \quad (32)$$

The inner part of (32), expressed through (30) and (15) is:

$$\begin{aligned} & F_{k+1}(\Delta p, \Delta q) F_{\mathcal{K}}(\Delta p^o - \Delta p, \Delta q^o - \Delta q) \\ &= \begin{cases} m_{\mathcal{K}}([\Delta p, \Delta q]^T), & \text{if } [\Delta p^o - \Delta p, \Delta q^o - \Delta q]^T \\ & \in \Omega_{k+1}^S, [\Delta p, \Delta q]^T \in \Omega_{\mathcal{K}}^S, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \quad (33)$$

Each non-zero point is obtained from combinations within the FSP capabilities. From $\sum_{\Delta p=-\infty}^{\infty} \sum_{\Delta q=-\infty}^{\infty}$, all possible combinations from the FSPs are explored. Therefore, if any combination reaching $\Delta p^o, \Delta q^o$ exists, it will be counted. Furthermore, the summation of combinations from a subset of the FSPs ($m_{\mathcal{K}}([\Delta p, \Delta q]^T)$), will ensure that if $\Delta p, \Delta q$ can be used to reach $s^o = [\Delta p^o, \Delta q^o]^T$, then the total number of feasible combinations is accounted. Therefore, the convolution result of $\mathcal{K} + 1 = 1, \dots, k + 1$ FSPs is described by:

$$\begin{aligned} \tilde{F}_{\mathcal{K}+1}(s^o) &= \begin{cases} m_{\mathcal{K}+1}(s^o), & \text{if } s^o \in \Omega_{\mathcal{K}+1}^S, \\ 0, & \text{otherwise,} \end{cases} \quad (34) \\ \Omega_{\mathcal{K}+1}^C &= \Omega_1^S \times \Omega_2^S \times \dots \times \Omega_{k+1}^S, \\ \Omega_{\mathcal{K}+1}^S &= \left\{ s_{\mathcal{K}+1} | s_{\mathcal{K}+1} = \sum_{s \in \pi} s, \forall \pi \in \Omega_{\mathcal{K}+1}^C \right\} \subset \Omega^{S^o}. \end{aligned} \quad (35)$$

The above result is similar to (30), (16). Therefore, the convolution between any number of FSPs will result in all the possible shift combinations for each reachable shift.

REFERENCES

- [1] N. Rodríguez Pérez et al., "ICT architectures for TSO-DSO coordination and data exchange: A European perspective," *IEEE Trans. Smart Grid*, vol. 14, no. 2, pp. 1300–1312, Mar. 2023.
- [2] A. G. Givisiez, K. Petrou, and L. F. Ochoa, "A review on TSO-DSO coordination models and solution techniques," *Electr. Power Syst. Res.*, vol. 189, Dec. 2020, Art. no. 106659.
- [3] M. Kalantar-Neyestanaki, F. Sossan, M. Bozorg, and R. Cherkaoui, "Characterizing the reserve provision capability area of active distribution networks: A linear robust optimization method," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2464–2475, May 2020.
- [4] F. Capitanescu, "TSO-DSO interaction: Active distribution network power chart for TSO ancillary services provision," *Electr. Power Syst. Res.*, vol. 163, pp. 226–230, Oct. 2018.
- [5] J. Silva, J. Sumaili, R. J. Bessa, L. Seca, M. Matos, and V. Miranda, "The challenges of estimating the impact of distributed energy resources flexibility on the TSO/DSO boundary node operating points," *Comput. Oper. Res.*, vol. 96, pp. 294–304, Aug. 2018.
- [6] N. Savvopoulos, C. Y. Evrenosoglu, T. Konstantinou, T. Demiray, and N. Hatzigiorgiou, "Contribution of residential PV and BESS to the operational flexibility at the TSO-DSO interface," in *Proc. Int. Conf. Smart Energy Syst. Technol. (SEST)*, 2021, pp. 1–6.
- [7] D. M. Gonzalez et al., "Determination of the time-dependent flexibility of active distribution networks to control their TSO-DSO interconnection power flow," in *Proc. Power Syst. Comput. Conf. (PSCC)*, 2018, pp. 1–8.
- [8] N. Savvopoulos and N. Hatzigiorgiou, "An effective method to estimate the aggregated flexibility at distribution level," *IEEE Access*, vol. 11, pp. 31373–31383, 2023.
- [9] G. Prionistis, C. Vournas, and M. Vrakopoulou, "A fast method to approximate the flexibility region of an active distribution network in PQ space," in *Proc. IEEE Belgrade PowerTech*, 2023, pp. 1–6.
- [10] A. Churkin, M. Sanchez-Lopez, M. I. Alizadeh, F. Capitanescu, E. A. Martínez Ceseña, and P. Mancarella, "Impacts of distribution network reconfiguration on aggregated DER flexibility," in *Proc. IEEE Belgrade PowerTech*, 2023, pp. 1–7.
- [11] T. Chen, Y. Song, D. J. Hill, and A. Y. S. Lam, "Enhancing flexibility at the transmission-distribution interface with power flow routers," *IEEE Trans. Power Syst.*, vol. 37, no. 4, pp. 2948–2960, Jul. 2022.
- [12] M. Heleno, R. Soares, J. Sumaili, R. J. Bessa, L. Seca, and M. A. Matos, "Estimation of the flexibility range in the transmission-distribution boundary," in *Proc. IEEE Eindhoven PowerTech*, 2015, pp. 1–6.
- [13] J. Silva et al., "Estimating the active and reactive power flexibility area at the TSO-DSO interface," *IEEE Trans. Power Syst.*, vol. 33, no. 5, pp. 4741–4750, Sep. 2018.
- [14] M. Bolfek and T. Capuder, "An analysis of optimal power flow based formulations regarding DSO-TSO flexibility provision," *Int. J. Electr. Power Energy Syst.*, vol. 131, Oct. 2021, Art. no. 106935.
- [15] E. Dall'Anese, S. S. Guggilam, A. Simonetto, Y. C. Chen, and S. V. Dhople, "Optimal regulation of virtual power plants," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1868–1881, Mar. 2018.

- [16] S. Nowak, Y. C. Chen, and L. Wang, "Measurement-based optimal DER dispatch with a recursively estimated sensitivity model," *IEEE Trans. Power Syst.*, vol. 35, no. 6, pp. 4792–4802, Nov. 2020.
- [17] A. Churkin, W. Kong, J. N. Melchor Gutierrez, P. Mancarella, and E. A. Martínez Ceseña, "Assessing distribution network flexibility via reliability-based PQ area segmentation," in *Proc. IEEE Belgrade PowerTech*, 2023, pp. 1–6.
- [18] B. Azimian, R. S. Biswas, S. Moshtagh, A. Pal, L. Tong, and G. Dasarathy, "State and topology estimation for unobservable distribution systems using deep neural networks," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–14, 2022.
- [19] B. Habib, E. Isufi, W. V. Breda, A. Jongepier, and J. L. Cremer, "Deep statistical solver for distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 39, no. 2, pp. 4039–4050, Mar. 2024.
- [20] A. Primadianto and C.-N. Lu, "A review on distribution system state estimation," *IEEE Trans. Power Syst.*, vol. 32, no. 5, pp. 3875–3883, Sep. 2017.
- [21] D. Chrysostomou, J. L. R. Torres, and J. L. Cremer, "Exploring operational flexibility of active distribution networks with low observability," in *Proc. IEEE Belgrade PowerTech*, 2023, pp. 1–6.
- [22] W. Dai, J. Jian, and S. Guo, "Fast characterization of nonlinear feasible region based on deep neural network association mining," *Energy Rep.*, vol. 9, pp. 1102–1111, Sep. 2023.
- [23] X. Pan, M. Chen, T. Zhao, and S. H. Low, "DeepOPF: A feasibility-optimized deep neural network approach for AC optimal power flow problems," *IEEE Syst. J.*, vol. 17, no. 1, pp. 673–683, Mar. 2023.
- [24] X. Lei, Z. Yang, J. Yu, J. Zhao, Q. Gao, and H. Yu, "Data-driven optimal power flow: A physics-informed machine learning approach," *IEEE Trans. Power Syst.*, vol. 36, no. 1, pp. 346–354, Jan. 2021.
- [25] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Trans. Power Syst.*, vol. 36, no. 3, pp. 2082–2092, May 2021.
- [26] A. Yaniv, P. Kumar, and Y. Beck, "Towards adoption of GNNs for power flow applications in distribution systems," *Electr. Power Syst. Res.*, vol. 216, Mar. 2023, Art. no. 109005.
- [27] S. Riaz and P. Mancarella, "Modelling and characterisation of flexibility from distributed energy resources," *IEEE Trans. Power Syst.*, vol. 37, no. 1, pp. 38–50, Jan. 2022.
- [28] S. W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. San Diego, CA, USA: California Tech. Publ., 1997.
- [29] S. Poudel, Z. Ni, and W. Sun, "Electrical distance approach for searching vulnerable branches during contingencies," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3373–3382, Jul. 2018.
- [30] J. Klaus, M. Blacher, and J. Giesen, "Compiling tensor expressions into Einsum," in *Proc. Int. Conf. Comput. Sci.*, 2023, pp. 129–136.
- [31] S. Repo et al., "Holistic view of needs and requirements of multi-market flexibility trading," in *Proc. CIRED Berlin Workshop (CIRED)*, 2020, pp. 807–810.
- [32] I. V. Oseledets, "Tensor-train decomposition," *SIAM J. Sci. Comput.*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [33] E. Bisong and E. Bisong, "Google colabatory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*. Berkeley, CA, USA: Apress, 2019, pp. 59–64.



José Luis Rueda Torres (Senior Member, IEEE) was born in 1980. He received the Diploma degree (cum laude) in electrical engineering from the Escuela Politécnica Nacional, Quito, Ecuador, in August 2004, and the Ph.D. degree (Hons.) in electrical engineering from the National University of San Juan, in November 2009. He is currently an Associate Professor leading the Research Team on Dynamic Stability of Sustainable Electrical Power Systems, Intelligent Electrical Power Grids Section, Electrical Sustainable Energy Department, Delft University of Technology, Delft, The Netherlands. From September 2003 to February 2005, he worked in Ecuador, in the fields of industrial control systems and electrical distribution networks operation and planning. From August 2010 to February 2014, he was a Postdoctoral Research Associate with the Institute of Electrical Power Systems, University Duisburg-Essen, Duisburg, Germany. His research interests include physics-driven analysis of stability phenomena dynamic equivalencing of HVDC-HVAC systems, probabilistic multi-systemic reliability and stability management, and adaptive-optimal resilient multi-objective controller design. He is currently a member of the Technical Committee on Power and Energy Systems of International Federation of Automatic Control, the Past-Chairman of IEEE POWER AND ENERGY SOCIETY Working Group on Modern Heuristic Optimization, the Secretary of CIGRE JWG C4/C2.58/IEEE "Evaluation of Voltage Stability Assessment Methodologies in Transmission Systems," the Vice-Chair of the IEEE PES Intelligent Systems Subcommittee, and the Vice-Chair of the IFAC Technical Committee TC 6.3. Power and Energy Systems on Social Media.



Demetris Chrysostomou (Graduate Student Member, IEEE) received the B.Sc. degree in electrical engineering from the University of Cyprus, Nicosia, Cyprus, in 2018, and the M.Sc. degree in robotics, systems and control from ETH Zürich, Zürich, Switzerland, in 2021. He is currently pursuing the Ph.D. degree with the Technische Universiteit Delft, Delft, The Netherlands. He previously worked as a Research Assistant with the PSM Laboratory, University of Cyprus and as a Research and Development Intern with ABB

Corporate Research. He is a member of the AI Energy Lab and the IEPG Group, EEMCS Department of TU Delft. His research is part of the MEGAMIND Program. His research interests include machine learning, control, and optimization in power systems operation.



Jochen Lorenz Cremer (Member, IEEE) received the Ph.D. degree from Imperial College London, U.K., in 2020. He works as an Assistant Professor with the Delft AI Energy Lab, Faculty of Electrical Engineering, Mathematics, and Computer Science, Delft University of Technology, The Netherlands. His research interests include machine learning and mathematical programming applied to the operation and planning of power systems.