A parallelization algorithm for the Capacity Expansion Problem of an Electrical Grid

Gabor Riccardi[1]

[1]Università degli Studi di Pavia [2]Joint Research Centre (JRC)

Introduction

The European electricity system is undergoing significant changes motivated by the EU's ambition to achieve climate neutrality.

The European Resource Adequacy Assessment, provides an instrument for detecting and measuring adequacy concerns, which are becoming the basis for the implementation of capacity mechanism

Given the size of the problem, only reduced Stochastic Models have been considered, using only 3 uncertainty realizations.

Consequently there is an institutional urge to develop a reliable and robust ERAA study at a pan-European level.

Goal:

Develop a parallelization algorithm for the Capacity Expansion Problem (CEP) used for the Adequacy Assessment.

Economic Dispatch (ED)

The Economic Dispatch calculates the operation cost of an electrical grid given a scenario $\omega = (\mathcal{PV}, \mathcal{W}, \mathcal{D})$ comprising of respectively solar power, wind power and loads.

$$\min_y q' y_\omega \tag{1}$$

$s.t.$ (Power Flow Conservation)

$$p_{n,g,t,\omega} + bd_{n,t,\omega} + \sum_{l \in \mathcal{L}(n)} f_{n,l,t,\omega} + ls_{n,t,\omega} + \mathcal{PV}_{n,t,\omega} + \mathcal{W}_{n,t,\omega} = \tag{2}$$

$$= \mathcal{D}_{n,t,\omega} + s_{nt.\omega} + bc_{n,t,\omega}$$

(Storage Dynamics)

$$v_{n,t,\omega} = v_{n,t-1,\omega} + BCE \cdot bc_{n,t,\omega} - BDE \cdot bd_{n,t,\omega} + A_{n,t,\omega} \tag{3}$$

(Storage and Balancing Limits)

$$(v_{n,t,\omega}, bc_{n,t,\omega}, bd_{n,t,\omega}) \leq (BV, BC, BD) \tag{4}$$

(Generation Capacity)

$$p_{n,g,t,\omega} \leq p^{\text{max}}_{n,g} + x_{n,g} \tag{5}$$

(Transmission Capacity)

$$L^{\text{min}}_{n,l} \leq f_{n,l,t,\omega} \leq L^{\text{max}}_{n,l} \tag{6}$$

(7)

Where $v_{n,t,w}$ is the power stored at bus n at time t and $y_\omega = (p_\omega, f_\omega, ls_\omega, s)'$ is the vector containing the power generation, power flows, line shedding and spillage variables.

Capacity Expansion Problem (CEP)

CEP is a two stage stochastic program in which first stage determines the capacity expansion $x_{n,g}$ for each generator $g \in \mathcal{G}$

The second stage solves the Economic Dispatch.

$$\min_x c'x + \mathbb{E}_\omega [\mathcal{V}(x, \omega)]$$

$$s.t.\ 0 \leq x_{n,g} \leq X_{n,g} \quad \text{(Generation Capacity Expansion Limits)} \tag{CEP}$$

Where $\mathcal{V}(\mathbf{x}, \omega)$ is the solution to (ED) in function of the expanded capacities x and the scenario ω.

Example on small AC-DC grid

We implemented the algorithm on the following network, consisting of both AC and DC lines, different kinds of storage units, solar, gas and wind power for a time horizon of 5 weeks and time steps of one hour.

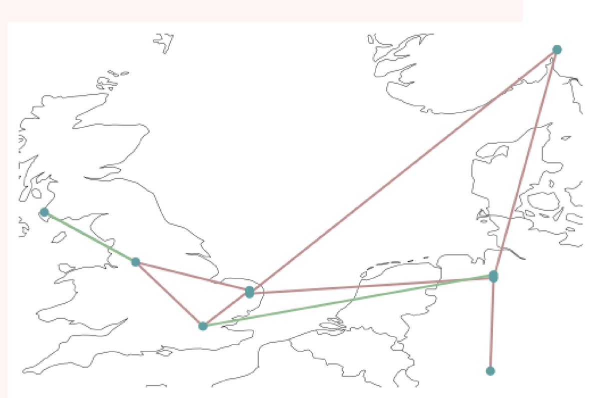


Figure 3. Network layout

In this instance the algorithm convergence to the optimal solutions in 12 iterations:

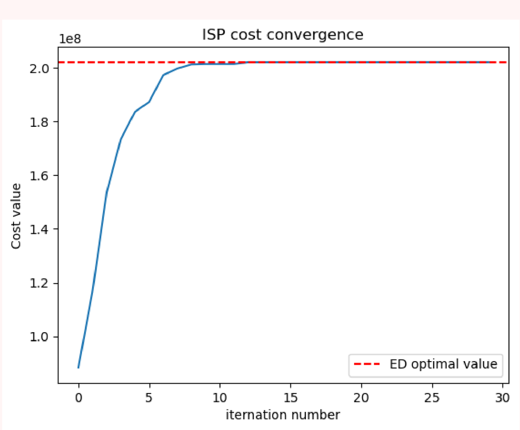


Figure 4. Objective value of (ISP) for each iteration.

Idea

We try to leverage the fact the the variables of different timesteps have few "interconnections". We can formalize the notion of "connection" between variables with the following definition.

Definition: The *hypergraph* associated to a linear programming problem LP, denoted by $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, is constructed as follows:

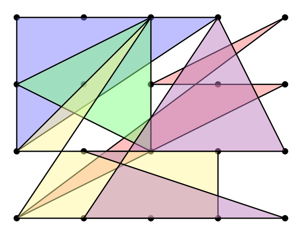


Figure 1. Example of LP hypergraph

The *nodes* $\mathcal{N}$ of $\mathcal{G}$ correspond to the variables of the LP.

The *hyperedges* $\mathcal{E}$ of $\mathcal{G}$ correspond to each set of variables that appears together in any constraint of the LP.

If a hypergraph features a partition of $\mathcal{N}$ with sparse interconnections (few edges) between subsets, we can remove these edges to solve each subset independently. To do this, we fix the variables in the removed edges based on the solutions from an iteratively tightened linear program, aiming to determine their optimal values. For instance, in the case of the Energy Dispatch (ED) problem, we proceed as follows:

We divide the time horizon into K intervals, $\{0, \dots, t_1\}, \dots, \{t_{K-1} + 1, \dots, t_K\}$ We fix a priori the intermediate storage values v_{t_k} for $k = 1, \dots, K$. We refer to the (ED) problems restricted to each time interval with fixed initial and final storage values as **(ED-k)** with optimal value $\mathcal{V}_\mathbf{k}(\mathbf{x}, \mathbf{v}_{\mathbf{t_k}}, \mathbf{v}_{\mathbf{t_{k+1}}}, \omega)$

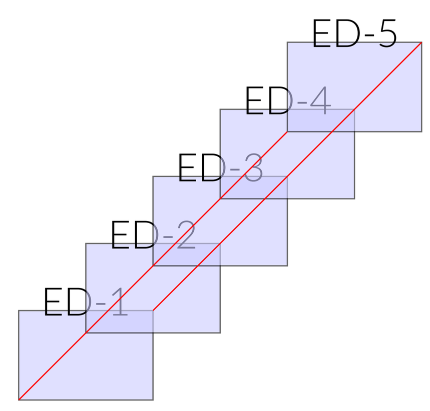


Figure 2. (ED) hypergraph representation

Preliminary Definitions

Oss 1: $\mathcal{V}(x, \omega) = \min_{\{v_{t_k}\}_{k=1}^K} \sum_{k=0}^{K-1} \mathcal{V}_k(x, v_{t_k}, v_{t_k+1}, \omega)$

Since each function $\mathcal{V}_k$ is piecewise linear convex in $x, v_{t_K}, v_{t_{K+1}}$, it can be approximated by a collection of supporting hyperplanes $\{\pi^w_{i,k}(x, v_{t_k}, v_{t_{k+1}})\}$ of each $\mathcal{V}_k$.

An approximation of (ED) is given by:

$$\begin{aligned} \hat{\mathcal{V}}(x, \omega) &= \min_{\{v_{t_k}\}_{k=1}^K} \sum_{k=0}^{K} \hat{\mathcal{V}}_k(x, v_{t_k}, v_{t_{k+1}}) = \\ &= \min_{\{v_{t_k}\}_{k=1}^K} \sum_{k=0}^{K} \theta_k^\omega \\ &\quad \text{s.t.} \quad \theta_k^\omega \geq \pi^\omega_{i,k}(x, v_{t_k}, v_{t_{k+1}}) \quad \forall i, k \end{aligned} \tag{ISP}$$

We refer to this problem as the **Intermediate Storage Problem (ISP)** (I know, very original)

By substituting $\mathcal{V}(x, \omega)$ with $\hat{\mathcal{V}}(x, \omega)$ in the definition of (CEP) we obtain the following relaxation:

$$\min_x c'x + \mathbb{E}_w \left[\hat{\mathcal{V}}(x, w)\right]$$

$$s.t.\ 0 \leq x_{n,g} \leq X_{n,g} \tag{CEP-R}$$

After every iteration, the ISP is tightened by adding cuts computed from the duals of the storage constraints of (ED-k) for $k = 1 \dots K$. After a finite number of iterations ISP becomes exact.

Future directions

It is not yet clear how the number of iterations to get convergence scales up with the size of the network and how to choose the optimal value for K.

Implement the algorithm in the open source python package Pypsa

Extend algorithm to general LP: Given the hypergraph $(\mathcal{N}, \mathcal{E})$ associated to LP, we seek a partition of $\mathcal{N}$ into disjoint subsets, represented as $\mathcal{N} = \sqcup_{i=1}^n \mathcal{N}_i$. Where any edge connecting $\mathcal{N}_i, \mathcal{N}_j$ with $i \neq j$ is not connected to any other $\mathcal{N}_k$ and has exactly one node in common with one of the two subsets involved.

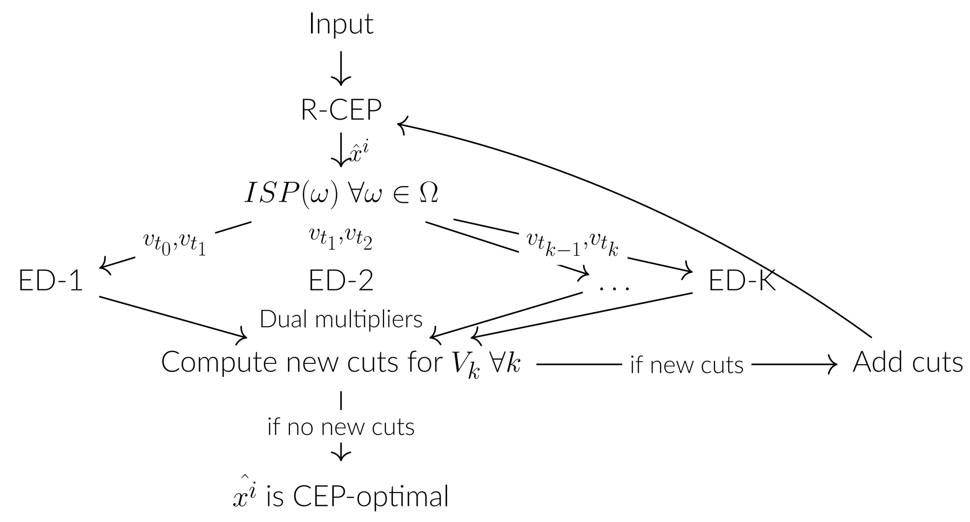
Then we could apply the same procedure used for parallelizing ED, splitting the LP in the various $\mathcal{N}_i$ and iteratively construct an LP model to get the optimal values of the shared variables.

The following questions need to be addressed:

How do we guarantee that (MP) gives solutions which make the LP restricted to each $\mathcal{N}_i$ have a feasible solution? → How to add feasibility cuts

Is there a way to tell a priori when applying this method is quicker than not parallelizing? → Consider the size of $\mathcal{N}_i$ and number of interconnections between the partition.

Algorithm's Flowchart



Convergence Results

Since $(CEP - R) \leq (CEP)$ if a $(CEP - R)$ optimal solution has the same cost for (CEP) then it's also (CEP)-optimal.

Oss 2: It is sufficient to prove that after a finite number of steps (i) of the algorithm we have:

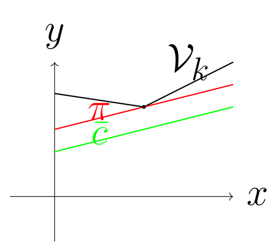
$$\hat{\mathcal{V}}(\hat{x}^i, \omega) = \mathcal{V}(\hat{x}^i, \omega) \text{ for all } \omega \in \Omega \tag{8}$$

Oss 3: After a finite number of iterations no new cuts are found for $\mathcal{V}_k$.

Proof.

We observe that the number possible normal vectors defining the cuts are finite because they correspond to dual solutions of (ED-k) and thus are less than the number of basis matrices of (ED-k), which do not depend on the intermediate storage values .

Thus after a finite number of steps we have:
a new cut:
$\bar{c}(x, v) = p'(x, v) + b$
and an old cut:
$\pi(x, v) = p'(x, v) + \bar{b}$
Having the same normal vector p.



Since both are supporting hyperplanes it follows that $b = \bar{b}$ (and therefore $\bar{c}$ is not a new cut). □

Since supporting hyperplanes calculated at $\hat{x}_i$ match the value of $\mathcal{V}(\hat{x}_i, \omega)$, if a new cut is redundant then the lower approximation $\hat{\mathcal{V}}$ was already exact at $\hat{x}_i$ thus:

Oss 4: If after the i-iteration no new cuts are added for some i and k then $\hat{\mathcal{V}}_k(\hat{x}^i, \hat{v}_k, \hat{v}_{k+1}) = \mathcal{V}_k(\hat{x}^i, \hat{v}_k, \hat{v}_{k+1})$.

Putting together the previous observations we obtain finally:

Proposition

The algorithm converges after a finite number of iterations and $\hat{x}^i$ is an optimal solution for (CEP).

References

[1] Daniel Ávila, Anthony Papavasiliou, Mauricio Junca, and Lazaros Exizidis. "Applying High- Performance Computing to the European Resource Adequacy Assessment". In: IEEE Trans- actions on Power Systems (2023), pp. 1–13.

[2] Daniel Bienstock, Mauro Escobar, Claudio Gentile, and Leo Liberti. "Mathematical Pro- gramming formulations for the Alternating Current Optimal Power Flow problem". In: 4OR 18.3 (July 2020), pp. 249–292. doi: 10.1007/s10288-020-00455-w.

[3] T. Brown, J. Hörsch, and D. Schlachtberger. "PyPSA: Python for Power System Analysis". In: Journal of Open Research Software 6.4 (1 2018).

[4] Fabian Hofmann. "Linopy: Linear optimization with n-dimensional labeled variables". In: Journal of Open Source Software 8.84 (2023), p. 4823.