# Decentralized Algorithms for Generalized Nash Equilibrium Seeking in Peer-to-peer Electricity Markets

Giuseppe Belgioioso

November 11, 2020

### 1 Problem statement

Consider the same problem statement in problem\_setup\_v7.tex.

### 2 Algorithms

#### 2.1 Semi-decentralized

The algorithm presented in this section corresponds to [1, Alg. 6B] applied on problem\_setup\_v6.tex. Consider the following choices for the step-sizes in Algorithm 1.

Assumption 1 (Step-size) Set the step sizes of prosumers and DSO as follows:

(i) 
$$\forall i \in \mathcal{N}$$
: set  $A_i = \operatorname{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j \in \mathcal{N}_i}) \otimes I_H$ , with  $\alpha_i^{pi}, \alpha_i^{st} > 1$ ,  $\alpha_i^{mg} > 3 + N \max_{h \in \mathcal{H}} d_h^{mg}$ ,  $\alpha_{(i,j)}^{tr} > 2$ ,  $\forall j \in \mathcal{N}_i$ ,  $\beta_{(i,j)}^{tr} = \beta_{(j,i)}^{tr} < \frac{1}{2}$ ,  $\forall j \in \mathcal{N}_i$ .

(ii) Set 
$$A_{N+1} := \operatorname{diag}\left(\left\{\alpha_y^{\theta}, \alpha_y^{v}, \alpha_y^{tg}, \left\{\alpha_{(y,z)}^{p}, \alpha_{(y,z)}^{q}\right\}_{z \in \mathcal{B}_y}\right\}_{y \in \mathcal{B}}\right) \otimes I_H$$
, with  $\alpha_y^{\theta}, \alpha_y^{v} > 0$ ,  $\alpha_y^{tg} > 2$ ,  $\alpha_{(y,z)}^{p} > 1$  and  $\alpha_{(y,z)}^{q} > 0$ ,  $\forall z \in \mathcal{B}_y$ ,  $\forall y \in \mathcal{B}$ . Set  $\gamma^{mg} < \frac{1}{N}$ ,  $\beta^{tg} < (|\mathcal{N}| + |\mathcal{B}|)^{-1}$  and  $\beta_y^{pb} < (1 + 2|\mathcal{N}_y| + |\mathcal{B}_y|)^{-1}$ , for all  $y \in \mathcal{B}$ .

### Algorithm 1 Semi-decentralized GWE seeking for P2P Energy Markets

```
1: Iterate until convergence
  2:
                for all prosumer i \in \mathcal{N} do
                         primal update
                                                                                                                                    ▷ power generated, stored, from the grid, traded
  3:
                               a_{i}(k) = \operatorname{col}\left(-\mu_{y}^{\operatorname{pb}}(k), -\mu_{y}^{\operatorname{pb}}(k), \begin{bmatrix} I_{H} \\ -I_{H} \end{bmatrix}^{\top} \lambda^{\operatorname{mg}}(k) + \mu^{\operatorname{tg}}(k), \left\{\mu_{(i,j)}^{\operatorname{tr}}(k)\right\}_{j \in \mathcal{N}_{i}}\right) \quad \triangleright \text{ aux. vector}
u_{i}(k+1) = \begin{cases} \operatorname{argmin}_{\xi \in \mathbb{R}^{n_{i}}} & J_{i}(\xi, \sigma^{\operatorname{mg}}(k)) + a_{i}(k)^{\top} \xi + \frac{1}{2} \left\|\xi - u_{i}(k)\right\|_{A_{i}}^{2} \\ \text{s.t.} \quad \xi \in \mathcal{U}_{i} \end{cases} \quad \triangleright \text{ quadratic progr.}
  4:
  5:
  6:
                         end
                         communication
                                                                                                                                                                             ▶ to DSO and trading partners
  7:
                                b_i(k+1) = p_i^{\rm d} - p_i^{\rm di}(k+1) - p_i^{\rm st}(k+1)

p_i^{\rm mg}(k+1), b_i(k+1) \longrightarrow {\rm DSO},
                                                                                                                                                                 \triangleright local load unbalance of prosumer i
  8:
                                                                                                                                                                                                            ⊳ forward to DSO
  9:
                                for all prosumer j \in \mathcal{N}_i do
                                p^{\mathrm{tr}}_{(i,j)}(k+1) \longrightarrow \text{prosumer } j end for
10:
                                                                                                                                                                  \triangleright forward local trade to prosumer j
11:
12:
13:
                         end
                         dual update
                                                                                                                                                                                              ▷ reciprocity constraints
14:
                                 for all j \in \mathcal{N}_i do
15:
                                         c_{(i,j)}^{\text{tr}}(k+1) = p_{(i,j)}^{\text{tr}}(k+1) + p_{(i,i)}^{\text{tr}}(k+1)
                                                                                                                                                                                                                        ▷ aux. vector
16:
                                        \mu^{\text{tr}}_{(i,j)}(k+1) = \mu_{(i,j)}(k) + \beta^{\text{tr}}_{ij} \left( 2c^{\text{tr}}_{(i,j)}(k+1) - c^{\text{tr}}_{(i,j)}(k) \right)
                                                                                                                                                                                                  ▷ reflected dual ascent
17:
18:
                         end
19:
                end for
20:
21:
                DSO update
22:
                         primal update
                                                                                                                                                                                                         ▶ physical variables
                                a_{N+1}(k) = \operatorname{col}\left(\left\{\mathbf{0}, \mathbf{0}, -\mu^{\operatorname{tg}}(k) - \mu^{\operatorname{pb}}_{y}(k), \left\{-\mu^{\operatorname{pb}}_{y}(k), \mathbf{0}\right\}_{z \in \mathcal{B}_{y}}\right\}_{y \in \mathcal{B}}\right)
u_{N+1}(k+1) = \operatorname{proj}_{\mathcal{U}_{N+1}}\left(u_{N+1}(k) - A_{N+1}a_{N+1}(k)\right)

    aux. vector

23:
                                                                                                                                                                                           ⊳ solved via Algorithm 2
24:
25:
                         aggregation update
26:
                                \sigma^{\text{mg}}(k+1) = \sum_{i \in \mathcal{N}} p_i^{\text{mg}}(k+1)\sigma^{\text{tg}}(k+1) = \sum_{y \in \mathcal{B}} p_y^{\text{tg}}(k+1)

    ▷ aggregate grid-to-prosumers power

27:

    ▷ aggregate grid-to-buses power

28:
29:
                         dual update
30:
                                b_{N+1}(k+1) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes (2\sigma^{\mathrm{mg}}(k+1) - \sigma^{\mathrm{mg}}(k)) - \begin{bmatrix} \overline{p}^{\mathrm{mg}} \mathbf{1}_H \\ -\underline{p}^{\mathrm{mg}} \mathbf{1}_H \end{bmatrix}\lambda^{\mathrm{mg}}(k+1) = \mathrm{proj}_{\mathbb{R}^{2H}}(\lambda^{\mathrm{mg}}(k) + \gamma^{\mathrm{mg}} b_{N+1}(k+1))
31:
                                                                                                                                                                                                                        ▷ aux. vector
                                                                                                                                                                                                              ▶ grid constraints
32:
                                 for all buses y \in \mathcal{B}^{\mathsf{T}}do
33:
                                        c_y^{\mathrm{pb}}(k+1) = p_y^{\mathrm{pd}} + \sum_{i \in \mathcal{N}_y} b_i(k+1) - p_y^{\mathrm{tg}}(k+1) - \sum_{z \in \mathcal{B}_y} p_{(y,z)}^{\ell}(k+1) \quad \triangleright \text{ aux. vector } \mu_y^{\mathrm{pb}}(k+1) = \mu_y^{\mathrm{pb}}(k) + \beta_y^{\mathrm{pb}}(2c_y^{\mathrm{pb}}(k+1) - c_y^{\mathrm{pb}}(k)) \quad \triangleright \text{ local power balance of bus } y
34:
35:
                                 end for
36:
                                \begin{split} c^{\text{tg}}(k+1) &= \sigma^{\text{mg}}(k+1) - \sigma^{\text{tg}}(k+1) \\ \mu^{\text{tg}}(k+1) &= \mu^{\text{tg}}(k) + \beta^{\text{tg}}(2c^{\text{tg}}(k+1) - c^{\text{tg}}(k)) \end{split}
37:

    b aux. vector

                                                                                                                                                                                        ▷ grid-to-buses constraints
38:
39:
40:
                         communication
                                                                                                                                                                                                                            ▶ broadcast
                                 \{\sigma(k+1), \lambda^{\mathrm{mg}}(k+1), \mu^{\mathrm{tg}}(k+1)\} \longrightarrow \mathcal{N}
                                                                                                                                                                                                            ▶ to all prosumers
41:
                                 for all buses y \in \mathcal{B} do
42:
                                         \mu_y^{\mathrm{pb}}(k+1) \longrightarrow \mathcal{N}_y
                                                                                                                                                                                      \triangleright to all prosumers on bus y
43:
                                 end for
44:
                         end
45:
                end
46:
47: end
```

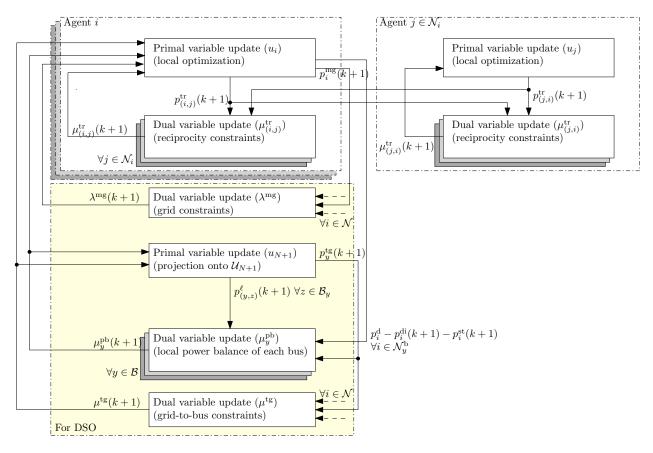


Figure 1: Flowchart of the iterations in Algorithm 1 for agent  $i \in \mathcal{N}$  and DSO. It also shows the flow of information between agent i and DSO as well as between agent i and its trading partner j.

### 2.2 Projection onto $\mathcal{U}_{N+1}$

Next, we propose an iterative method to solve line 24 in Algorithm 1, namely, to compute the projection onto  $\mathcal{U}_{N+1}$ . First, let us define the sets  $C_1 := (17) \cap (18) \cap (19) \cap (20)$  and  $C_2 = (15) \cap (16)$  and recall that the decision vector of the DSO reads as  $u_{N+1} = \operatorname{col}\left(\{\theta_y, v_y, p_y^{\operatorname{tg}}, \{p_{(y,z)}^\ell, q_{(y,z)}^\ell\}_{z \in \mathcal{B}_y}\}_{y \in \mathcal{B}}\right)$ . The projection onto  $C_1$  can be characterize in closed-form as follows:

$$\mathrm{proj}_{C_1}(u_{N+1}) = \mathrm{col}\left(\{\theta_y^+, v_y^+, {p_y^{\mathrm{tg}}}^+, \{{p_{(y,z)}^\ell}^+, {q_{(y,z)}^\ell}^+\}_{z \in \mathcal{B}_y}\}_{y \in \mathcal{B}}\right),\,$$

where, for all  $y \in \mathcal{B}$ ,

$$\theta_y^+ = \begin{cases} \frac{\theta_y}{\overline{\theta}_y}, & \text{if } \theta_y < \underline{\theta}_y \\ \overline{\theta}_y, & \text{otherwise} \end{cases},$$

$$v_y^+ = \begin{cases} \frac{v}{\overline{y}}, & \text{if } v_y < \underline{v}_y \\ \overline{v}_y, & \text{if } v_y > \overline{v}_y \\ v_y, & \text{otherwise} \end{cases}$$

$$p_y^{\text{tg}^+} = \begin{cases} p_y^{\text{tg}}, & \text{if } y \in \mathcal{B}^{\text{mg}} \\ 0, & \text{otherwise} \end{cases}$$

$$(p_{(y,z),h}^{\ell})^+ = \frac{\overline{s}_{(y,z)}}{\max\left\{\|\operatorname{col}(p_{(y,z),h}^{\ell}, q_{(y,z),h}^{\ell})\|, \overline{s}_{(y,z)}\right\}} p_{(y,z),h}^{\ell}, \qquad \forall z \in \mathcal{B}_y, \forall h \in \mathcal{H}$$

$$(q_{(y,z),h}^{\ell})^+ = \frac{\overline{s}_{(y,z)}}{\max\left\{\|\operatorname{col}(p_{(y,z),h}^{\ell}, q_{(y,z),h}^{\ell})\|, \overline{s}_{(y,z)}\right\}} q_{(y,z),h}^{\ell}, \qquad \forall z \in \mathcal{B}_y, \forall h \in \mathcal{H}$$

The projection onto  $C_2$  can be computed by solving a quadratic programming (e.g. via lsqlin, quadprog, osqp, etc...) with appropriate matrices.

# **Algorithm 2** Douglas–Rachford splitting to compute the projection of x onto $\mathcal{U}_{N+1} = C_1 \cap C_2$

- 1: Iterate until convergence
- 2:  $z(k) = \operatorname{proj}_{C_1}(\frac{1}{2}\xi(k) + \frac{1}{2}x)$
- 3:  $\xi(k+1) = \xi(k) + \lambda \left( \operatorname{proj}_{C_2}(2z(k) \xi(k)) z(k) \right)$ , with  $\lambda \in (0,2)$
- 4: **end**

### 2.3 Fully-distributed

Consider the following choices for the step-sizes.

Assumption 2 (Step size selection) Set the step sizes of prosumers and DSO as follows:

- (i)  $\forall i \in \mathcal{N}$ : set  $A_i = \operatorname{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j \in \mathcal{N}_i}) \otimes I_H$ , with  $\alpha_i^{pi}, \alpha_i^{st} > 1$ ,  $\alpha_i^{mg} > 3 + N \max_{h \in \mathcal{H}} d_h^{mg}$ ,  $\alpha_{(i,j)}^{tr} > 2$ ,  $\forall j \in \mathcal{N}_i$ ,  $\beta_{(i,j)}^{tr} = \beta_{(j,i)}^{tr} < \frac{1}{2}$ ,  $\forall j \in \mathcal{N}_i$ .
- (ii)  $\forall y \in \mathcal{B}$ : set  $A_y := \operatorname{diag}\left(\alpha_y^{\theta}, \alpha_y^{v}, \alpha_y^{tg}, \{\alpha_{(y,z)}^{p}, \alpha_{(y,z)}^{q}\}_{z \in \mathcal{B}_y}\right) \otimes I_H$ , with  $\alpha_y^{\theta}, \alpha_y^{v} > 2|\mathcal{B}_y|(\|B\| + \|G\|)$ ,  $\alpha_y^{tg} > 2$ ,  $\alpha_{(y,z)}^{p} > 2$  and  $\alpha_{(y,z)}^{q} > 1$ ,  $\forall z \in \mathcal{B}_y$ ,  $\forall y \in \mathcal{B}$ . Set  $\delta_y^{mg} = \delta^{mg} < \frac{1}{2}$ ,  $\delta_y^{tg} = \delta^{tg} < \frac{1}{2}$ . Set  $\gamma_y^{mg} < (|\mathcal{N}_y| + |\mathcal{B}_y|)^{-1}$ . Set  $\beta_y^{tg} < (2|\mathcal{N}_y| + |\mathcal{B}_y|)^{-1}$  and  $\beta_y^{pb} < (1 + 2|\mathcal{N}_y| + |\mathcal{B}_y|)^{-1}$ . Set  $\beta_{(z,y)}^{p,c} = \beta_{(y,z)}^{p} < (2|\mathcal{B}_{(y,z)}\| + 2|\mathcal{G}_{(y,z)}\| + 2)^{-1}$ ,  $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q} < (2|\mathcal{B}_{(y,z)}\| + 2|\mathcal{G}_{(y,z)}\| + 2)^{-1}$ ,  $\forall z \in \mathcal{B}_y$ . Set  $\beta_{(z,y)}^{p,c} = \beta_{(y,z)}^{p,c} < \frac{1}{2}$ ,  $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q,c} < \frac{1}{2}$ ,  $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q,c} < \frac{1}{2}$ ,  $\forall z \in \mathcal{B}_y$ .
  - the decision variable of bus y is  $u_y := \left(\theta_y, v_y^{\text{v}}, p_y^{\text{tg}}, \{p_{(y,z)}^{\ell}, q_{(y,z)}^{\ell}\}_{z \in \mathcal{B}_y}\right) \in \mathbb{R}^d$ , with  $d := (3 + 2|\mathcal{B}_y|)24$ .
  - Let us introduce the local sets of constraints of bus y, i.e.,  $\mathcal{U}_y := \{u \in \mathbb{R}^d \mid (17) (20) \text{ are satisfied}\}$ .
  - We recast the power flow equation (15) as

$$0 = 2B(\theta_y - \theta_z) - 2G(v_y - v_z) - (p_{(y,z)}^{\ell} - p_{(z,y)}^{\ell})$$
  
$$0 = p_{(y,z)}^{\ell} + p_{(z,y)}^{\ell}$$

• Similarly, we recast the power flow equation (16) as

$$0 = 2G(\theta_y - \theta_z) + 2B(v_y - v_z) - (q_{(y,z)}^{\ell} - q_{(z,y)}^{\ell})$$
  
$$0 = q_{(y,z)}^{\ell} + q_{(z,y)}^{\ell}$$

GB: It is convenient for deriving an effective distributed algorithm. Intuitively, the dual variables of such constraints will evolve (almost) identically on bus y and z.

#### Algorithm 3 Fully-decentralized GWE seeking for P2P Energy Markets

```
1: Iterate until convergence
       for all Bus y \in \mathcal{B} (in parallel) do
2:
           for all prosumer i \in \mathcal{N}_y (in parallel) do
3:
               Prosumer i update
4:
                   primal update
                                                             ▷ power generated, stored, from the grid, traded
5:
6:
                   communication
                                                           \triangleright with bus y operator and trading partners i \in \mathcal{N}_i
                   dual update
                                                                                        ▷ reciprocity constraints
7:
               end prosumer i update
8:
           end for
9:
10:
           DSO update (local bus y unit)
               primal update
                                                                             \triangleright local physical variables of bus y
11:
12:
               aggregation update
                                                   \triangleright total grid-to-pros. power and load unbalance on bus y
               auxiliary update
                                                                          ▷ for consensus of the dual variables
13:
               dual update
                                                                                           ▶ physical constraints
14:
                                                  \triangleright with prosumers on bus y and neighbouring buses j \in \mathcal{B}_y
15:
               communication
16:
           end
       end for
17:
18: end
```

### Algorithm 4 Prosumer i update

```
1: primal update
                                                                                                                         ▷ power generated, stored, from the grid, traded
              a_{i}(k) = \operatorname{col}\left(-\mu_{y}^{\operatorname{pb}}(k), -\mu_{y}^{\operatorname{pb}}(k), \bar{\lambda}_{y}^{\operatorname{mg}}(k) + \mu_{y}^{\operatorname{tg}}(k), \left\{\mu_{(i,j)}^{\operatorname{tr}}(k)\right\}_{j \in \mathcal{N}_{i}}\right)
u_{i}(k+1) = \begin{cases} \operatorname{argmin}_{\xi \in \mathbb{R}^{n_{i}}} & J_{i}(\xi, \sum_{y \in \mathcal{B}} \sigma_{y}^{\operatorname{mg}}(k)) + a_{i}(k)^{\top} \xi + \frac{1}{2} \|\xi - u_{i}(k)\|_{A_{i}}^{2} \\ \operatorname{s.t.} & \xi \in \mathcal{U}_{i} \end{cases}
                                                                                                                                                                                                        ⊳ aux. vector
 3:
                                                                                                                                                                                              ▶ quadratic progr.
 4: end
 5: communication
               b_i(k+1) = p_i^{\rm d} - p_i^{\rm di}(k+1) - p_i^{\rm st}(k+1)
p_i^{\rm mg}(k+1), b_i(k+1) \longrightarrow \text{Bus y},
                                                                                                                                                     \triangleright local load unbalance of prosumer i
 6:
                                                                                                                                                                                            \triangleright forward to Bus y
 7:
               for all prosumer j \in \mathcal{N}_i do
              p_{(i,j)}^{\mathrm{tr}}(k+1) \longrightarrow \mathrm{prosumer}\ j end for
 9:
                                                                                                                                                       \triangleright forward local trade to prosumer j
10:
11: end
12: dual update

▷ reciprocity constraints

               for all j \in \mathcal{N}_i do
13:
                      c_{(i,j)}^{\text{tr}}(k+1) = p_{(i,j)}^{\text{tr}}(k+1) + p_{(j,i)}^{\text{tr}}(k+1)
14:
                                                                                                                                                                                                        ⊳ aux. vector
                      \mu_{(i,j)}^{\text{tr}}(k+1) = \mu_{(i,j)}(k) + \beta_{ij}^{\text{tr}} \left( 2c_{(i,j)}^{\text{tr}}(k+1) - c_{(i,j)}^{\text{tr}}(k) \right)
                                                                                                                                                                                    \rhd reflected dual ascent
15:
               end for
16:
17: end
```

### **Algorithm 5** DSO bus y update

```
1: primal update
                                                                                                                                                                                                                                                                                              ▷ physical variables
                       a_y^{\theta}(k) = 4 \sum_{z \in \mathcal{B}_y} \left( B_{(y,z)}^{\top} \mu_{(y,z)}^p(k) + G_{(y,z)}^{\top} \mu_{(y,z)}^q(k) \right)

    aux. vector

                       a_{y}^{\mathbf{v}}(k) = 4\sum_{z \in \mathcal{B}_{y}} \left(B_{(y,z)}^{(y,z)} \mu_{(y,z)}^{(y,z)}(k) - G_{(y,z)}^{(y,z)} \mu_{(y,z)}^{(p)}(k)\right)
                                                                                                                                                                                                                                                                                                                   ▷ aux. vector
                      a_y(k) = \operatorname{col}\left(a_y^{\theta}(k), a_y^{\mathsf{v}}(k), -\mu_y^{\mathsf{tg}}(k) - \mu_y^{\mathsf{pb}}(k), \left\{\mu_{(y,z)}^{p,c}(k) - \mu_y^{\mathsf{pb}}(k) - \mu_y^{\mathsf{pb}}(k) - \mu_{(y,z)}^{p}(k), \mu_{(y,z)}^{q,c} - \mu_{(y,z)}^{q}(k)\right\}_{z \in \mathcal{B}_{\mathsf{u}}}
                       u_y(k+1) = \operatorname{proj}_{\mathcal{U}_y}(u_y(k) - A_y a_y(k))
                                                                                                                                                                                                                                                                          ⊳ solved via Algorithm 2
  6: end
  7: aggregation update
                       \sigma_y^{\text{mg}}(k+1) = \sum_{i \in \mathcal{N}_y} p_i^{\text{mg}}(k+1)
                                                                                                                                                                                                   \triangleright aggregate grid-to-prosumers power on bus y
                       b_y(k+1) = \sum_{i \in \mathcal{N}_x} b_i(k+1)
                                                                                                                                                                                                                                    \triangleright aggregate load unbalance on bus y
11: auxiliary update
                       w_{y}^{\text{mg}}(k+1) = w_{y}^{\text{mg}}(k) + \delta_{y}^{\text{mg}}\left(|\mathcal{N}_{y}|\lambda_{y}^{\text{mg}}(k) - \sum_{z \in \mathcal{B}_{y}} \lambda_{y}^{\text{mg}}(z)\right)
                                                                                                                                                                                                                                                                                         \triangleright consensus on \lambda_u^{\text{mg}}'s
                       w_y^{\text{tg}}(k+1) = w_y^{\text{tg}}(k) + \delta_y^{\text{tg}}\left(|\mathcal{N}_y|\lambda_y^{\text{tg}}(k) - \sum_{z \in \mathcal{B}_u} \lambda_y^{\text{tg}}(z)\right)
13:
                                                                                                                                                                                                                                                                                           \triangleright consensus on \lambda_u^{\text{tg}}'s
14: end
15: dual update(global grid constraints)
                       c_y^{\mathrm{mg}}(k+1) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \sigma_y^{\mathrm{mg}}(k+1) - |\mathcal{B}|^{-1} \begin{bmatrix} \overline{p}^{\mathrm{mg}} \mathbf{1}_H \\ -\underline{p}^{\mathrm{mg}} \mathbf{1}_H \end{bmatrix} + w_y^{\mathrm{mg}}(k+1)

    aux. vector

                       \lambda_y^{\mathrm{mg}}(k+1) = \mathrm{proj}_{\mathbb{R}^{2H}_{>0}} \left( \lambda_y^{\mathrm{mg}}(k) + \gamma_y^{\mathrm{mg}} (2c_y^{\mathrm{ng}}(k+1) - c_y^{\mathrm{mg}}(k)) \right)
                                                                                                                                                                                                                                                                                                                   ⊳ grid constr.
                      \begin{array}{l} c_y^{\mathrm{tg}}(k+1) = \sigma_y^{\mathrm{mg}}(k+1) - p_y^{\mathrm{tg}}(k+1) + w_y^{\mathrm{tg}}(k+1) \\ \mu_y^{\mathrm{tg}}(k+1) = \mu_y^{\mathrm{tg}}(k) + \beta_y^{\mathrm{tg}}(2c^{\mathrm{tg}}(k+1) - c^{\mathrm{tg}}(k)) \end{array}

    b aux. vector

                                                                                                                                                                                                                                                                      19:
20: end
           dual update (power balance on bus y)
                      c_y^{\text{pb}}(k+1) = p_y^{\text{pd}} + b_y(k+1) - p_y^{\text{tg}}(k+1) - \sum_{z \in \mathcal{B}_y} p_{(y,z)}^{\ell}(k+1)\mu_y^{\text{pb}}(k+1) = \mu_y^{\text{pb}}(k) + \beta_y^{\text{pb}}(2c_y^{\text{pb}}(k+1) - c_y^{\text{pb}}(k))

    aux. vector

23:
                                                                                                                                                                                                                                                                            \triangleright power balance of bus y
24: end
            {\bf communication}
                                                                                                                                                                                                                                                                                                                         ▷ broadcast
                        \begin{split} & \overline{\lambda}_y^{\text{mg}}(k+1) = \begin{bmatrix} I_H \\ -I_H \end{bmatrix}^\top \lambda_y^{\text{mg}}(k+1) \\ & \{ \overline{\lambda}_y^{\text{mg}}(k+1), \mu_y^{\text{tg}}(k+1), \mu_y^{\text{pb}}(k+1) \} \longrightarrow \mathcal{N}_y \\ & \{ \lambda_y^{\text{mg}}(k+1), \mu_y^{\text{tg}}(k+1) \} \longrightarrow \mathcal{B}_y \end{split} 
26:
                                                                                                                                                                                                                                                                  \triangleright to all prosumers on bus y
27:
                                                                                                                                                                                                                                                                            ▷ to the neighbour buses
28:
                       for all neighbour bus z \in \mathcal{B}_y do
29:
                                    \{\theta_y(k+1), v_y(k+1), \mu_{(y,z)}^p(k+1), \mu_{(y,z)}^q(k+1)\} \longrightarrow \text{bus } z
                                                                                                                                                                                                                                                                                                 \triangleright forward to bus z
30:
                       end for
31:
32: end
            dual update (power flow equations)
33:
                       for all bus z \in \mathcal{B}_z (in parallel) do
34:
                                  \begin{array}{ll} b_{(y,z)}^{p}(k+1) = p_{(y,z)}^{\ell}(k+1) - p_{(z,y)}^{\ell}(k+1) & > \text{aux. vectors} \\ c_{(y,z)}^{p}(k+1) = 2B_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) - 2G_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}(k+1) \\ \mu_{(y,z)}^{p}(k+1) = \mu_{(y,z)}^{p}(k) + \beta_{(y,z)}^{p}(2c_{(y,z)}^{p}(k+1) - c_{(y,z)}^{p}(k)) & > \text{power flow eq.} \\ d_{(y,z)}^{p}(k+1) = p_{(y,z)}^{\ell}(k+1) + p_{(z,y)}^{\ell}(k+1) & > \text{aux. vectors} \end{array}
35:
36:
37:
                                                                                                                                                                                                                                                                                                                  ⊳ aux. vector
38:
                                  \mu_{(y,z)}^{p,c}(k+1) = \mu_{(y,z)}^{p,c}(k) + \beta_{(y,z)}^{p,c}\left(2d_{(y,z)}^{p}(k+1) - d_{(y,z)}^{p}(k)\right) \qquad \text{physical reciprocity constraints} \\ b_{(y,z)}^{q}(k+1) = q_{(y,z)}^{\ell}(k+1) - q_{(z,y)}^{\ell}(k+1) \qquad \qquad \text{paux. vector} \\ c_{(y,z)}^{q}(k+1) = 2G_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) + 2B_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}^{q}(k+1) \\ \vdots \\ aux. vector \\ c_{(y,z)}^{q}(k+1) = 2G_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) + 2B_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}^{q}(k+1) \\ \vdots \\ aux. vector \\ c_{(y,z)}^{q}(k+1) = 2G_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) + 2B_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}^{q}(k+1) \\ \vdots \\ aux. vector \\ c_{(y,z)}^{q}(k+1) = 2G_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) + 2B_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}^{q}(k+1) \\ \vdots \\ aux. vector \\ c_{(y,z)}^{q}(k+1) = 2G_{(y,z)}(\theta_{y}(k+1) - \theta_{z}(k+1)) + 2B_{(y,z)}(v_{y}(k+1) - v_{z}(k+1)) - b_{(y,z)}^{q}(k+1) \\ \vdots \\ aux. vector \\ aux. 
                                                                                                                                                                                                                                                             \triangleright physical reciprocity constr.
39:
40:
41:
                                  \mu_{(y,z)}^{qq}(k+1) = \mu_{(y,z)}^{q}(k) + \beta_{(y,z)}^{q}(2c_{(y,z)}^{q}(k+1) - c_{(y,z)}^{q}(k))
d_{(y,z)}^{q}(k+1) = q_{(y,z)}^{\ell}(k+1) + q_{(z,y)}^{\ell}(k+1)
                                                                                                                                                                                                                                                                                                    ⊳ power flow eq.2
42:
43:
                                   \mu_{(y,z)}^{q,c}(k+1) = \mu_{(y,z)}^{q,c}(k) + \beta_{(y,z)}^{q,c} \left( 2d_{(y,z)}^q(k+1) - d_{(y,z)}^q(k) \right)
                                                                                                                                                                                                                                                             ▶ physical reciprocity constr.
44:
                       end for
45:
46: end
```

# References

- [1] G. Belgioioso and S. Grammatico, "Semi-decentralized generalized nash equilibrium seeking in monotone aggregative games,"  $arXiv\ preprint\ arXiv:2003.04031$ , 2020.
- [2] —, "A distributed proximal-point algorithm for nash equilibrium seeking in generalized potential games with linearly coupled cost functions," in 2019 18th European Control Conference (ECC). IEEE, 2019, pp. 1–6.