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* $A_i = \text{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j \in \mathcal{N}_i}) \otimes I_H$*, 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* $A_{N+1} := \text{diag}\left(\left\{\alpha_y^\theta, \alpha_y^v, \alpha_y^{tg}, \{\alpha_{(y,z)}^p, \alpha_{(y,z)}^q\}_{z \in \mathcal{B}_y}\right\}_{y \in \mathcal{B}}\right) \otimes I_H$*, 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**
3:         **primal update**   ▷ power generated, stored, from the grid, traded
4:                 $a_i(k) = \mathrm{col}\left(-\mu_y^{\mathrm{pb}}(k), -\mu_y^{\mathrm{pb}}(k), \left[\begin{smallmatrix} I_H \\ -I_H \end{smallmatrix}\right]^\top \lambda^{\mathrm{mg}}(k) + \mu^{\mathrm{tg}}(k), \left\{\mu_{(i,j)}^{\mathrm{tr}}(k)\right\}_{j \in \mathcal{N}_i}\right)$ ▷ aux. vector
5:                 $u_i(k+1) = \begin{cases} \underset{\xi \in \mathbb{R}^{n_i}}{\mathrm{argmin}} & J_i\left(\xi, \sigma^{\mathrm{mg}}(k)\right) + a_i(k)^\top \xi + \frac{1}{2}\|\xi - u_i(k)\|_{A_i}^2 \\ \text{s.t.} & \xi \in \mathcal{U}_i \end{cases}$ ▷ quadratic progr.
6:         **end**
7:         **communication**   ▷ to DSO and trading partners
8:             $b_i(k+1) = p_i^{\mathrm{d}} - p_i^{\mathrm{di}}(k+1) - p_i^{\mathrm{st}}(k+1)$ ▷ local load unbalance of prosumer i
9:             $p_i^{\mathrm{mg}}(k+1), b_i(k+1) \longrightarrow \mathrm{DSO},$ ▷ forward to DSO
10:             **for all** prosumer $j \in \mathcal{N}_i$ **do**
11:                 $p_{(i,j)}^{\mathrm{tr}}(k+1) \longrightarrow \text{prosumer } j$ ▷ forward local trade to prosumer j
12:             **end for**
13:         **end**
14:         **dual update**   ▷ reciprocity constraints
15:             **for all** $j \in \mathcal{N}_i$ **do**
16:                 $c_{(i,j)}^{\mathrm{tr}}(k+1) = p_{(i,j)}^{\mathrm{tr}}(k+1) + p_{(j,i)}^{\mathrm{tr}}(k+1)$ ▷ aux. vector
17:                 $\mu_{(i,j)}^{\mathrm{tr}}(k+1) = \mu_{(i,j)}(k) + \beta_{ij}^{\mathrm{tr}}\left(2c_{(i,j)}^{\mathrm{tr}}(k+1) - c_{(i,j)}^{\mathrm{tr}}(k)\right)$ ▷ reflected dual ascent
18:             **end for**
19:         **end**
20:     **end for**
21:     **DSO update**
22:         **primal update**   ▷ physical variables
23:             $a_{N+1}(k) = \mathrm{col}\left(\left\{\mathbf{0}, \mathbf{0}, -\mu^{\mathrm{tg}}(k) - \mu_y^{\mathrm{pb}}(k), \{-\mu_y^{\mathrm{pb}}(k), \mathbf{0}\}_{z \in \mathcal{B}_y}\right\}_{y \in \mathcal{B}}\right)$ ▷ aux. vector
24:             $u_{N+1}(k+1) = \mathrm{proj}_{\mathcal{U}_{N+1}}\left(u_{N+1}(k) - A_{N+1}a_{N+1}(k)\right)$ ▷ solved via Algorithm 2
25:         **end**
26:         **aggregation update**
27:             $\sigma^{\mathrm{mg}}(k+1) = \sum_{i \in \mathcal{N}} p_i^{\mathrm{mg}}(k+1)$ ▷ aggregate grid-to-prosumers power
28:             $\sigma^{\mathrm{tg}}(k+1) = \sum_{y \in \mathcal{B}} p_y^{\mathrm{tg}}(k+1)$ ▷ aggregate grid-to-buses power
29:         **end**
30:         **dual update**
31:             $b_{N+1}(k+1) = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right] \otimes (2\sigma^{\mathrm{mg}}(k+1) - \sigma^{\mathrm{mg}}(k)) - \left[\begin{smallmatrix} \bar{p}^{\mathrm{mg}}\mathbf{1}_H \\ -\underline{p}^{\mathrm{mg}}\mathbf{1}_H \end{smallmatrix}\right]$ ▷ aux. vector
32:             $\lambda^{\mathrm{mg}}(k+1) = \mathrm{proj}_{\mathbb{R}_{\geq 0}^{2H}}\left(\lambda^{\mathrm{mg}}(k) + \gamma^{\mathrm{mg}}b_{N+1}(k+1)\right)$ ▷ grid constraints
33:             **for all** buses $y \in \mathcal{B}$ **do**
34:                 $c_y^{\mathrm{pb}}(k+1) = p_y^{\mathrm{pd}} + \sum_{i \in \mathcal{N}_y} b_i(k+1) - p_y^{\mathrm{tg}}(k+1) - \sum_{z \in \mathcal{B}_y} p_{(y,z)}^{\ell}(k+1)$ ▷ aux. vector
35:                 $\mu_y^{\mathrm{pb}}(k+1) = \mu_y^{\mathrm{pb}}(k) + \beta_y^{\mathrm{pb}}(2c_y^{\mathrm{pb}}(k+1) - c_y^{\mathrm{pb}}(k))$ ▷ local power balance of bus y
36:             **end for**
37:             $c^{\mathrm{tg}}(k+1) = \sigma^{\mathrm{mg}}(k+1) - \sigma^{\mathrm{tg}}(k+1)$ ▷ aux. vector
38:             $\mu^{\mathrm{tg}}(k+1) = \mu^{\mathrm{tg}}(k) + \beta^{\mathrm{tg}}(2c^{\mathrm{tg}}(k+1) - c^{\mathrm{tg}}(k))$ ▷ grid-to-buses constraints
39:         **end**
40:         **communication**   ▷ broadcast
41:             $\{\sigma(k+1), \lambda^{\mathrm{mg}}(k+1), \mu^{\mathrm{tg}}(k+1)\} \longrightarrow \mathcal{N}$ ▷ to all prosumers
42:             **for all** buses $y \in \mathcal{B}$ **do**
43:                 $\mu_y^{\mathrm{pb}}(k+1) \longrightarrow \mathcal{N}_y$ ▷ to all prosumers on bus y
44:             **end for**
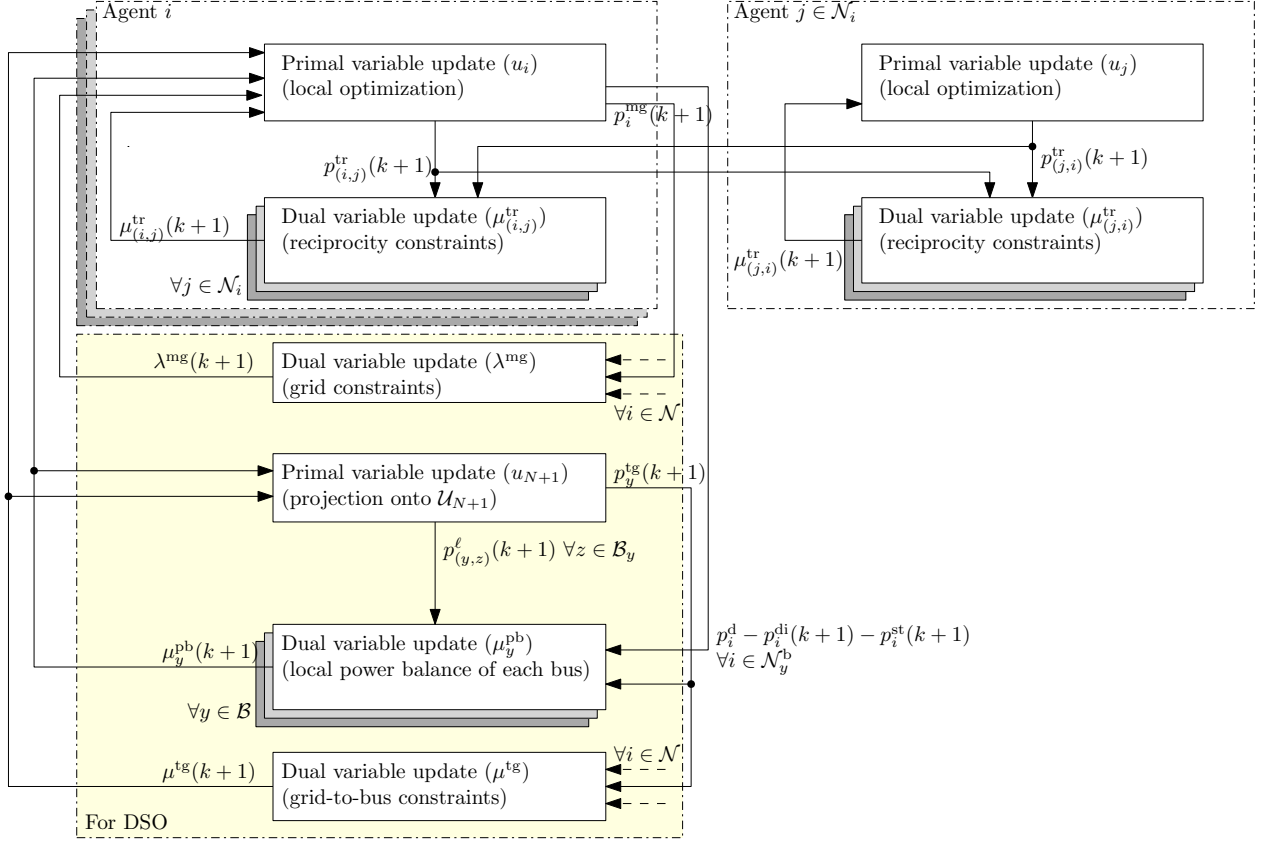45:   **end**
46:     **end**
47: **end**

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 $u_{N+1} = \text{col}\left(\{\theta_y, v_y, p_y^{\text{tg}}, \{p_{(y,z)}^\ell, q_{(y,z)}^\ell\}_{z \in \mathcal{B}_y}\}_{y \in \mathcal{B}}\right)$. The projection onto C_1 can be characterize in closed-form as follows:

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

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

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

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

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

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

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

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: $\quad z(k) = \text{proj}_{C_1}(\frac{1}{2}\xi(k) + \frac{1}{2}x)$
3: $\quad \xi(k+1) = \xi(k) + \lambda\left(\text{proj}_{C_2}(2z(k) - \xi(k)) - z(k)\right), \quad$ 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 $A_i = \mathrm{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j \in \mathcal{N}_i}) \otimes I_H$, 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 $A_y := \mathrm{diag}\left(\alpha_y^{\theta}, \alpha_y^{v}, \alpha_y^{tg}, \{\alpha_{(y,z)}^{p}, \alpha_{(y,z)}^{q}\}_{z \in \mathcal{B}_y}\right) \otimes I_H$, 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 $\beta_{(z,y)}^{p,c} = \beta_{(y,z)}^{p} < (2\|B_{(y,z)}\| + 2\|G_{(y,z)}\| + 2)^{-1}$, $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q} < (2\|B_{(y,z)}\| + 2\|G_{(y,z)}\| + 2)^{-1}$, $\forall z \in \mathcal{B}_y$. Set $\beta_{(z,y)}^{p,c} = \beta_{(y,z)}^{p,c} < \frac{1}{2}$, $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q,c} < \frac{1}{2}$, $\forall z \in \mathcal{B}_y$.

$\square$

- the decision variable of bus y is $u_y := \left(\theta_y, v_y^{\mathrm{v}}, p_y^{\mathrm{tg}}, \{p_{(y,z)}^{\ell}, q_{(y,z)}^{\ell}\}_{z \in \mathcal{B}_y}\right) \in \mathbb{R}^d$, with $d := (3 + 2|\mathcal{B}_y|)24$.

- Let us introduce the local sets of constraints of bus y, i.e., $\mathcal{U}_y := \left\{u \in \mathbb{R}^d \,|\, (17) - (20) \text{ are satisfied}\right\}$.

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

Algorithm 4 Prosumer i update

1: **primal update**   ▷ power generated, stored, from the grid, traded

2:     $a_i(k) = \text{col}\left(-\mu_y^{\text{pb}}(k), -\mu_y^{\text{pb}}(k), \bar{\lambda}_y^{\text{mg}}(k) + \mu_y^{\text{tg}}(k), \left\{\mu_{(i,j)}^{\text{tr}}(k)\right\}_{j \in \mathcal{N}_i}\right)$ ▷ aux. vector

3:     $u_i(k+1) = \begin{cases} \underset{\xi \in \mathbb{R}^{n_i}}{\text{argmin}} & J_i\left(\xi, \sum_{y \in \mathcal{B}} \sigma_y^{\text{mg}}(k)\right) + a_i(k)^\top \xi + \frac{1}{2}\|\xi - u_i(k)\|_{A_i}^2 \\ \text{s.t.} & \xi \in \mathcal{U}_i \end{cases}$ ▷ quadratic progr.

4: **end**

5: **communication**

6:     $b_i(k+1) = p_i^{\text{d}} - p_i^{\text{di}}(k+1) - p_i^{\text{st}}(k+1)$ ▷ local load unbalance of prosumer i

7:     $p_i^{\text{mg}}(k+1), b_i(k+1) \longrightarrow \text{Bus y,}$ ▷ forward to Bus y

8:     **for all** prosumer $j \in \mathcal{N}_i$ **do**

9:        $p_{(i,j)}^{\text{tr}}(k+1) \longrightarrow \text{prosumer } j$ ▷ forward local trade to prosumer j

10:     **end for**

11: **end**

12: **dual update**   ▷ reciprocity constraints

13:     **for all** $j \in \mathcal{N}_i$ **do**

14:        $c_{(i,j)}^{\text{tr}}(k+1) = p_{(i,j)}^{\text{tr}}(k+1) + p_{(j,i)}^{\text{tr}}(k+1)$ ▷ aux. vector

15:        $\mu_{(i,j)}^{\text{tr}}(k+1) = \mu_{(i,j)}(k) + \beta_{ij}^{\text{tr}}\left(2c_{(i,j)}^{\text{tr}}(k+1) - c_{(i,j)}^{\text{tr}}(k)\right)$ ▷ reflected dual ascent

16:     **end for**

17: **end**

Algorithm 5 DSO bus y update

1: **primal update** $\hspace{8cm}$ ▷ physical variables

2: $\quad a_y^\theta(k) = 4\sum_{z\in\mathcal{B}_y}\left(B_{(y,z)}^\top \mu_{(y,z)}^p(k) + G_{(y,z)}^\top \mu_{(y,z)}^q(k)\right)$ $\hspace{3cm}$ ▷ aux. vector

3: $\quad a_y^{\mathrm{v}}(k) = 4\sum_{z\in\mathcal{B}_y}\left(B_{(y,z)}^\top \mu_{(y,z)}^q(k) - G_{(y,z)}^\top \mu_{(y,z)}^p(k)\right)$ $\hspace{3cm}$ ▷ aux. vector

4: $\quad a_y(k) = \mathrm{col}\left(a_y^\theta(k), a_y^{\mathrm{v}}(k), -\mu_y^{\mathrm{tg}}(k)-\mu_y^{\mathrm{pb}}(k), \left\{\mu_{(y,z)}^{p,c}(k)-\mu_y^{\mathrm{pb}}(k)-\mu_{(y,z)}^p(k), \mu_{(y,z)}^{q,c}(k)-\mu_{(y,z)}^q(k)\right\}_{z\in\mathcal{B}_y}\right)$

5: $\quad u_y(k+1) = \mathrm{proj}_{\mathcal{U}_y}\left(u_y(k) - A_y a_y(k)\right)$ $\hspace{4cm}$ ▷ solved via Algorithm 2

6: **end**

7: **aggregation update**

8: $\quad \sigma_y^{\mathrm{mg}}(k+1) = \sum_{i\in\mathcal{N}_y} p_i^{\mathrm{mg}}(k+1)$ $\hspace{2.5cm}$ ▷ aggregate grid-to-prosumers power on bus y

9: $\quad b_y(k+1) = \sum_{i\in\mathcal{N}_y} b_i(k+1)$ $\hspace{4cm}$ ▷ aggregate load unbalance on bus y

10: **end**

11: **auxiliary update**

12: $\quad w_y^{\mathrm{mg}}(k+1) = w_y^{\mathrm{mg}}(k) + \delta_y^{\mathrm{mg}}\left(|\mathcal{N}_y|\lambda_y^{\mathrm{mg}}(k) - \sum_{z\in\mathcal{B}_y}\lambda_y^{\mathrm{mg}}(z)\right)$ $\hspace{1.5cm}$ ▷ consensus on λ_y^{mg}'s

13: $\quad w_y^{\mathrm{tg}}(k+1) = w_y^{\mathrm{tg}}(k) + \delta_y^{\mathrm{tg}}\left(|\mathcal{N}_y|\lambda_y^{\mathrm{tg}}(k) - \sum_{z\in\mathcal{B}_y}\lambda_y^{\mathrm{tg}}(z)\right)$ $\hspace{1.5cm}$ ▷ consensus on λ_y^{tg}'s

14: **end**

15: **dual update**(global grid constraints)

16: $\quad c_y^{\mathrm{mg}}(k+1) = \left[\begin{smallmatrix}1\\-1\end{smallmatrix}\right]\otimes\sigma_y^{\mathrm{mg}}(k+1) - |\mathcal{B}|^{-1}\left[\begin{smallmatrix}\overline{p}^{\mathrm{mg}}\mathbf{1}_H\\-\underline{p}^{\mathrm{mg}}\mathbf{1}_H\end{smallmatrix}\right] + w_y^{\mathrm{mg}}(k+1)$ $\hspace{2cm}$ ▷ aux. vector

17: $\quad \lambda_y^{\mathrm{mg}}(k+1) = \mathrm{proj}_{\mathbb{R}^{2H}_{\geq 0}}\left(\lambda_y^{\mathrm{mg}}(k) + \gamma_y^{\mathrm{mg}}(2c_y^{\mathrm{mg}}(k+1) - c_y^{\mathrm{mg}}(k))\right)$ $\hspace{2cm}$ ▷ grid constr.

18: $\quad c_y^{\mathrm{tg}}(k+1) = \sigma_y^{\mathrm{mg}}(k+1) - p_y^{\mathrm{tg}}(k+1) + w_y^{\mathrm{tg}}(k+1)$ $\hspace{3cm}$ ▷ aux. vector

19: $\quad \mu_y^{\mathrm{tg}}(k+1) = \mu_y^{\mathrm{tg}}(k) + \beta_y^{\mathrm{tg}}(2c^{\mathrm{tg}}(k+1) - c^{\mathrm{tg}}(k))$ $\hspace{2cm}$ ▷ grid-to-buses constraints

20: **end**

21: **dual update** (power balance on bus y)

22: $\quad c_y^{\mathrm{pb}}(k+1) = p_y^{\mathrm{pd}} + b_y(k+1) - p_y^{\mathrm{tg}}(k+1) - \sum_{z\in\mathcal{B}_y} p_{(y,z)}^\ell(k+1)$ $\hspace{2cm}$ ▷ aux. vector

23: $\quad \mu_y^{\mathrm{pb}}(k+1) = \mu_y^{\mathrm{pb}}(k) + \beta_y^{\mathrm{pb}}(2c_y^{\mathrm{pb}}(k+1) - c_y^{\mathrm{pb}}(k))$ $\hspace{2.5cm}$ ▷ power balance of bus y

24: **end**

25: **communication** $\hspace{10cm}$ ▷ broadcast

26: $\quad \bar{\lambda}_y^{\mathrm{mg}}(k+1) = \left[\begin{smallmatrix}I_H\\-I_H\end{smallmatrix}\right]^\top \lambda_y^{\mathrm{mg}}(k+1)$

27: $\quad \{\bar{\lambda}_y^{\mathrm{mg}}(k+1), \mu_y^{\mathrm{tg}}(k+1), \mu_y^{\mathrm{pb}}(k+1)\} \longrightarrow \mathcal{N}_y$ $\hspace{3cm}$ ▷ to all prosumers on bus y

28: $\quad \{\lambda_y^{\mathrm{mg}}(k+1), \mu_y^{\mathrm{tg}}(k+1)\} \longrightarrow \mathcal{B}_y$ $\hspace{4cm}$ ▷ to the neighbour buses

29: $\quad$ **for all** neighbour bus $z\in\mathcal{B}_y$ **do**

30: $\quad\quad \{\theta_y(k+1), v_y(k+1), \mu_{(y,z)}^p(k+1), \mu_{(y,z)}^q(k+1)\} \longrightarrow$ bus z $\hspace{2cm}$ ▷ forward to bus z

31: $\quad$ **end for**

32: **end**

33: **dual update** (power flow equations)

34: $\quad$ **for all** bus $z\in\mathcal{B}_z$ (in parallel) **do**

35: $\quad\quad b_{(y,z)}^p(k+1) = p_{(y,z)}^\ell(k+1) - p_{(z,y)}^\ell(k+1)$ $\hspace{4cm}$ ▷ aux. vectors

36: $\quad\quad c_{(y,z)}^p(k+1) = 2B_{(y,z)}(\theta_y(k+1) - \theta_z(k+1)) - 2G_{(y,z)}(v_y(k+1) - v_z(k+1)) - b_{(y,z)}(k+1)$

37: $\quad\quad \mu_{(y,z)}^p(k+1) = \mu_{(y,z)}^p(k) + \beta_{(y,z)}^p(2c_{(y,z)}^p(k+1) - c_{(y,z)}^p(k))$ $\hspace{2.5cm}$ ▷ power flow eq.1

38: $\quad\quad d_{(y,z)}^p(k+1) = p_{(y,z)}^\ell(k+1) + p_{(z,y)}^\ell(k+1)$ $\hspace{4cm}$ ▷ aux. vector

39: $\quad\quad \mu_{(y,z)}^{p,c}(k+1) = \mu_{(y,z)}^{p,c}(k) + \beta_{(y,z)}^{p,c}\left(2d_{(y,z)}^p(k+1) - d_{(y,z)}^p(k)\right)$ $\hspace{1.5cm}$ ▷ physical reciprocity constr.

40: $\quad\quad b_{(y,z)}^q(k+1) = q_{(y,z)}^\ell(k+1) - q_{(z,y)}^\ell(k+1)$ $\hspace{4cm}$ ▷ aux. vectors

41: $\quad\quad c_{(y,z)}^q(k+1) = 2G_{(y,z)}(\theta_y(k+1) - \theta_z(k+1)) + 2B_{(y,z)}(v_y(k+1) - v_z(k+1)) - b_{(y,z)}^q(k+1)$

42: $\quad\quad \mu_{(y,z)}^q(k+1) = \mu_{(y,z)}^q(k) + \beta_{(y,z)}^q(2c_{(y,z)}^q(k+1) - c_{(y,z)}^q(k))$ $\hspace{2.5cm}$ ▷ power flow eq.2

43: $\quad\quad d_{(y,z)}^q(k+1) = q_{(y,z)}^\ell(k+1) + q_{(z,y)}^\ell(k+1)$ $\hspace{4cm}$ ▷ aux. vector

44: $\quad\quad \mu_{(y,z)}^{q,c}(k+1) = \mu_{(y,z)}^{q,c}(k) + \beta_{(y,z)}^{q,c}\left(2d_{(y,z)}^q(k+1) - d_{(y,z)}^q(k)\right)$ $\hspace{1.5cm}$ ▷ physical reciprocity constr.

45: $\quad$ **end for**

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.