

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

Giuseppe Belgioioso

November 23, 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 [?, 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  $y \in \mathcal{B}$ .  $\square$

---

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

```

---

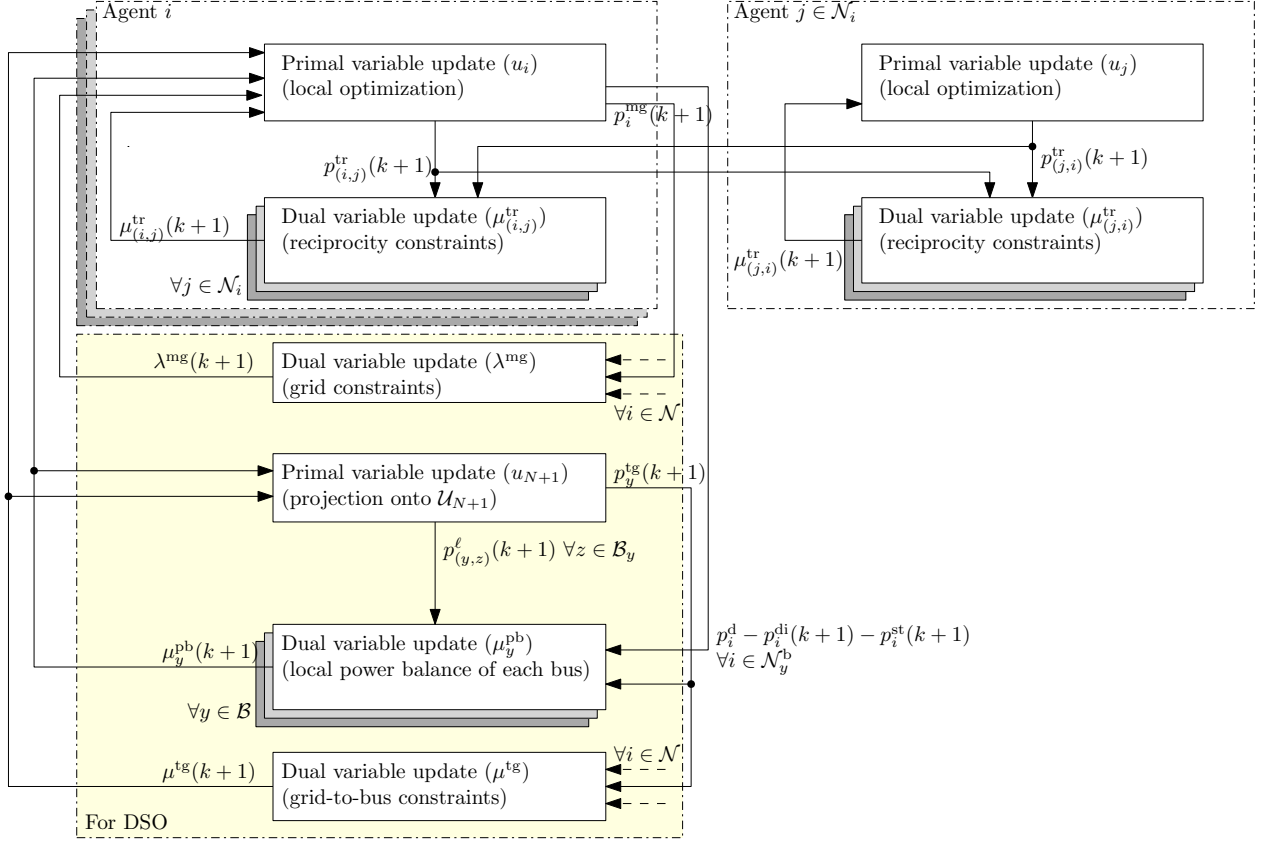


Figure 1: Flowchart of the iterations in Algorithm 1 for agent  $i \in \mathcal{N}$  and DSO. It also shows the flow of information between agent  $i$  and DSO as well as between agent  $i$  and its trading partner  $j$ .

## 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  $y \in \mathcal{B}$ ,

$$\begin{aligned} \theta_y^+ &= \begin{cases} \underline{\theta}_y, & \text{if } \theta_y < \underline{\theta}_y \\ \bar{\theta}_y, & \text{if } \theta_y > \bar{\theta}_y \\ \theta_y, & \text{otherwise} \end{cases}, \\ v_y^+ &= \begin{cases} \underline{v}_y, & \text{if } v_y < \underline{v}_y \\ \bar{v}_y, & \text{if } v_y > \bar{v}_y \\ v_y, & \text{otherwise} \end{cases} \\ p_y^{\text{tg}+} &= \begin{cases} p_y^{\text{tg}}, & \text{if } y \in \mathcal{B}^{\text{mg}} \\ 0, & \text{otherwise} \end{cases} \\ (p_{(y,z),h}^\ell)^+ &= \frac{\bar{s}_{(y,z)}}{\max \left\{ \|\text{col}(p_{(y,z),h}^\ell, q_{(y,z),h}^\ell)\|, \bar{s}_{(y,z)} \right\}} p_{(y,z),h}^\ell, & \forall z \in \mathcal{B}_y, \forall h \in \mathcal{H} \\ (q_{(y,z),h}^\ell)^+ &= \frac{\bar{s}_{(y,z)}}{\max \left\{ \|\text{col}(p_{(y,z),h}^\ell, q_{(y,z),h}^\ell)\|, \bar{s}_{(y,z)} \right\}} q_{(y,z),h}^\ell, & \forall z \in \mathcal{B}_y, \forall h \in \mathcal{H} \end{aligned}$$

The projection onto  $C_2$  can be computed by solving a quadratic programming (e.g. via lsqin, 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:  $z(k) = \text{proj}_{C_1}(\frac{1}{2}\xi(k) + \frac{1}{2}x)$
  - 3:  $\xi(k+1) = \xi(k) + \lambda (\text{proj}_{C_2}(2z(k) - \xi(k)) - z(k)), \quad \text{with } \lambda \in (0, 2)$
  - 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  $A_i = \text{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j \in \mathcal{N}_i}) \otimes I_H$ , with  $\alpha_i^{pi}, \alpha_i^{st} > 1$ ,  $\alpha_i^{mg} > 3 + N \max_{h \in \mathcal{H}} d_h^{mg}$ ,  $\alpha_{(i,j)}^{tr} > 2$ ,  $\forall j \in \mathcal{N}_i$ ,  $\beta_{(j,i)}^{tr} = \beta_{(j,i)}^{tr} < \frac{1}{2}$ ,  $\forall j \in \mathcal{N}_i$ .
- (ii)  $\forall y \in \mathcal{B}$ : set  $A_y := \text{diag}(\alpha_y^\theta, \alpha_y^v, \alpha_y^{tg}, \{\alpha_{(y,z)}^p, \alpha_{(y,z)}^q\}_{z \in \mathcal{B}_y}) \otimes I_H$ , with  $\alpha_y^\theta, \alpha_y^v > 2 \sum_{z \in \mathcal{B}_y} (\|B_{(y,z)}\| + \|G_{((y,z))}\|)$ ,  $\alpha_y^{tg} > 2$ ,  $\alpha_{(y,z)}^p > 2$  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  $\gamma_y^{mg} < (|\mathcal{N}_y^b| + |\mathcal{B}_y|)^{-1}$ . Set  $\beta_y^{tg} < (2|\mathcal{N}_y^b| + |\mathcal{B}_y|)^{-1}$  and  $\beta_y^{pb} < (1 + 2|\mathcal{N}_y^b| + |\mathcal{B}_y|)^{-1}$ . Set  $\beta_{(z,y)}^p = \beta_{(y,z)}^p < (2\|B_{(y,z)}\| + 2\|G_{(y,z)}\| + 2)^{-1}$ ,  $\beta_{(z,y)}^q = \beta_{(y,z)}^q < (2\|B_{(y,z)}\| + 2\|G_{(y,z)}\| + 2)^{-1}$ ,  $\forall z \in \mathcal{B}_y$ . Set  $\beta_{(z,y)}^{p,c} = \beta_{(y,z)}^{p,c} < \frac{1}{2}$ ,  $\beta_{(z,y)}^{q,c} = \beta_{(y,z)}^{q,c} < \frac{1}{2}$ ,  $\forall z \in \mathcal{B}_y$ . □

- the decision variable of bus  $y$  is  $u_y := (\theta_y, v_y, p_y^{tg}, \{p_{(y,z)}^\ell, q_{(y,z)}^\ell\}_{z \in \mathcal{B}_y}) \in \mathbb{R}^d$ , 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

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

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

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

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  $i \in \mathcal{N}_y^b$  (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:    $u_i(k+1) = \begin{cases} \underset{\xi \in \mathbb{R}^{n_i}}{\text{argmin}} & J_i(\xi, \sum_{y \in \mathcal{B}} \sigma_y^{\text{mg}}(k)) + a_i(k)^\top \xi + \frac{1}{2} \|\xi - u_i(k)\|_{A_i}^2 \\ \text{s.t.} & \xi \in \mathcal{U}_i \end{cases}$  ▷ 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 ▷ physical variables
2:    $a_y^\theta(k) = 4 \sum_{z \in \mathcal{B}_y} (B_{(y,z)}^\top \mu_{(y,z)}^p(k) + G_{(y,z)}^\top \mu_{(y,z)}^q(k))$  ▷ aux. vector
3:    $a_y^v(k) = 4 \sum_{z \in \mathcal{B}_y} (B_{(y,z)}^\top \mu_{(y,z)}^q(k) - G_{(y,z)}^\top \mu_{(y,z)}^p(k))$  ▷ aux. vector
4:    $a_y(k) = \text{col} \left( a_y^\theta(k), a_y^v(k), -\mu_y^{\text{tg}}(k) - \mu_y^{\text{pb}}(k), \left\{ \mu_{(y,z)}^{p,c}(k) - \mu_y^{\text{pb}}(k) - \mu_{(y,z)}^p(k), \mu_{(y,z)}^{q,c}(k) - \mu_{(y,z)}^q(k) \right\}_{z \in \mathcal{B}_y} \right)$ 
5:    $u_y(k+1) = \text{proj}_{\mathcal{U}_y} (u_y(k) - A_y a_y(k))$  ▷ solved via Algorithm 2
6: end
7: aggregation update
8:    $\sigma_y^{\text{mg}}(k+1) = \sum_{i \in \mathcal{N}_y} p_i^{\text{mg}}(k+1) + p_y^{\text{pd}}$  ▷ aggregate grid-to-prosumers power on bus  $y$ 
9:    $b_y(k+1) = \sum_{i \in \mathcal{N}_y} b_i(k+1)$  ▷ aggregate load unbalance on bus  $y$ 
10: end
11: auxiliary update
12:    $w_y^{\text{mg}}(k+1) = w_y^{\text{mg}}(k) + \delta_y^{\text{mg}} \left( |\mathcal{B}_y| \lambda_y^{\text{mg}}(k) - \sum_{z \in \mathcal{B}_y} \lambda_z^{\text{mg}}(k) \right)$  ▷ consensus on  $\lambda_y^{\text{mg}}$ 's
13:    $w_y^{\text{tg}}(k+1) = w_y^{\text{tg}}(k) + \delta_y^{\text{tg}} \left( |\mathcal{B}_y| \lambda_y^{\text{tg}}(k) - \sum_{z \in \mathcal{B}_y} \lambda_z^{\text{tg}}(k) \right)$  ▷ consensus on  $\lambda_y^{\text{tg}}$ 's
14: end
15: dual update(global grid constraints)
16:    $c_y^{\text{mg}}(k+1) = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \otimes \sigma_y^{\text{mg}}(k+1) - |\mathcal{B}|^{-1} \begin{bmatrix} \bar{p}^{\text{mg}} \mathbf{1}_H \\ -\bar{p}^{\text{mg}} \mathbf{1}_H \end{bmatrix} - w_y^{\text{mg}}(k+1)$  ▷ aux. vector
17:    $\lambda_y^{\text{mg}}(k+1) = \text{proj}_{\mathbb{R}_{\geq 0}^{2H}} (\lambda_y^{\text{mg}}(k) + \gamma_y^{\text{mg}} (2c_y^{\text{mg}}(k+1) - c_y^{\text{mg}}(k)))$  ▷ grid constr.
18:    $c_y^{\text{tg}}(k+1) = \sigma_y^{\text{mg}}(k+1) - p_y^{\text{tg}}(k+1) - w_y^{\text{tg}}(k+1)$  ▷ aux. vector
19:    $\mu_y^{\text{tg}}(k+1) = \mu_y^{\text{tg}}(k) + \beta_y^{\text{tg}} (2c_y^{\text{tg}}(k+1) - c_y^{\text{tg}}(k))$  ▷ grid-to-buses constraints
20: end
21: dual update (power balance on bus  $y$ )
22:    $c_y^{\text{pb}}(k+1) = p_y^{\text{pd}} + b_y(k+1) - p_y^{\text{tg}}(k+1) - \sum_{z \in \mathcal{B}_y} p_{(y,z)}^\ell(k+1)$  ▷ aux. vector
23:    $\mu_y^{\text{pb}}(k+1) = \mu_y^{\text{pb}}(k) + \beta_y^{\text{pb}} (2c_y^{\text{pb}}(k+1) - c_y^{\text{pb}}(k))$  ▷ power balance of bus  $y$ 
24: end
25: communication ▷ broadcast
26:    $\bar{\lambda}_y^{\text{mg}}(k+1) = \begin{bmatrix} I_H \\ -I_H \end{bmatrix}^\top \lambda_y^{\text{mg}}(k+1)$ 
27:    $\{\lambda_y^{\text{mg}}(k+1), \mu_y^{\text{tg}}(k+1), \mu_y^{\text{pb}}(k+1)\} \longrightarrow \mathcal{N}_y$  ▷ to all prosumers on bus  $y$ 
28:    $\{\lambda_y^{\text{mg}}(k+1), \mu_y^{\text{tg}}(k+1)\} \longrightarrow \mathcal{B}_y$  ▷ to the neighbour buses
29:   for all neighbour bus  $z \in \mathcal{B}_y$  do
30:      $\{\theta_y(k+1), v_y(k+1), \mu_{(y,z)}^p(k+1), \mu_{(y,z)}^q(k+1)\} \longrightarrow \text{bus } z$  ▷ forward to bus  $z$ 
31:   end for
32: end
33: dual update (power flow equations)
34:   for all bus  $z \in \mathcal{B}_z$  (in parallel) do
35:      $b_{(y,z)}^p(k+1) = p_{(y,z)}^\ell(k+1) - p_{(z,y)}^\ell(k+1)$  ▷ aux. vectors
36:      $c_{(y,z)}^p(k+1) = 2B_{(y,z)}(\theta_y(k+1) - \theta_z(k+1)) - 2G_{(y,z)}(v_y(k+1) - v_z(k+1)) - b_{(y,z)}^p(k+1)$ 
37:      $\mu_{(y,z)}^p(k+1) = \mu_{(y,z)}^p(k) + \beta_{(y,z)}^p (2c_{(y,z)}^p(k+1) - c_{(y,z)}^p(k))$  ▷ power flow eq.1
38:      $d_{(y,z)}^p(k+1) = p_{(y,z)}^\ell(k+1) + p_{(z,y)}^\ell(k+1)$  ▷ aux. vector
39:      $\mu_{(y,z)}^{p,c}(k+1) = \mu_{(y,z)}^{p,c}(k) + \beta_{(y,z)}^{p,c} \left( 2d_{(y,z)}^p(k+1) - d_{(y,z)}^p(k) \right)$  ▷ physical reciprocity constr.
40:      $b_{(y,z)}^q(k+1) = q_{(y,z)}^\ell(k+1) - q_{(z,y)}^\ell(k+1)$  ▷ aux. vectors
41:      $c_{(y,z)}^q(k+1) = 2G_{(y,z)}(\theta_y(k+1) - \theta_z(k+1)) + 2B_{(y,z)}(v_y(k+1) - v_z(k+1)) - b_{(y,z)}^q(k+1)$ 
42:      $\mu_{(y,z)}^q(k+1) = \mu_{(y,z)}^q(k) + \beta_{(y,z)}^q (2c_{(y,z)}^q(k+1) - c_{(y,z)}^q(k))$  ▷ power flow eq.2
43:      $d_{(y,z)}^q(k+1) = q_{(y,z)}^\ell(k+1) + q_{(z,y)}^\ell(k+1)$  ▷ aux. vector
44:      $\mu_{(y,z)}^{q,c}(k+1) = \mu_{(y,z)}^{q,c}(k) + \beta_{(y,z)}^{q,c} \left( 2d_{(y,z)}^q(k+1) - d_{(y,z)}^q(k) \right)$  ▷ physical reciprocity constr.
45:   end for
46: end

```

---