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

Giuseppe Belgioioso

November 5, 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* $A_i = \mathrm{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j\in\mathcal{N}_i}) \otimes I_H$, *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) Set $A_{N+1} := \mathrm{diag}\left(\left\{\alpha_y^{\theta}, \alpha_y^{v}, \alpha_y^{tg}, \{\alpha_{(y,z)}^{p}, \alpha_{(y,z)}^{q}\}_{z\in\mathcal{B}_y}\right\}_{y\in\mathcal{B}}\right) \otimes I_H$, *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* $\beta_y^{pb} < (1 + |\mathcal{N}_y| + |\mathcal{B}_y|)^{-1}$, *for all* $y \in \mathcal{B}$. □

Algorithm 1 Semi-decentralized GWE seeking for P2P Energy Markets

Iterate until convergence
  for all prosumer $i \in \mathcal{N}$ **do**
    primal update ▷ power generated, stored, from the grid, traded
      $a_i(k) = \mathrm{col}\big(-\mu_y^{\mathrm{pb}}(k), -\mu_y^{\mathrm{pb}}(k), \left[\begin{smallmatrix} I_H \\ -I_H \end{smallmatrix}\right]^\top \lambda^{\mathrm{mg}}(k) + \mu^{\mathrm{tg}}(k), \Big\{\mu^{\mathrm{tr}}_{(i,j)}(k)\Big\}_{j\in\mathcal{N}_i}\big)$ ▷ aux. vector
      $u_i(k+1) = \begin{cases} \underset{\xi\in\mathbb{R}^{n_i}}{\mathrm{argmin}} & J_i\big(\xi,\sigma(k)\big) + a_i(k)^\top \xi + \frac{1}{2}\left\|\xi - u_i(k)\right\|^2_{A_i} \\ \quad \text{s.t.} & \xi \in \mathcal{U}_i \end{cases}$ ▷ quadratic progr.
    end
    communication ▷ to DSO and trading partners
      $b_i(k+1) = p_i^{\mathrm{d}} - p_i^{\mathrm{di}}(k+1) - p_i^{\mathrm{st}}(k+1)$ ▷ local load unbalance of prosumer i
      $p_i^{\mathrm{mg}}(k+1), b_i(k+1) \longrightarrow \mathrm{DSO},$ ▷ forward to DSO
      for all prosumer $j \in \mathcal{N}_i$ **do**
        $p^{\mathrm{tr}}_{(i,j)}(k+1) \longrightarrow$ prosumer j ▷ forward local trade to prosumer j
      end for
    end
    dual update ▷ reciprocity constraints
      for all $j \in \mathcal{N}_i$ **do**
        $c^{\mathrm{tr}}_{(i,j)}(k+1) = p^{\mathrm{tr}}_{(i,j)}(k+1) + p^{\mathrm{tr}}_{(j,i)} x(k+1)$ ▷ aux. vector
        $\mu_{(i,j)}(k+1) = \mu_{(i,j)}(k) + \beta^{\mathrm{tr}}_{ij}\Big(2c^{\mathrm{tr}}_{(i,j)}(k+1) - c^{\mathrm{tr}}_{(i,j)}(k)\Big)$ ▷ reflected dual ascent
      end for
    end
  end for
  DSO update
    primal update ▷ physical variables
      $a_{N+1}(k+1) = \mathrm{col}\Big(\big\{\mathbf{0}, \mathbf{0}, -\mu^{\mathrm{tg}}(k) - \mu_y^{\mathrm{pb}}(k), \{-\mu_y^{\mathrm{pb}}(k), \mathbf{0}\}_{z\in\mathcal{B}_y}\big\}_{y\in\mathcal{B}}\Big)$ ▷ aux. vector
      $u_{N+1}(k+1) = \mathrm{proj}_{\mathcal{U}_{N+1}}\left(u_{N+1}(k) - A_{N+1}a_{N+1}(k)\right)$ ▷ solved via Algorithm 2
    end
    aggregation update
      $\sigma^{\mathrm{mg}}(k+1) = \sum_{i\in\mathcal{N}} p_i^{\mathrm{mg}}(k+1)$ ▷ aggregate grid-to-prosumers power
      $\sigma^{\mathrm{tg}}(k+1) = \sum_{y\in\mathcal{B}} p_y^{\mathrm{tg}}(k+1)$ ▷ aggregate grid-to-buses power
    end
    dual update
      $b_{N+1}(k+1) = \left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right] \otimes \left(2\sigma^{\mathrm{mg}}(k+1) - \sigma^{\mathrm{mg}}(k)\right) - \left[\begin{smallmatrix} \overline{p}^{\mathrm{mg}}\mathbf{1}_H \\ -\underline{p}^{\mathrm{mg}}\mathbf{1}_H \end{smallmatrix}\right]$ ▷ aux. vector
      $\lambda^{\mathrm{mg}}(k+1) = \mathrm{proj}_{\mathbb{R}^{2H}_{\geq 0}}\left(\lambda^{\mathrm{mg}}(k) + \gamma^{\mathrm{mg}} b_{N+1}(k+1)\right)$ ▷ grid constraints
      for all buses $y \in \mathcal{B}$ **do**
        $c_y^{\mathrm{pb}}(k+1) = p_y^{\mathrm{pd}} + \sum_{i\in\mathcal{N}_y} b_i(k+1) - p_y^{\mathrm{tg}}(k+1) - \sum_{z\in\mathcal{B}_y} p^{\ell}_{(y,z)}(k+1)$ ▷ aux. vector
        $\mu_y^{\mathrm{pb}}(k+1) = \mu_y^{\mathrm{pb}}(k) + \beta_y^{\mathrm{pb}}(2c_y^{\mathrm{pb}}(k+1) - c_y^{\mathrm{pb}}(k))$ ▷ local power balance of bus y
      end for
      $c^{\mathrm{tg}}(k+1) = \sigma^{\mathrm{mg}}(k+1) - \sigma^{\mathrm{tg}}(k+1)$ ▷ aux. vector
      $\mu^{\mathrm{tg}}(k+1) = \mu^{\mathrm{tg}}(k) + \beta^{\mathrm{tg}}(2c^{\mathrm{tg}}(k+1) - c^{\mathrm{tg}}(k))$ ▷ grid-to-buses constraints
    end
    communication ▷ broadcast
      $\{\sigma(k+1), \lambda^{\mathrm{mg}}(k+1), \mu^{\mathrm{tg}}(k+1)\} \longrightarrow \mathcal{N}$ ▷ to all prosumers
      for all buses $y \in \mathcal{B}$ **do**
        $\mu_y^{\mathrm{pb}}(k+1) \longrightarrow \mathcal{N}_y$ ▷ to all prosumers on bus y
      end for
    end
  end
end

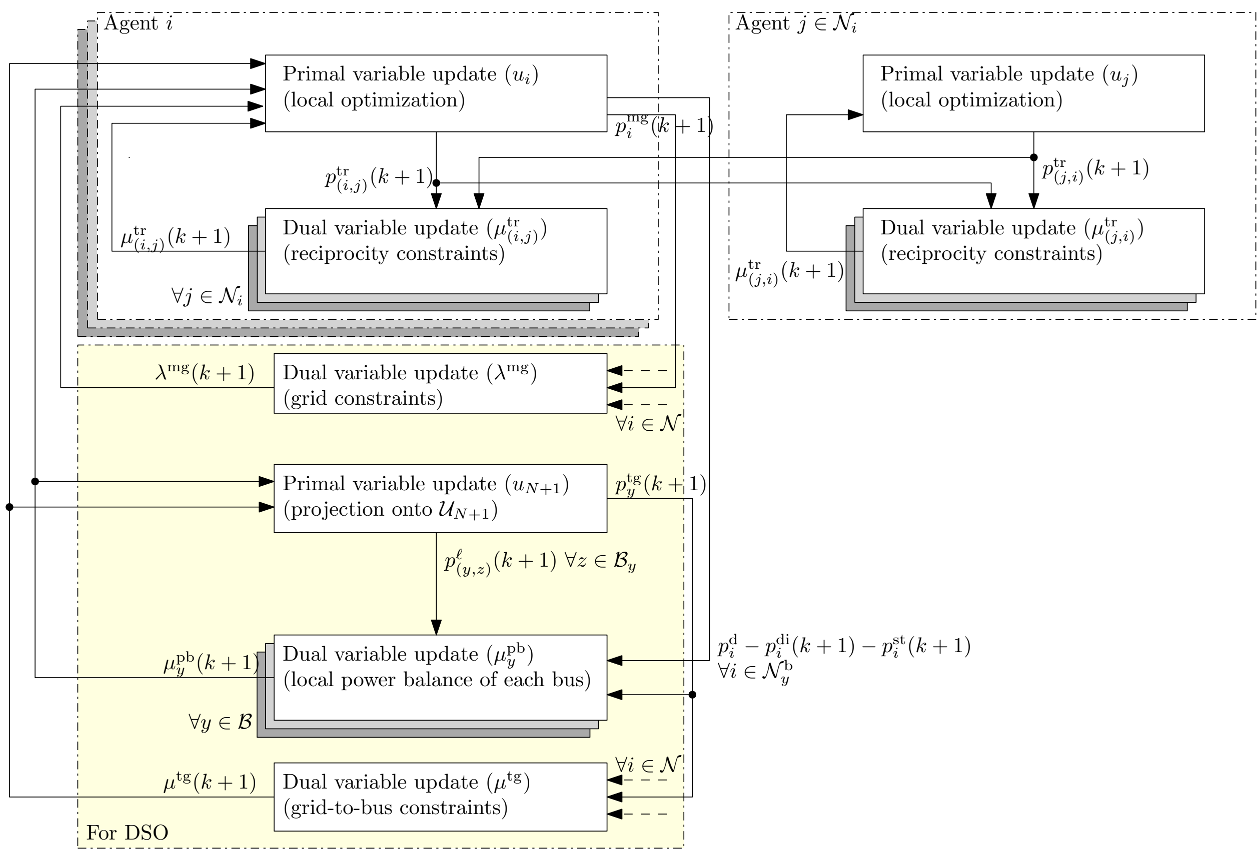

Figure 1: Flowchart of the iterations in Algorithm 1 for agent $i \in \mathcal{N}$ and DSO. It also shows the flow of information between agent i and DSO as well as between agent i and its trading partner j.

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

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

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

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

The projection onto C_2 can be computed by solving a quadratic programming (e.g. via 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: $\quad z(k) = \mathrm{proj}_{C_1}(\frac{1}{2}\xi(k) + \frac{1}{2}x)$
3: $\quad \xi(k+1) = \xi(k) + \lambda\left(\mathrm{proj}_{C_2}(2z(k) - \xi(k)) - z(k)\right), \quad$ with $\lambda \in (0, 2)$
4: **end**

2.3 Fully-distributed

The algorithm presented in this section is a variation of [?, Alg. 3] for the setup considered in the ECC 2020 paper.

Consider the following mixing matrix.

Assumption 2 *For all $k \in \mathbb{N}$, the matrix $W = [w_{i,j}]$ satisfies the following conditions:*

(i) (Edge utilization) Let $i, j \in \mathcal{I}$, $i \neq j$. If $(i,j) \in \mathcal{E}_k$, $w_{i,j} \geq \epsilon$, for some $\epsilon > 0$; $w_{i,j} = 0$ otherwise;

(ii) (Positive diagonal) For all $i \in \mathcal{I}$, $w_{i,i} > \epsilon$;

(iii) (Double-stochasticity) $W\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W = \mathbf{1}^\top$. □

Assumption 2 is strong but typical for multiagent coordination and optimization [?]. For an undirected graph it can be fulfilled, for example, by using Metropolis weights:

$$w_{i,j} = \begin{cases} (\max\{|\mathcal{N}_i|, |\mathcal{N}_j|\})^{-1} & \text{if } (i,j) \in \mathcal{E}, \\ 0 & \text{if } (i,j) \notin \mathcal{E}, \\ 1 - \sum_{\ell \in \mathcal{N}_i} w_{i,\ell} & \text{if } i = j. \end{cases} \tag{1}$$

Consider the following choices for the step-sizes in Algorithm 2.

Assumption 3 (Step size selection) *For all $i \in \mathcal{N}$, set $A_i = \mathrm{blkdiag}(A_{i,1}, \ldots, A_{i,H})$, where $A_{i,h} = \mathrm{diag}\left(\alpha^{\mathrm{dg}}_{i,h},\ \alpha^{\mathrm{st}}_{i,h},\ \alpha^{\mathrm{mg}}_{i,h},\ \{\alpha^{\mathrm{tr}}_{(i,j),h}\}_{j \in \mathcal{N}_i}\right)$, for all $h \in \mathcal{H}$, with $\alpha^{\mathrm{dg}}_{i,h}$, $\alpha^{\mathrm{st}}_{i,h} > 0$, $\alpha^{mg}_{i,h} > q^{mg}_h + 1$ and $\alpha^{\mathrm{tr}}_{(i,j),h} > |\mathcal{N}_i|$, for all $j \in \mathcal{N}_i$; $\beta_{(i,j)} = \beta < \frac{1}{2}$ for all $j \in \mathcal{N}_i$; $\gamma < 1$.*

□

Assumption 4 (Kransol'skii–Mann step) *The sequence $(\delta(k))_{k \in \mathbb{N}}$ satisfies the following conditions:*

(i) (non-increasing) $0 \leq \delta(k)^{k+1} \leq \delta(k) \leq 1$, for all $k \geq 0$;

(ii) (non-summable) $\sum_{k=0}^{\infty} \delta(k) = \infty$;

(iii) (square-summable) $\sum_{k=0}^{\infty} \delta(k)^2 < \infty$. □

Remark 1 (Vanishing KM steps seem not necessary in practice) *For the numerical studies, we can use $\delta(k) = 1$, for all $k \in \mathbb{N}$.*

Algorithm 3 Fully-Distributed GWE seeking for P2P Energy Markets

Initialization: For all $i \in \mathcal{N}$, set locally:

(a) Initial conditions: $u_i(-1), u_i(0), \tilde{u}_i(-1) \in \mathcal{U}_i$, $\mu_{(i,j)}(0) = \mathbf{0}$ for all $j \in \mathcal{N}_i$, $\lambda_i(0) \in \mathbb{R}^{2H}_{\geq 0}$, $\sigma_i(0) = x_i(0)$, $z_i(0) = \lambda_i(0)$, $y_i(0) = 2\tilde{u}_i(-1) - u_i(-1) - \frac{1}{N}\begin{bmatrix} \overline{p}^{\text{mg}}\mathbf{1}_H \\ -\underline{p}^{\text{mg}}N\mathbf{1}_H \end{bmatrix}$.

(b) Step-sizes: A_i, $\{\beta_{ij}\}_{j\in\mathcal{N}_i}$ and γ_i as in Assumption 3; $\{\delta_i(k)\}_{k\in\mathbb{N}}$ as in Assumption 4.

Iterate until convergence:

Local. For each agent $i \in \mathcal{N}$:

(1) Communication with neighboring agents

$$\left\{S^{\text{mg}}u_i(k+1),\, \sigma_i(k),\, y_i(k),\, z_i(k)\right\} \quad \longrightarrow \quad j, \text{ for all } j \in \mathcal{N}_i,$$

(2) Distributed averaging

$$\hat{\sigma}_i(k) = \textstyle\sum_{j=1}^N w_{i,j}\sigma_j(k), \quad \hat{y}_i(k) = \sum_{j=1}^N w_{i,j}y_j(k), \quad \hat{z}_i(k) = \sum_{j=1}^N w_{i,j}z_j(k),$$

(3) Strategy update

$$d_i(k) = A_i^{-1}\left(\left(\left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right] \otimes S_i^{\text{mg}}\right)^\top \hat{z}_i(k) + \sum_{j\in\mathcal{N}_i}\left(S_{(i,j)}^{\text{tr}}\right)^\top \mu_{(i,j)}(k)\right)$$

$$\tilde{u}_i(k) = \begin{cases} \underset{\xi\in\mathbb{R}^{n_i}}{\operatorname{argmin}} & J_i\big(\xi, \hat{\sigma}_i(k)\big) + \frac{1}{2}\left\|\xi - u_i(k) + d_i(k)\right\|_{A_i}^2 \\ \quad \text{s.t.} & \xi \in \mathcal{U}_i \end{cases}$$

(4) Dual variables (reciprocity constraints) update, $\forall j \in \mathcal{N}_i$:

$$\tilde{\mu}_{(i,j)}(k) = \mu_{(i,j)}(k) + \beta_{ij}\left(2S_{(i,j)}^{\text{tr}}\tilde{u}_i(k) - S_{(i,j)}^{\text{tr}}u_i(k) + 2S_{(j,i)}^{\text{tr}}\tilde{u}_j(k) - S_{(j,i)}^{\text{tr}}u_j(k)\right).$$

(5) Dynamic tracking of the estimate y_i:

$$y_i(k+1) = \hat{y}_i(k) + \big(2\tilde{u}_i(k) - u_i(k)\big) - \big(2\tilde{u}_i(k-1) - u_i(k-1)\big),$$

(6) Dual variable (grid constraints) update:

$$\tilde{\lambda}_i(k) = \operatorname{proj}_{\mathbb{R}^{2H}_{\geq 0}}\left(\lambda_i(k) + \gamma_i\left(\left[\begin{smallmatrix} 1 \\ -1 \end{smallmatrix}\right] \otimes y_i(k+1) - \lambda_i(k) + \hat{z}_i(k)\right)\right),$$

(7) Krasnosel'skii-Mann process

$$\begin{aligned} u_i(k+1) &= u_i(k) + \delta_i(k)\big(\tilde{u}_i(k) - u_i(k)\big), \\ \mu_{(i,j)}(k+1) &= \mu_{(i,j)}(k) + \delta_i(k)\big(\tilde{\mu}_{(i,j)}(k) - \mu_{(i,j)}(k)\big), \quad \forall j \in \mathcal{N}_i, \\ \lambda_i(k+1) &= \lambda_i(k) + \delta_i(k)\big(\tilde{\lambda}(k) - \lambda_i(k)\big), \end{aligned}$$

(8) Dynamic tracking of the estimates σ_i and z_i:

$$\begin{aligned} \sigma_i(k+1) &= \hat{\sigma}(k) + u_i(k+1) - u_i(k), \\ z_i(k+1) &= \hat{z}_i(k) + \lambda_i(k+1) - \lambda_i(k) \end{aligned}$$