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

Giuseppe Belgioioso

September 18, 2020

1 Problem statement

Consider the same problem statement in `problem_setup_v6.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* $A_i = \mathrm{diag}(\alpha_i^{pi}, \alpha_i^{st}, \alpha_i^{mg}, \{\alpha_{(i,j)}^{tr}\}_{j\in\mathcal{N}_i}) \otimes I_H$*, with* $\alpha_i^{pi}, \alpha_i^{st} > 1$*,* $\alpha_i^{mg} > 3 + N \max_{h\in\mathcal{H}} d_h$*,* $\beta_{(i,j)}^{tr} = \beta_{(j,i)}^{tr} < \frac{1}{2}$*,* $\forall j \in \mathcal{N}_i$*.*

(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 Euclid's algorithm

```
procedure EUCLID(a, b)                              ▷ The g.c.d. of a and b
    r ← a mod b
    while r ≠ 0 do                                  ▷ We have the answer if r is 0
        a ← b
        b ← r
        r ← a mod b
    end while
    return b                                        ▷ The gcd is b
end procedure
```

Algorithm 2 Semi-decentralized GWE seeking for P2P Energy Markets

procedure INITIALIZATION PROSUMERS
    for all prosumer $i \in \mathcal{N}$ **do**
        Set the initial conditions: $u_i(0) \in \mathcal{U}_i$, $\mu^{\mathrm{tr}}_{(i,j)}(0) = \mathbf{0}$, $\forall j \in \mathcal{N}_i$.
        Set the step sizes as in Assumption 1(i).
    end for
end procedure
procedure INITIALIZATION DSO
    Set the initial conditions: $u_{N+1}(0) \in \mathcal{U}_{N+1}$, $\lambda^{\mathrm{mg}} = \mathbf{0}$, $\mu^{\mathrm{tg}} = \mathbf{0}$, $\mu^{\mathrm{pb}}_y(0) = \mathbf{0}$, $\forall y \in \mathcal{B}$.
    Set the step sizes as in Assumption 1(ii).
end procedure

repeat
    for all prosumer $i \in \mathcal{N}$ **do**
        procedure STRATEGY UPDATE

$$a_i(k) = \mathrm{col}\big(-\mu^{\mathrm{pb}}_y(k), -\mu^{\mathrm{pb}}_y(k), \left[\begin{smallmatrix} I_H \\ -I_H \end{smallmatrix}\right]^\top \lambda^{\mathrm{mg}}(k) + \mu^{\mathrm{tg}}(k), \left\{\mu^{\mathrm{tr}}_{(i,j)}(k)\right\}_{j \in \mathcal{N}_i}\big) \quad \triangleright \text{(aux. vector)}$$

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

        end procedure

(1) Strategy update

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

where

$$a_i(k) = \mathrm{col}\left(-\mu^{\mathrm{pb}}_y(k), -\mu^{\mathrm{pb}}_y(k), \left[\begin{smallmatrix} I_H \\ -I_H \end{smallmatrix}\right]^\top \lambda^{\mathrm{mg}}(k) + \mu^{\mathrm{tg}}(k), \left\{\mu^{\mathrm{tr}}_{(i,j)}(k)\right\}_{j \in \mathcal{N}_i}\right)$$

(2) Communication with trading partners and DSO:

(a) $\left\{\begin{array}{l} p^{\mathrm{mg}}_i(k+1) \\ b_i(k+1) := p^{\mathrm{d}}_i - p^{\mathrm{di}}_i(k+1) - p^{\mathrm{st}}_i(k+1) \end{array}\right\} \longrightarrow$ DSO,

(b) $p^{\mathrm{tr}}_{(i,j)}(k+1) \longrightarrow$ prosumer j, for all $j \in \mathcal{N}_i$.

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

$$\mu_{(i,j)}(k+1) = \mu_{(i,j)}(k) + \beta^{\mathrm{tr}}_{ij}\left(2c^{\mathrm{tr}}_{(i,j)}(k+1) - c^{\mathrm{tr}}_{(i,j)}(k)\right).$$

where

$$c^{\mathrm{tr}}_{(i,j)}(k+1) = p^{\mathrm{tr}}_{(i,j)}(k+1) + p^{\mathrm{tr}}_{(j,i)}x(k+1),$$

    end for

DSO.

(1) Aggregate energy demand update

$$\sigma^{\mathrm{mg}}(k+1) = \sum_{i \in \mathcal{N}} p^{\mathrm{mg}}_i(k+1) + \sum_{i \in \mathcal{B}} p^{\mathrm{pd}}_k$$
$$\sigma^{\mathrm{tg}}(k+1) = \sum_{y \in \mathcal{B}} p^{\mathrm{tg}}_y(k+1)$$

(2) Dual variable (grid constraints) update:

References

[1] G. Belgioioso and S. Grammatico, "Semi-decentralized generalized nash equilibrium seeking in monotone aggregative games," *arXiv preprint arXiv:2003.04031*, 2020.