Appendix
Supplementary Material

A. Algorithm 1: Derivation and Convergence Analysis

The derivation and convergence analysis of Algorithm 1 relies (for the most part) on the *customized preconditioned proximal-point (cPPP)* algorithm for generalized aggregative games proposed in [16, Algorithm 6]. The objective of this appendix is to show that the proposed market-clearing game (1), with cost functions and constraints sets defined in (17)-(19), satisfies all the technical conditions in [16, Theorem 2], among which is the existence of a variational GNE, i.e., item (i) of Proposition 1. Therefore, we invoke [16, Theorem 2] to prove convergence of Algorithm 1, i.e., item (ii) of Proposition 1. For a complete convergence analysis of the cPPP algorithm for aggregative games we refer to [16, Appendix C].

Aggregative cost functions: First, we show that the cost functions (17) can be cast as in [16, Eqn. (30)], i.e.

$$J_i(u_i, u_{-i}) = g_i(u_i) + \left(C\operatorname{avg}(u)\right)^\top u_i, \tag{21}$$

where $\operatorname{avg}(u) := \frac{1}{N}\sum_{i\in\mathcal{N}} u_i$ denotes the average strategy. Let $\mathcal{N}_i = \mathcal{N}$, for all $i \in \mathcal{N}$, without loss of generality[6]. In this case, $u_i \in \mathbb{R}^{(3+N)H}$, for all $i \in \mathcal{N}$. Moreover, let $\Xi^{\mathrm{mg}} \in \mathbb{R}^{H\times(3+N)H}$ denote the matrix that selects the p_i^{mg}-component from the decision vectors u_i's, and define the matrix $D := N\operatorname{diag}(d_1^{\mathrm{mg}}, \dots, d_H^{\mathrm{mg}})$, where d_h^{mg} is the price coefficient for the main grid power. Then, the cost functions in (17) can be recast as [16, Eqn. (30)], or (21), with

$$\begin{aligned} g_i(u_i) &= f_i^{\mathrm{di}}(p_i^{\mathrm{di}}) + f_i^{\mathrm{st}}(p_i^{\mathrm{st}}) + f_i^{\mathrm{tr}}\left(\{p_{(i,j)}^{\mathrm{tr}}\}_{j\in\mathcal{N}_i}\right), \\ &\quad + \tfrac{1}{N}(Db)^\top p_i^{\mathrm{mg}}, \end{aligned} \tag{22a}$$

$$C = (\Xi^{\mathrm{mg}})^\top D\, \Xi^{\mathrm{mg}}. \tag{22b}$$

Technical assumptions: Next, we show that all the assumptions in [16, Theorem 2] are satisfied.

(i) For all $i \in \mathcal{N}^+$, the cost function $J_i(u_i, u_{-i})$ in (22a) is convex in u_i, since all the components of g_i are convex. Hence, [16, Assumption 1] holds.

(ii) For all $i \in \mathcal{N}^+$, the local set $\mathcal{U}_i$ in (18) is nonempty, closed and convex. Moreover, Slater's constraint qualification on the global feasible set $\left(\prod_{i\in\mathcal{N}^+}\mathcal{U}_i\right)\cap\mathcal{C}$ holds under an appropriate choice of the parameters. Therefore, [16, Assumption 2] is satisfied.

(iii) The *pseudo-subdifferential* mapping of the game (1) reads as $F : u \mapsto \prod_{i\in\mathcal{N}}\left(\partial_{u_i} J_i(u_i, u_{-i})\right) \times \mathbf{0}$, since $J_{N+1} = 0$. It follows by [30, Corollary 1], that the first term of F, i.e., $u \mapsto \prod_{i\in\mathcal{N}} \partial_{u_i} J_i(u_i, u_{-i})$, is maximally monotone, since C in (22b) is positive semidefinite, i.e., $C = (\Xi^{\mathrm{mg}})^\top D\, \Xi^{\mathrm{mg}} \succeq 0$. Moreover, also the second term of F, i.e., the zero mapping $\mathbf{0}$, is maximally monotone. Therefore, it follows by [23, Proposition 20.23] that their cartesian product $\prod_{i\in\mathcal{N}}\left(\partial_{u_i} J_i(u_i, u_{-i})\right) \times \mathbf{0} = F$ is maximally monotone. Hence, [16, Assumption 6] holds.

(iv) By [16, Lemma 1 (i)], there exists a variational GNE of the game in (1), since the constraint sets $\mathcal{U}_i$ in (18) are bounded, and the pseudo-subdifferential mapping F is monotone. Hence, [16, Assumption 4] is satisfied.

∎

[6] For example, by defining, for all $i \in \mathcal{N}$, the "dummy variables" $\{p_{(i,j)}^{\mathrm{tr}}\}_{j\in\mathcal{N}\setminus\mathcal{N}_i}$ for all the prosumers that do not trade with i.

B. Alternating Projection for Operational Feasibility

In this appendix, we propose an efficient algorithm to compute the projection onto the set $\mathcal{U}_{N+1}$ (line 30 in Algorithm 1). First, let us recall the structure of u_{N+1}, i.e.,

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

and let us define the sets

$$\mathcal{S}_1 := \{u_{N+1} \mid \text{(15) and (16a) hold}\}, \tag{23}$$

$$\mathcal{S}_2 := \{u_{N+1} \mid \text{(14a) and (14b) hold}\}, \tag{24}$$

such that $\mathcal{U}_{N+1} = \mathcal{S}_1 \cap \mathcal{S}_2$. The proposed method, summarized in Algorithm 2, is essentially a *Douglas–Rachford splitting* (DRS) [23, § 26.3] applied to the *best approximation problem* $\operatorname{argmin}_{\xi\in\mathcal{S}_1\cap\mathcal{S}_2}\|\xi - u_{N+1}\| = \operatorname{proj}_{\mathcal{S}_1\cap\mathcal{S}_2}(u_{N+1})$, see e.g. [33, § 4.3] for a formal derivation of the algorithm.

Unlike $\mathcal{U}_{N+1}$, the projections onto $\mathcal{S}_1$ and $\mathcal{S}_2$ have closed-form expressions, hence Algorithm 2 only involves elementary operations. Specifically, $\operatorname{proj}_{\mathcal{S}_1}(u_{N+1}) = u_{N+1}^+$, where

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

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

and for all $y \in \mathcal{B}$, $z \in \mathcal{B}_z$, and $h \in \mathcal{H}$

$$\begin{aligned} L_{(y,z),h} &= \max\left\{\|\operatorname{col}(p_{(y,z),h}^{\ell}, q_{(y,z),h}^{\ell}\|, \overline{s}_{(y,z)}\right\}, \\ (p_{(y,z),h}^{\ell})^+ &= \tfrac{\overline{s}_{(y,z)}}{L_{(y,z),h}}\, p_{(y,z),h}^{\ell}, \\ (q_{(y,z),h}^{\ell})^+ &= \tfrac{\overline{s}_{(y,z)}}{L_{(y,z),h}}\, q_{(y,z),h}^{\ell}. \end{aligned}$$

Whereas, since $\mathcal{S}_2$ is an affine set, a closed-form expression for $\operatorname{proj}_{\mathcal{S}_2}$ is given in [23, Example 29.17(ii)].

Algorithm 2: DRS for computing $\operatorname{proj}_{\mathcal{U}_{N+1}}(u_{N+1})$

1: Initialize $\xi(0) \in \mathbb{R}^{n_{N+1}}$, and set $\eta \in (0, 2)$
2: **While convergence is not achieved do:**
3: $\quad z(k) = \operatorname{proj}_{\mathcal{S}_1}(\frac{1}{2}\xi(k) + \frac{1}{2}u_{N+1})$
4: $\quad \xi(k+1) = \xi(k) + \eta\left(\operatorname{proj}_{\mathcal{S}_2}(2z(k) - \xi(k)) - z(k)\right)$
5: **end while**