On decentralized convex optimization in a multi-agent setting with separable constraints and its application to optimal charging of electric vehicles

Luca Deori, Kostas Margellos, Maria Prandini

Abstract— **We develop a decentralized algorithm for multi-agent, convex optimization programs, subject to separable constraints, where the constraint function of each agent involves only its local decision vector, while the decision vectors of all agents are coupled via a common objective function. We construct a variant of the so called Jacobi algorithm and show that, when the objective function is quadratic, convergence to some minimizer of the centralized problem counterpart is achieved. Our algorithm serves then as an effective alternative to gradient based methodologies. We illustrate its efficacy by applying it to the problem of optimal charging of electric vehicles, where, as opposed to earlier approaches, we show convergence to an optimal charging scheme for a finite, possibly large, number of vehicles.**

I. Introduction

Optimization in multi-agent systems has attracted significant attention in the control and operations research communities, due to its applicability to different domains, e.g., energy systems, mobility systems, robotic networks, etc. In this paper we focus on a specific class of multi-agent optimization programs that are convex and are subject to constraints that are separable, i.e., the constraint function of each agent involves only its local decision vector. The agents' decision vectors are, however, coupled by means of a common objective function. The considered structure, although specific, captures a wide class of engineering problems, like the electric vehicle optimal charging problem studied in this paper. Solving such problems in a centralized fashion would require agents to share their local constraint functions with each other. This would raise, however, information privacy issues. Even if this were not an issue, solving the problem in one shot, without exploiting the separable structure of the constraints, would lead to an optimization program of larger size, involving the decision variables and constraints of all agents, and possibly pose computational challenges.

To allow for a computationally tractable solution, while accounting for information privacy, we adopt a decentralized perspective, where agents cooperate to obtain an optimal solution of the centralized problem. We follow an iterative algorithm, where at every iteration each agent solves a local optimization problem with respect to its own local decision vector using the tentative solutions computed by the other agents at the previous iteration. Agents then exchange with each other their new tentative solutions, or broadcast them to some central authority that sends an update to each agent; the process is repeated on the basis of the received information.

Algorithms for the decentralized solution to convex optimization problems with separable constraints can be found in [1], [2], and references therein. Two main algorithmic directions can be distinguished, both of them relying on an iterative process. The first one is based on each agent performing at every iteration a local gradient descent step, while keeping the decision variables of all other agents fixed to the values communicated at the previous iteration [3–5]. Under certain structural assumptions (differentiability of the objective function and Lipschitz continuity of its gradient), it is shown that this scheme converges to some minimizer of the centralized problem, for an appropriate gradient step-size. The second direction involves mainly the so called Jacobi algorithm, which serves as an alternative to gradient algorithms. In this framework also the Gauss-Seidel algorithm which however is not of parallelizable nature unless a coloring scheme is adopted (see [1]), and block coordinate descent methods [6] can be considered. Under this set-up, at every iteration, instead of performing a gradient step, each agent minimizes the common objective function subject to local constraints, while keeping the decision vectors of all other agents fixed to their values at the previous iteration. It is shown that the Jacobi algorithm converges under certain contractiveness requirements, that are typically satisfied only if the objective function is jointly strictly convex with respect to the decision vectors of all agents.

An alternative research direction with a notable research activity involves a non-cooperative treatment of the problem, using tools from mean-field and aggregative game theory. A complete theoretical characterization for the stochastic, continuous-time variant of the problem, but in the absence of constraints, is provided in [7], [8]. The deterministic, discrete-time problem variant, accounting for the presence of separable constraints, is treated using fixed-point theoretic tools in [9]. In all cases, the considered algorithm is shown to converge not to a minimizer, but to a Nash equilibrium of a related game, in the limiting case where the number of agents tends to infinity. Several applications in this context have been provided, e.g., optimal power flow type of problems [10], optimal charging of electric vehicles [11], [12], etc.

In this paper we adopt a cooperative point of view, and construct a Jacobi-like algorithm. In contrast to the standard Jacobi algorithm, the local minimization that each agent

Research was supported by the European Commission, H2020, under the project UnCoVerCPS, grant number 643921.

L. Deori and M. Prandini are with the Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, e-mail: {luca.deori, maria.prandini}@polimi.it

K. Margellos is with the Department of Engineering Science, University of Oxford, Parks Road, OX1 3PJ, Oxford, United Kingdom, e-mail: kostas.margellos@eng.ox.ac.uk

solves at every iteration of the algorithm includes also an inertial term that encompasses the solution of that agent at the previous iteration. Our contributions can be summarized as follows: 1) We establish an equivalence between the set of minimizers of the problem under study and the set of fixed-points of the mapping induced by our algorithm, for any convex objective function. 2) For the case where the objective function is quadratic we show that our algorithm converges to some minimizer of the centralized problem, thus constituting an alternative to gradient methods, and without requiring strict convexity of the objective function as in the standard Jacobi algorithm. This result extends the equivalence between proximal operators and gradient algorithms observed in [2] for the single-agent case, to the multi-agent setting. 3) We apply the proposed algorithm to the problem of optimal charging of electric vehicles, extending the results of [9], [11], [12], and achieving convergence to an optimal charging scheme with a finite number of vehicles.

Notation

For any $a \in \mathbb{R}^n$, $\|a\|$ denotes the Euclidean norm of a, and $\|a\|_Q$ denotes the Q-weighted Euclidean norm of a. For a vector a, we denote by a_i the i-th block component of a, whereas a_{-i} denotes the vector emanating from a by removing a_i. Similarly for matrix A, $A_{i,i}$ denotes the i-th diagonal block of A, whereas $A_{-i,i}$ denotes the matrix composed by the i-th block column of A and all, but the i-th block row. For a continuously differentiable function $J(\cdot) : \mathbb{R}^n \to \mathbb{R}$, $\nabla J(a)$ is the gradient of $J(\cdot)$ evaluated at $a \in \mathbb{R}^n$, and $\nabla_i J(a)$ is its i-th component, $i = 1, \ldots, n$. $[a]_Q^U$ denotes the projection of a vector a on the set U with respect to the Q-weighted Euclidean norm. $\mathbf{1}_{n \times m}$ denotes the matrix with all entries equal to 1 with dimension $n \times m$, and I_c denotes the identity matrix with appropriate dimension, multiplied by the scalar $c \in \mathbb{R}$.

II. Problem statement

Consider the following optimization problem

$$\mathcal{P} : \min_{\{u_i \in \mathbb{R}^{n_i}\}_{i=1}^m} J(u_1, \ldots, u_m) \tag{1}$$

$$\text{subject to} \quad u_i \in U_i, \text{ for all } i = 1, \ldots, m, \tag{2}$$

where $J(\cdot, \ldots, \cdot) : \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_m} \to \mathbb{R}$, and $U_i \subseteq \mathbb{R}^{n_i}$, for all $i = 1, \ldots, m$. Let $n = \sum_{i=1}^m n_i$ and $U = U_1 \times \ldots \times U_m$. We impose the following assumption throughout the paper.

Assumption 1: The function $J(\cdot, \ldots, \cdot) : \mathbb{R}^{n_1} \times \ldots \times \mathbb{R}^{n_m} \to \mathbb{R}$ is continuously differentiable, and jointly convex with respect to all arguments. Moreover, the sets $U_i \subseteq \mathbb{R}^{n_i}$, $i = 1, \ldots, m$, are non-empty, compact and convex.

Under Assumption 1, by the Weierstrass' theorem (Proposition A8, p. 625 in [1]), $\mathcal{P}$ admits at least one optimal solution. Note, however, that $\mathcal{P}$ does not necessarily admit a unique minimizer. With a slight abuse of notation, for each i, $i = 1, \ldots, m$, let $J(\cdot, u_{-i}) : \mathbb{R}^{n_i} \to \mathbb{R}$ be the objective function in (1) as a function of the decision vector u_i of agent i, when the decision vectors of all other agents are fixed to $u_{-i} \in \mathbb{R}^{n-n_i}$. We will occasionally also write $J(u)$ instead of $J(u_1, \ldots, u_m)$, for $u = (u_1, \ldots, u_m)$, the interpretation will always be clear from the context. Problem $\mathcal{P}$ can be thought of as a multi-agent problem, where agents have a local decision vector u_i and a local constraint set U_i, and cooperate to determine a minimizer of J, which couples the individual decision vectors. Motivated by the particular structure of $\mathcal{P}$ with separable constraint sets, we follow a decentralized, iterative approach described in Algorithm 1. This allows to cope with privacy and computational issues. Initially, each agent i, $i = 1, \ldots, m$, starts with some tentative value $u_i(0) \in U_i$, such that $\big(u_1(0), \ldots, u_m(0)\big)$ is feasible and constitutes an estimate of what the minimizer of $\mathcal{P}$ might be (step 3, Algorithm 1). At iteration $k+1$, each agent i receives the values of all other agents $u_{-i}(k)$ (step 5, Algorithm 1) from the central authority, and updates its estimate by averaging with weight $\lambda \in (0,1)$ the previous estimate and the solution of a local minimization problem (step 6, Algorithm 1). The performance criterion in this local problem is a linear combination of the objective $J(z_i, u_{-i}(k))$, where the variables of all other agents apart from the i-th one are fixed to their values at iteration k, and a quadratic term, penalizing the difference between the decision variables and the value of agent's i own variable at iteration k, i.e., $u_i(k)$. The relative importance of these two terms is dictated by $c \in \mathbb{R}_+$; we defer the discussion on the importance of the penalty term until Section IV. Note that under Assumption 1, and due to the presence of the quadratic penalty term, the resulting problem is strictly convex with respect to z_i, and hence admits a unique minimizer.

Algorithm 1 Decentralized algorithm

```
1: **Initialization**
2:   $k = 0$.
3:   Consider $u_i(0) \in U_i$, for all $i = 1, \ldots, m$.
4: **For** $i = 1, \ldots, m$ **repeat until convergence**
5:   Agent $i$ receives $u_{-i}(k)$ from central authority.
6:   $u_i(k+1) = \lambda u_i(k)$
     $+(1-\lambda)\mathrm{argmin}_{z_i \in U_i}\big\{J(z_i, u_{-i}(k)) + c\|z_i - u_i(k)\|^2\big\}$.
7:   $k \leftarrow k + 1$.
```

III. Preliminary definitions and results

A. Definitions

1) Minimizers: By (1)-(2), the set of minimizers of $\mathcal{P}$ is

$$M = \big\{u \in U : \; u \in \arg \min_{\{z_i \in U_i\}_{i=1}^m} J(z_1, \ldots, z_m)\big\}. \tag{3}$$

Following the discussion below Assumption 1, M is non-empty. Note that set of optimizers of M is not necessarily a singleton; this will be the case if J is jointly strictly convex with respect to its arguments.

2) Fixed-points: For each i, $i = 1, \ldots, m$, consider the mappings $T_i(\cdot) : U \to U_i$ and $\widetilde{T}_i(\cdot) : U \to U_i$, defined such that, for any $u = (u_1, \ldots, u_m) \in U$,

$$T_i(u) = \arg \min_{z_i \in U_i} \|z_i - u_i\|^2 \tag{4}$$
$$\text{subject to} \quad J(z_i, u_{-i}) \leq \min_{\zeta_i \in U_i} J(\zeta_i, u_{-i})$$

$$\widetilde{T}_i(u) = \arg \min_{z_i \in U_i} \big\{J(z_i, u_{-i}) + c\|z_i - u_i\|^2\big\}. \tag{5}$$

The mapping in (4) serves as a tie-break rule to select, in case $J(\cdot, u_{-i})$ admits multiple minimizers, the one closer to

u_i with respect to the Euclidean norm. Note that both the minimizers of (4) and (5) are unique, so that both mappings are well defined. Note also that with $u(k)$ in place of u, (5) implies that the update step 6 in Algorithm 1 can be equivalently written as $u_i(k+1) = \lambda u_i(k) + (1-\lambda)\widetilde{T}_i(u(k))$. Define also the mappings $T(\cdot) : U \to U$ and $\widetilde{T}(\cdot) : U \to U$, such that their components are given by $T_i(\cdot)$ and $\widetilde{T}_i(\cdot)$, respectively, for $i = 1, \ldots, m$, i.e., $T(\cdot) = \big(T_1(\cdot), \ldots, T_m(\cdot)\big)$ and $\widetilde{T}(\cdot) = \big(\widetilde{T}_1(\cdot), \ldots, \widetilde{T}_m(\cdot)\big)$. The mappings $T(\cdot)$ and $\widetilde{T}(\cdot)$ can be equivalently written as

$$
\begin{aligned}
T(u) = &\arg\min_{z\in U} \sum_{i=1}^{m} \|z_i - u_i\|^2 \qquad (6)\\
&\text{subject to } J(z_i, u_{-i}) \leq \min_{\zeta_i \in U_i} J(\zeta_i, u_{-i}),\ \forall i = 1, \ldots, m\\
\widetilde{T}(u) = &\arg\min_{z\in U} \sum_{i=1}^{m} \left\{ J(z_i, u_{-i}) + c\|z_i - u_i\|^2 \right\}, \qquad (7)
\end{aligned}
$$

where the terms inside the summations are decoupled. The set of fixed-points of $T(\cdot)$ and $\widetilde{T}(\cdot)$ is, respectively, given by $F_T = \big\{u \in U : u = T(u)\big\}$, and $F_{\widetilde{T}} = \big\{u \in U : u = \widetilde{T}(u)\big\}$.

B. Connections between minimizers and fixed-points

We report here a fundamental optimality result (e.g., see Proposition 3.1 in [1]), that we will often use in the sequel.

Proposition 1 (Proposition 3.1 in [1]): Consider any $n \in \mathbb{N}_+$, and assume that $J(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, and $U \subseteq \mathbb{R}^n$ is non-empty, closed and convex. It holds that i) if $u \in U$ minimizes $J(\cdot)$ over U, then $(z-u)^\top \nabla J(u) \geq 0$, for all $z \in U$; ii) if $J(\cdot)$ is also convex on U, then the condition of the previous part is also sufficient for u to minimize $J(\cdot)$ over U, i.e., $u \in \arg\min_{z\in U} J(z)$.

The following propositions show that the set of minimizers M of $\mathcal{P}$ in (3) and the set of fixed-points F_T of the mapping T in (6) coincide, and that the set of fixed-points F_T of $T(\cdot)$ and the set of fixed-points $F_{\widetilde{T}}$ of $\widetilde{T}(\cdot)$ coincide.

Proposition 2: Under Assumption 1, $M = F_T$.

Proposition 3: Under Assumption 1, $F_T = F_{\widetilde{T}}$.

Proof: The proofs are omitted due to space limitation, they can be found in [13, Proposition 2 and 3]. ■

Note that the connection between minimizers and fixed-points, similar to the ones in Proposition 2, has been also investigated in [14], in the context of Nash equilibria in non-cooperative games.

IV. Main convergence result

In this section we strengthen Assumption 1, and focus on convex optimization problems with a convex, quadratic objective function.

Assumption 2: For any $u \in U$, $J(u) = u^\top Q u + q^\top u$, where $Q \succeq 0$ and $q \in \mathbb{R}^n$.

Note that Q can be assumed to be symmetric (i.e., $Q = Q^\top$) without loss of generality.Moreover if additional terms that depend on the local decision vectors u_i, $i = 1, \ldots, m$, and encode the utility function of each agent were present in the objective function, they could be incorporated in the local constraint set U_i, $i = 1, \ldots, m$, by means of an epigraphic reformulation, thus bringing the cost back to be quadratic.

Under Assumption 2, the mapping $\widetilde{T}(\cdot)$ in (7) is given by

$$
\begin{aligned}
\widetilde{T}(u) &= \arg\min_{z\in U} \sum_{i=1}^{m} J(z_i, u_{-i}) + c\|z_i - u_i\|^2 \qquad (8)\\
&= \arg\min_{z\in U} \sum_{i=1}^{m} z_i^\top (Q_{i,i} + I_c) z_i + (2u_{-i}^\top Q_{-i,i} - 2u_i^\top I_c + q_i^\top) z_i\\
&= \arg\min_{z\in U} z^\top (Q_d + I_c) z + (2u^\top Q_z - 2u^\top I_c + q^\top) z,
\end{aligned}
$$

where, for all $i = 1, \ldots, m$, $Q_{i,i}$ is the i-th block of Q, corresponding to the decision vector z_i, Q_d is a block diagonal matrix whose i-th block is $Q_{i,i}$, and $Q_z = Q - Q_d$. Notice the slight abuse of notation in (8), where the weighted identity matrix I_c in the second and the third equality are not of the same dimension. Let $\xi(u) = (Q_d + I_c)^{-1}(I_c u - Q_z u - q/2)$ denote the unconstrained minimizer of (8). Then

$$
\widetilde{T}(u) = \arg\min_{z\in U} (z - \xi)^\top (Q_d + I_c)(z - \xi) \quad = [\xi(u)]^U_{Q_d + I_c}.
$$

Note that $Q_d + I_c$ is always positive definite for $c \in \mathbb{R}_+$, so that the projection $[\xi(u)]^U_{Q_d+I_c}$ is well defined.

In the next proposition the non-expansive property of the mapping $\widetilde{T}(\cdot)$ is proven. This property will be exploited in Theorem 1 to establish convergence of Algorithm 1.

Proposition 4: Consider Assumptions 1 and 2. If

$$
\begin{bmatrix} 2Q & Q \\ Q & Q_d + I_c \end{bmatrix} \succeq 0, \qquad (9)
$$

the mapping $\widetilde{T}(u) = [\xi(u)]^U_{Q_d+I_c}$ is non-expansive with respect to $\|\cdot\|_{Q_d+I_c}$, namely $\|\widetilde{T}(u) - \widetilde{T}(v)\|_{Q_d+I_c} \leq \|u - v\|_{Q_d+I_c}$, for all $u, v \in U$.

Proof: Any projection mapping is non-expansive (see Proposition 3.2 in [1]). Therefore, we have that

$$
\begin{aligned}
\|\widetilde{T}(u) - \widetilde{T}(v)\|_{Q_d+I_c} &= \|\, [\xi(u)]^U_{Q_d+I_c} - [\xi(v)]^U_{Q_d+I_c} \,\|_{Q_d+I_c}\\
&\leq \|\xi(u) - \xi(v)\|_{Q_d+I_c}. \qquad (10)
\end{aligned}
$$

We will show that, if (9) holds,

$$
\|\xi(u) - \xi(v)\|_{Q_d+I_c} \leq \|u - v\|_{Q_d+I_c}, \qquad (11)
$$

proving that the mapping $\widetilde{T}(u)$ is non-expansive. Replacing in (11) the expression of ξ, and raising to the square yields

$$
\begin{aligned}
&\|(Q_d + I_c)^{-1}(I_c - Q_z)(u - v)\|^2_{Q_d+I_c} \leq \|u - v\|^2_{Q_d+I_c} \Leftrightarrow\\
&\|(I - (Q_d + I_c)^{-1}Q)(u - v)\|^2_{Q_d+I_c} \leq \|u - v\|^2_{Q_d+I_c} \qquad (12)
\end{aligned}
$$

where I is an identity matrix of appropriate dimension, and the equivalence follows from the fact that $Q_z = Q - Q_d$. By bringing both terms in (12) in the left-hand side and by the definition of $||\cdot||_{Q_d+I_c}$, (12) is satisfied if

$$
\begin{aligned}
&(I - (Q_d + I_c)^{-1}Q)^\top (Q_d + I_c)(I - (Q_d + I_c)^{-1}Q)\\
&\quad - (Q_d + I_c) \preceq 0 \Leftrightarrow 2Q - Q(Q_d + I_c)^{-1}Q \succeq 0. \qquad (13)
\end{aligned}
$$

where the last inequality follows after some algebraic calculations and the fact that Q and $Q_d + I_c$ are symmetric. Equation (13) can be rewritten by means of Schur's complement finally obtaining (9). ■

Notice that if $\bar{c}$ satisfies (9), then any $c \geq \bar{c}$ satisfies (9) as well. To see this, take any $c \geq \bar{c}$ and rewrite (9) as

$$\begin{bmatrix} 2Q & Q \\ Q & Q_d + I_{\bar{c}} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & I_{\tilde{c}} \end{bmatrix} \succeq 0, \tag{14}$$

where $\tilde{c} = c - \bar{c}$. The matrices in (14) are both positive semi-definite and, hence, their sum is also positive semi-definite. Condition (9) can be easily checked by means of standard LMI solvers. In fact, it can be shown that for any $Q \succeq 0$ there exists c such that (9) is satisfied. Indeed condition, (9) can be equivalently written as

$$\begin{bmatrix} u^\top & v^\top \end{bmatrix} \begin{bmatrix} 2Q & Q \\ Q & Q_d + I_c \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} \geq 0, \text{ for all } u, v \in \mathbb{R}^n$$
$$\Leftrightarrow u^\top 2Qu + v^\top (Q_d + I_c)v + 2v^\top Qu \geq 0. \tag{15}$$

From the fact that $(u + v)^\top Q(u + v) \geq 0$ it follows that $2v^\top Qu \geq -u^\top Qu - v^\top Qv$. Replacing the latter in (15), the following sufficient condition to satisfy (9) is obtained.

$$u^\top Qu + v^\top (Q_d + I_c - Q)v \geq 0, \text{ for all } u, v \in \mathbb{R}^n. \tag{16}$$

The first term in (16) is non-negative because $Q \succeq 0$, while the second term can be made non-negative exploiting the matrix I_c to move the eigenvalues of $Q_d - Q$ by c. Indeed, letting $\lambda_{I_c + Q_d - Q}$ and $\lambda_{Q_d - Q}$ denote the eigenvalues of $I_c + Q_d - Q$ and $Q_d - Q$, respectively, we have

$$\lambda_{I_c + Q_d - Q} = \lambda_{Q_d - Q} + c. \tag{17}$$

Hence, (16) can be satisfied by choosing c such that $c \geq \lambda^{\max}_{Q - Q_d}$, where $\lambda^{\max}_{Q - Q_d}$ denotes the maximum eigenvalue of matrix $Q - Q_d$. Note that, since $Q - Q_d$ is symmetric with zero trace, its eigenvalues will be real and at least one should be non-negative. As a result, $c \geq \lambda^{\max}_{Q - Q_d} \geq 0$.

Theorem 1: Consider Assumptions 1 and 2. If $c \in \mathbb{R}_+$ is chosen so that (9) is satisfied, then Algorithm 1 converges to a minimizer of $\mathcal{P}$.

Proof: Step 6 of Algorithm 1 corresponds to the so called Krasnoselskij iteration [15] (referred to as averaged operator in [2]),

$$u(k + 1) = \lambda u(k) + (1 - \lambda)\widetilde{T}(u(k)), \tag{18}$$

of the mapping $\widetilde{T}(\cdot)$, which, according to Proposition 4, is non-expansive if (9) is satisfied. By Theorem 3.2 in [15], we have that for any non-expansive mapping $\widetilde{T}(u) : U \to U$, with U compact and convex, the Krasnoselskij iteration (18) converges to a fixed-point of $\widetilde{T}(\cdot)$ for any $\lambda \in (0, 1)$, and for any initial condition $u(0) \in U$. Under Assumptions 1 and 2, the mapping $\widetilde{T}(\cdot)$ defined in (8), satisfies the aforementioned requirements, hence, Algorithm 1 leads to a fixed-point of $\widetilde{T}(\cdot)$. By Propositions 2 and 3, this fixed-point will also be a minimizer of $\mathcal{P}$. ■

It should be noted that the condition on c in Proposition 4 is related to the requirement imposed in [1] (Proposition 3.4, p. 214) on the step-size of a gradient based approach. This is due to the fact that in the case where $J(\cdot)$ satisfies Assumption 2, step 6 of Algorithm 1 can be shown to be equivalent to a scaled gradient projection algorithm (see Section 3.3.3 in [1]), with $1/c$ playing the role of the gradient step-size and with $(Q_d + I_c)^{-1}$ (inverse of the Hessian of the objective function in step 6 of Algorithm 1) being the scaling matrix. For such algorithms convergence results exist for an appropriate step-size, which is in turn related to c. In particular, the step-size for which convergence is guaranteed is related to the Lipschitz constant of the gradient of the objective function, and, under Assumption 2, it can be shown to be equivalent with choosing c so that (9) is satisfied.

Hence, in the case of a convex quadratic objective function, both the proposed algorithm and a gradient based approach are applicable. Our analysis, however, not only complements the scaled gradient projection algorithm by constituting its Jacobi-like equivalent without requiring strict convexity of the objective function as it is usually the case for the standard Jacobi algorithm, but also follows a different analysis from that in [1], motivated by the fixed-point theoretic results of [9]. Moreover, the results of Section II are valid for any convex objective function, not necessarily quadratic, thus opening the road for extending the convergence results of Section III by relaxing Assumption 2; see [13] for preliminary results.

A. Alternative implementations

We investigate the convergence properties of Algorithm 1 when the so called Krasnoselskij iteration (step 6 of Algorithm 1) is replaced by the simpler Picard-Banach iteration:

$$u(k + 1) = \widetilde{T}(u(k)). \tag{19}$$

Note that this corresponds to setting $\lambda = 0$ in step 6 of Algorithm 1. As in the proof of Theorem 1, once convergence is proven, then it is easily shown that the solution is a minimizer of $\mathcal{P}$ due to Propositions 2 and 3.

For Algorithm 1 to converge when step 6 is replaced by (19), the mapping $\widetilde{T}(\cdot)$ has to be either contractive or firmly non-expansive [15]. We first investigate conditions under which $\widetilde{T}(\cdot)$ is contractive. Following a reasoning similar to the one in the proof of Proposition 4, it can be seen that if

$$\begin{bmatrix} 2Q - (1 - \alpha^2)(Q_d + I_c) & Q \\ Q & Q_d + I_c \end{bmatrix} \succeq 0, \tag{20}$$

is satisfied for some $\alpha \in [0, 1)$, then $\|\widetilde{T}(u) - \widetilde{T}(v)\|_{Q_d + I_c} \leq \alpha \|u - v\|_{Q_d + I_c}$, for all $u, v \in U$, which in turn implies that the mapping $\widetilde{T}(\cdot)$ is contractive with respect to $|| \cdot ||_{Q_d + I_c}$ [15]. Condition (20), however, can not be always satisfied by appropriately choosing c; indeed, it is necessary that Q is positive definite for (20) to be satisfied. The latter is equivalent to requiring that the objective function in Assumption 2 is strictly convex.

We now investigate conditions under which $\widetilde{T}(\cdot)$ is firmly non-expansive. Motivated by the analysis of [9], we have that

$$\begin{aligned} &\|\widetilde{T}(u) - \widetilde{T}(v)\|^2_{Q_d + I_c} = \| \, [\xi(u)]^U_{Q_d + I_c} - [\xi(v)]^U_{Q_d + I_c} \, \|^2_{Q_d + I_c} \\ &\leq (\xi(u) - \xi(v))^\top (Q_d + I_c)([\xi(u)]^U_{Q_d + I_c} - [\xi(v)]^U_{Q_d + I_c}) \\ &= (u - v)^\top (I - Q(Q_d + I_c)^{-1})(Q_d + I_c)([\xi(u)]^U_{Q_d + I_c} - [\xi(v)]^U_{Q_d + I_c}) \\ &= (u - v)^\top (Q_d + I_c - Q)([\xi(u)]^U_{Q_d + I_c} - [\xi(v)]^U_{Q_d + I_c}) \end{aligned} \tag{21}$$

where the first inequality follows from the definition of a firmly non-expansive mapping and the fact that any projection mapping is firmly non-expansive (see Proposition 4.8 in

[16]). The second equality is due to the definition $\xi(\cdot)$, and the last one follows after performing computation.

Since $Q \succeq 0$, then $\|\widetilde{T}(u) - \widetilde{T}(v)\|^2_{Q_d+I_c-Q} \leq \|\widetilde{T}(u) - \widetilde{T}(v)\|^2_{Q_d+I_c}$. This, together with (21), implies that

$$\begin{aligned} &\| \, [\xi(u)]^U_{Q_d+I_c} - [\xi(v)]^U_{Q_d+I_c} \, \|^2_{Q_d+I_c-Q} \\ &\leq (u-v)^\top (Q_d + I_c - Q)([\xi(u)]^U_{Q_d+I_c} - [\xi(v)]^U_{Q_d+I_c}). \end{aligned} \tag{22}$$

By the definition of a firmly non-expansive mapping [15], (22) implies that, if $Q_d + I_c - Q \succ 0$, $\widetilde{T}(\cdot)$ is firmly non-expansive with respect to $|| \cdot ||_{Q_d+I_c-Q}$. The condition $Q_d + I_c - Q \succ 0$ can be satisfied by choosing c as in (15)-(17), rendering (19) an alternative to step 6 of Algorithm 1.

B. Information exchange

To implement Algorithm (1), at iteration $k+1$, it is needed that some central authority, or common processing node, collects and broadcasts the tentative solution of each agent to all others, and that the agents have knowledge of the common objective function $J(\cdot)$ so that each of them can compute $J(\cdot, u_{-i}(k))$ (alternatively the central authority can broadcast it to each agent i, $i = 1, \ldots, m$). However, the required amount of information that needs to be exchanged can be significantly reduced when the objective function exhibits some particular structure. This is the case, e.g., for objective functions that are coupled only through the average of some variables. The central authority needs then to collect the solutions of all agents, but it only has to broadcast the average value. Each agent will then be able to compute $J(\cdot, u_{-i}(k))$ by subtracting from the average the value of its local decision vector at iteration k, i.e., $u_i(k)$.

V. Optimal charging of electric vehicles

We consider the problem of optimizing the charging strategy for a fleet of m plug-in electric vehicles (PEVs) over a finite horizon T. We follow the formulation of [9], [11], [12]; but, our algorithm converges to a minimizer of the centralized problem counterpart and with a finite number of agents/vehicles, as opposed to the aforementioned references, where convergence to a Nash equilibrium at the limiting case of an infinite population of agents is established. The PEV charging problem is formulated as follows:

$$\min_{\left\{\{u_{i,t}\}_{i=1}^m\right\}_{t=0}^T} \frac{1}{m} \sum_{t=0}^{T} p_t (d_t + \sum_{i=1}^{m} u_{i,t})^2 \tag{23}$$

$$\text{subject to} \left\{ \begin{array}{l} \sum_{t=0}^T u_{i,t} = \gamma_i, \forall i{=}1,\ldots,m \\ \underline{u}_{i,t} \leq u_{i,t} \leq \overline{u}_{i,t}, \forall i{=}1,\ldots,m, \; t{=}0,\ldots,T, \end{array} \right.$$

where $p_t \in \mathbb{R}$ is an electricity price coefficient at time t, $d_t \in \mathbb{R}$ represents the non-PEV demand at time t, $u_{i,t} \in \mathbb{R}$ is the charging rate of vehicle i at time t, $\gamma_i \in \mathbb{R}$ represents a prescribed charging level to be reached by each vehicle i at the end of the considered time horizon, and $\underline{u}_{i,t}, \overline{u}_{i,t} \in \mathbb{R}$ are bounds on the minimum and maximum value of $u_{i,t}$, respectively. The objective function in (23) encodes the total electricity cost given by the demand (both PEVs and non-PEVs) multiplied by the price of electricity, which in turn depends linearly on the total demand through p_t, thus giving rise to the quadratic function in (23). This linear dependency of price with respect to the total demand models the fact that agents/vehicles are price anticipating authorities, anticipating their consumption to have an effect on the electricity price (see [17]). Problem (23) can be written in compact form as

$$\begin{aligned} &\min_{u \in \mathbb{R}^{m(T+1)}} (d + Au)^\top P (d + Au) \\ &\text{subject to } u_i \in U_i, \text{ for all } i = 1, \ldots, m, \end{aligned} \tag{24}$$

where $P = (1/m)\mathrm{diag}(p) \in \mathbb{R}^{(T+1)\times(T+1)}$, and $\mathrm{diag}(p)$ is a matrix with $p = (p_0, \ldots, p_T) \in \mathbb{R}^{T+1}$ on its diagonal. $A = \mathbf{1}_{1\times m} \otimes I \in \mathbb{R}^{(T+1)\times m}$, where $\otimes$ denotes the Kronecker product. Moreover, $d = (d_0, \ldots, d_T) \in \mathbb{R}^{T+1}$, $u = (u_1, \ldots, u_m) \in \mathbb{R}^{m(T+1)}$ with $u_i = (u_{i,0}, \ldots, u_{i,T}) \in \mathbb{R}^{T+1}$, and U_i encodes the constraints of each vehicle i, $i = 1, \ldots, m$, in (23). Problem (24) can be solved in a decentralized fashion by means of Algorithm 1. We compute the value of c so that (9) is satisfied and the mapping $\widetilde{T}(\cdot)$ associated to problem (24) is non-expansive. Note that the objective function in (24) is not strictly convex as $A^\top P A = \mathbf{1}_{m\times m} \otimes P$, and it exhibits a structure that allows for reduced information exchange as described in Section IV-B. Indeed, at iteration $k+1$ of Algorithm 1, the central authority needs to collect the solution of each agent but it only has to broadcast $V(k) = d + Au(k)$. Each agent i, can then compute its tentative objective as $J(z_i, u_{-i}(k)) = (V(k) - u_i(k) + z_i)^\top P (V(k) - u_i(k) + z_i)$. Step 6 in Algorithm 1 for problem (24) reduces then to

$$\begin{aligned} &u_i(k+1) = \lambda u_i(k) + (1-\lambda)\widetilde{T}(u(k)) = \\ &\lambda u_i(k) {+} (1{-}\lambda) \operatorname*{argmin}_{z_i \in U_i} (V(k){-}u_i(k){+}z_i)^\top P (V(k){-}u_i(k){+}z_i). \end{aligned}$$

A. Simulation results

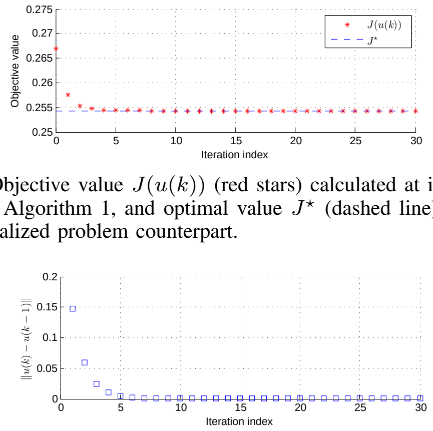
We consider first a fleet of $m = 100$ PEVs, each of them having to reach a different level of charge $\gamma_i \in [0.1, 0.3]$, $i = 1, \ldots, m$, at the end of a time horizon $T = 24$, corresponding to hourly steps. The bounds on $u_{i,t}$ are taken to be $\underline{u}_{i,t} = 0$ and $\overline{u}_{i,t} = 0.02$, for all $i = 1, \ldots, m$, $t = 0, \ldots, T$. The non-PEV demand profile is retrieved from [11], whereas the price coefficient is $p_t = 0.15$, $t = 0, \ldots, T$. Note that, as in [12], $u_{i,t}$ corresponds to normalized charging rate, which is then rescaled to be turned into reasonable power values. All optimization problems are solved using CPLEX, [18].

For comparison purposes, problem (24) is solved first in a centralized fashion, achieving an optimal objective value $J^\star = 2.67$. It is then solved in a decentralized fashion by means of Algorithm 1, setting $c = 0.1$ and $\lambda = 0.4$. Note that for (9) to be satisfied, according to the analysis of Section IV, we should have $c \geq 0.0735$. After 30 iterations the difference between the decentralized and the centralized objective is $J(u(30)) - J^\star = 1.36 \cdot 10^{-6}$, thus achieving numerical convergence.

We perform a parametric analysis, running Algorithm 1 for different values of λ and c. In Table I the number of iterations needed to achieve a relative error between the decentralized and the centralized objective value $\frac{J(u(k))-J^\star}{J^\star} < 10^{-6}$ is reported. Note that if we choose a value of c that does not satisfy (9), Algorithm 1 does not always converge (see first row of Table I, for $\lambda = 0.01, 0.1$). As c and λ increase, numerical convergence requires more iterations,

TABLE I: Number of iterations needed for $\frac{J(u(k))-J^\star}{J^\star} < 10^{-6}$, for different values of λ and c.

	$\lambda = 0.01$	$\lambda = 0.1$	$\lambda = 0.4$	$\lambda = 0.6$	$\lambda = 0.9$
$c = 0.05$	-	-	11	19	91
$c = 0.0735$	13	9	13	22	98
$c = 0.1$	13	15	23	34	137
$c = 0.15$	25	27	41	63	252
$c = 0.2$	34	38	57	86	348



(a) Objective value $J(u(k))$ (red stars) calculated at iteration k of Algorithm 1, and optimal value $J^\star$ (dashed line) of the centralized problem counterpart.

(b) Iteration error $\|u(k-1) - u(k)\|$ (blue squares).

Fig. 1: Objective value and iteration error for $m = 1000$.

with the exception of $c = 0.0735$, where $\lambda = 0.01$ does not correspond to the faster convergence behavior. This case corresponds to the minimum value of c for which (9) is satisfied, thus being likely to perform numerically unstable, and for λ close to zero, as also revealed by the analysis of Section III-B (see below (20)), the algorithm tends to be non-convergent.

We consider now a fleet of $m = 1000$ PEVs, and modify the required charging level such that $\gamma_i \in [0.005, 0.025]$, for all $i = 1, \dots, m$, and set the bounds on $u_{i,t}$ to $\underline{u}_{i,t} = 0$ and $\overline{u}_{i,t} = 0.0025$, for all $i = 1, \dots, m$, $t = 0, \dots, T$. All the other parameters are left unchanged with respect to the previous set-up. In Figure 1a the objective value $J(u(k))$ achieved at iteration k of Algorithm 1 is depicted, whereas Figure 1b shows $\|u(k) - u(k-1)\|$. After 30 iterations the difference between the decentralized and the centralized objective is $J(u(30)) - J^\star = 1.36 \cdot 10^{-6}$, thus achieving numerical convergence. Figure 2 depicts the PEV, non-PEV, and total demand along the considered time horizon. The PEV demand is optimized so as to fill the over-night valley of the non-PEV demand.

VI. Concluding remarks

In this paper, a decentralized algorithm for multi-agent, convex optimization programs with a common objective and subject to separable constraints, was developed. For the case where the objective function is quadratic, the proposed scheme was shown to converge to some minimizer of the centralized problem, and its performance was illustrated by applying it to the problem of optimal charging of electric vehicles. Current work concentrates on relaxing Assumption 2, extending our convergence results to the case of continuously differentiable, convex, but not necessarily quadratic, objective functions (see [13]). Moreover, we aim at adapting the analysis of this paper to the non-cooperative set-up, establishing convergence to a Nash equilibrium by using a finite number of agents.

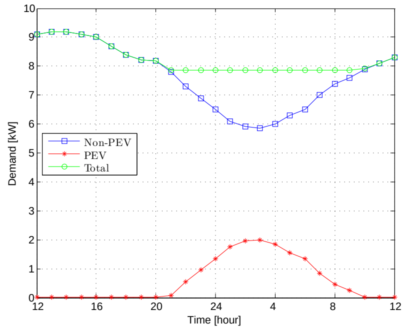


Fig. 2: Demand along an one day time horizon for $m = 1000$ vehicles: non-PEV demand (blue squares), PEV demand computed via Algorithm 1 (red stars), and total demand (green circles).

References

[1] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Athena Scientific (republished in 1997), 1989.

[2] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 123 – 231, 2013.

[3] A. Goldstein, "Convex programming in Hilbert space," *Bulletin of American Mathematical Society*, vol. 70, 1964.

[4] E. Levitin and B. Poljak, "Costrained minimization methods," *USSR Computational Mathematics and Physics (English transaltion)*, 1965.

[5] D. Bertsekas, "On the Goldstein-Levitin-Polyak gradient projection method," *IEEE Transactions on Automatic Control*, vol. 2, 1976.

[6] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran, "Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality," *Math. Oper. Res.*, no. 35, 2010.

[7] M. Huang, P. Caines, and R. Malhame, "Large-population cost-coupled LQG problems with nonuniform agents: Individual-mass behavior and decentralized ε-Nash equilibria," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, 2007.

[8] J. Lasry and P. Lions, "Mean field games," *Japanese Journal of Mathematics*, vol. 2, 2007.

[9] S. Grammatico, F. Parise, M. Colombino, and J. Lygeros, "Decentralized convergence to Nash equilibria in constrined mean field control," *IEEE Transactions on Automatic Control*, 2016.

[10] F. Bagagiolo and D. Bauso, "Mean-field games and dynamic demand management in power grids," *Dynamic Games and Applications*, 2014.

[11] Z. Ma, D. Callaway, and I. Hiskens, "Decentralized charging control of large populations of plug-in electric vehicles," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 1, 2013.

[12] F. Parise, M. Colombino, S. Grammatico, and J. Lygeros, "Mean field constrained charging policy for large populations of plug-in electric vehicles," 2014.

[13] L. Deori, K. Margellos, and M. Prandini, "A regularized jacobi algorithm for multi-agent convex optimization problems with separable constraint sets," *https://arxiv.org/abs/1604.07814*.

[14] F. Facchinei, A. Fischer, and V. Piccialli, "On generalized Nash games and variational inequalities," *Operations Research Letters*, 2007.

[15] V. Berinde, *Iterative Approximation of Fixed Points*. Springer-Verlag Berlin Heidelberg, 2007.

[16] H. Bauschke and P. Combettes, *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, 2010.

[17] B. Gharesifard, T. Basar, and A. Dominguez-Garcia, "Price-based coordinated aggregation of networked distributed energy resources," *IEEE Transactions on Automatic Control*, 2016.

[18] "IBM ILOG CPLEX Optimizer," 2015. [Online]. Available: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html