Data privacy in multi-agent optimization under uncertainty

Kostas Margellos[a], Alessandro Falsone[b], Simone Garatti[b] and Maria Prandini[b]
[a]University of Oxford, Department of Engineering Science, Parks Road 17, OX2 6TE Oxford, United Kingdom
[b]Politecnico di Milano, Department of Electronics, Information and Bioengineering, Piazza Leonardo da Vinci 32, 20133 Milano, Italy

Chapter

Abstract

We consider multi-agent optimization problems over networks, where agents cooperate to reach agreement on a common decision, or on the share of a common resource. This gives rise to the class of so called decision coupled and constraint coupled problems, respectively. We investigate the setting where the environment within which agents operate is uncertain, affecting agents' objective functions and/or constraint sets. We provide a data driven framework to address this problem, however, we view data as a finite and private resource. This implies that each agent has access to a finite amount of data (e.g., scenarios/realizations of the uncertain phenomenon), and these data neither are common nor are shared among agents, but constitute a private resource. Using tools from statistical learning and randomized optimization based on the so called scenario approach, we show how to accompany decisions in such a private-data regime with probabilistic certificates of their robustness when it comes into uncertainty realizations different from those included in the data. Moreover, we show that the class of decision coupled and constraint coupled problems naturally fits in this framework, and hence discuss general architectures and privacy preserving algorithms for computing decisions that enjoy such robustness certificates in a distributed manner.

Keywords: data-driven optimization; distributed optimization; multi-agent systems; scenario approach; statistical learning; uncertain systems

Key points

- unifying privacy-oriented framework for multi-agent learning with finite sample complexity bounds
- multi-agent scenario approach with agent-private scenarios/data
- distributed algorithms for multi-agent learning in decision coupled optimization problems
- distributed algorithms for multi-agent learning in constraint coupled optimization problems

1 Introduction

Current systems are increasing in complexity as they involve multiple agents interacting with each other either physically or by means of message passing, trying to cooperative make optimal decisions to minimize some collective cost or maximize social welfare. This gives rise to multi-agent cooperative optimization over networks, and is encountered across different application domains, ranging from energy systems, Bolognani et al. (2015); Zhang and Giannakis (2016); Casagrande and Boem (2022), and wireless networks, Mateos and Giannakis (2012); Baingana et al. (2014), to robotics, Martínez et al. (2007).

Optimal decision making in multi-agent networks is a challenging task, as agents are often not willing to share information pertaining their local objective functions and/or local constraint sets with all other agents in the network due to privacy concerns. In most cases they are only willing to share their tentative decisions during the cooperative optimization process, thus masking information considered as private. To address this challenge, most of the literature in multi-agent cooperative optimization over networks focuses on the design of algorithms that are compatible with the networked structure of the system, distribute the computations among agents, and preserve privacy of local information. This is a broad area of research with intense activity; several algorithms exist each of them with different appealing characteristics according to the specific multi-agent problem class they tackle. However, most of these algorithms are inspired by the distributed architectures proposed in Tsitsiklis et al. (1986); Jadbabaie et al. (2003); Olfati-Saber and Murray (2004); Nedíc and Ozdaglar (2009); Nedíc et al. (2009, 2010); Zhu and Martínez (2012); Notarnicola and Notarstefano (2017); Falsone et al. (2017a); Krishnamoorthy et al. (2018). We refer also to Bertsekas and Tsitsiklis (1989); Boyd et al. (2010); Notarstefano et al. (2019) for a more detailed review of methods and their convergence properties.

Despite the advancements in distributed optimization to address these challenges, the problem has become more complex as in most realistic applications agents operate in uncertain environments, with uncertainty affecting their local constraint sets and objective functions, and hence the interim decisions they exchange. Distributed techniques to take uncertainty into account in a principled manner have appeared in recent years Towfic and Sayed (2014); Carlone et al. (2014); Chamanbaz et al. (2017); Lee and Nedic (2013, 2016); Sayin et al. (2017). However, the techniques proposed therein are tailored to specific algorithms and are hence not applicable to general multi-agent architectures. Specifically, the approaches in Towfic and Sayed (2014); Lee and Nedic (2016) require some regularity conditions on the agents' cost function; Sayin et al. (2017) and Lee and Nedic (2013) require to extract an infinite number of uncertainty realizations; the randomized algorithm of Carlone et al. (2014) requires to exchange constraints over a time-invariant communication network, whereas Chamanbaz et al. (2017) allows for time-varying communications but is confined to linear programs. Among these references, Carlone et al. (2014); Lee and Nedic (2016); Chamanbaz et al. (2017) adopt a data driven perspective. Despite being theoretically sound, uncertainty is not treated in a unified way, and the data requirements are often unrealistic for practical purposes.

In this chapter, we present a framework that tries to overcome the limitations in the existing algorithms for uncertain optimization using data, with data being treated as a centralized, i.e., common across agents, and infinite resource, as often an infinite number of uncertainty realizations is considered to be available. We view the problem of cooperative optimization over uncertain multi-agent networks under a data driven lens, where uncertainty is represented by means of data. Data can be thought of as scenarios/realizations of the uncertainty, and could be available in the form of historical datasets or could be synthetic, generated by means of a probabilistic model. This viewpoint prevents us from making assumptions on the underlying distribution of the uncertainty and/or the geometry of its support set.

Our framework has the following key features:

1. *Decisions with prescribed robustness properties are made based on finite data.* We provide the means to make decisions in multi-agent networks affected by uncertainty on the basis of a finite number of data, i.e., a finite number of realizations of the uncertainty that affects the system. Importantly, we accompany these decisions with *a priori* probabilistic certificates on their robustness properties, i.e., we quantify how likely it is that they remain feasible when it comes to new, unseen realizations of the uncertainty not included in the data. To achieve this, we show how to apply to our context the finite sample complexity bounds in data driven optimization derived based the so called scenario approach in Calafiore and Campi (2006); Campi and Garatti (2008); Campi et al. (2009); Campi and Garatti (2011); Schildbach et al. (2013); Garatti and Campi (2013); Campi and Garatti (2018a).

2. *Data is treated as a private resource.* We do not require agents to have access or share the same set of data. In contrast, we only assume that each agent has access to a different set of data, not necessarily the same amount as other agents. This widens the applicability of the proposed methodologies, as even when it comes to modeling the same uncertain phenomenon, it is often unrealistic to assume that there is a centralized database that agents have access to. Moreover, exchanging data among agents is often impracticable and raises privacy and data ownership concerns. To allow for data to constitute private information of each agent, we view the problem under a statistical learning theoretic lens Campi and Garatti (2023); Margellos et al. (2015) and capitalize on recent results of Campi et al. (2015, 2018); Garatti and Campi (2022).

3. *The data private approach can be naturally embedded in distributed algorithms for multi-agent cooperative optimization.* It is applicable to a wide range of algorithms, and only requires agents to exchange interim decisions with agents considered as neighbors according to an underlying network. These decisions, however, are based on their local data sets that do not need to be exchanged with other agents over the network. We discuss two important algorithm classes that fit this regime, namely, the case where agents seek consensus on a common decision, and the case where agents seek agreement on the way they will share a given resource. We term these cases as *decision coupled* and *constraint coupled* problems, respectively. For each case, we also detail one distributed algorithm that fits our setting just to illustrate the overall distributed architecture, and other algorithms could be used with minor modifications. To this end, we unify and outline the algorithms introduced in Margellos et al. (2018), Falsone et al. (2017b).

The remainder of the paper is structured as follows. Section 2 shows how to unify decision and constraint coupled optimization programs, and poses the problem of providing probabilistic robustness certificates in multi-agent optimization with agent-private data. Section 3 establishes such probabilistic certificates and investigates their complexity with respect to the number of agents, thus providing insights on their scalability properties. In Section 4 we provide general algorithmic architectures to solve data-private multi-agent optimization problems in a distributed manner. Section 5 concludes the paper and provides some research vistas.

2 Multi-agent learning with private data

2.1 Problem set-up

We consider a population of m agents indexed by $i = 1, \dots, m$, that interact with each other over a network and cooperate to achieve a common objective. The agent population is affected by uncertainty; and in our exposition, we collect all sources of uncertainty that affect the multi-agent system in a vector $\delta \in \Delta \subseteq \mathbb{R}^{n_\delta}$. We assume that Δ is endowed with a σ-algebra $\mathcal{D}$, and that $\mathbb{P}$ is a probability measure defined over $\mathcal{D}$. We adopt a data driven perspective and model uncertainty by resorting to data/scenarios. To this end, we impose the following assumption.

Assumption 1 (Sample independence)**.** Each agent i has access to a collection $S_i \subset \Delta$ of $N_i \in \mathbb{N}_+$ independent scenarios of δ drawn according to $\mathbb{P}$, $i = 1, \dots, m$. Moreover, the scenarios across all agents are assumed to be independent.

Assumption 1 implies that each agent has access to a (possibly different) finite set of scenarios about a common uncertainty source. This finite set constitutes a dataset (e.g., historical data) that is local to agent i, $i = 1, \dots, m$, and should not be shared with the entire population, mainly for information privacy reasons, but also because such datasets could be (depending on the application) large objects and exchanging such information could be intensive from a communication point of view.

Depending on the population objective, we distinguish two different classes of cooperative multi-agent optimization problems, namely, *decision coupled* and *constraint coupled* optimization problems.

1. *Decision coupled problems:* In this case, agents seek to reach consensus on a common decision vector $x \in \mathbb{R}^n$ that optimizes a global objective that is in turn represented as the sum of local ones. At the same time, this decision needs to satisfy all local constraints that are imposed by each agent, i.e., it has to lie at the intersection of the local constraint sets.

More formally, each agent i, $i = 1, \dots, m$, contributes to the global objective function (that is to be optimized collectively by all agents) by means of a local objective functions $f_i(\cdot) : \mathbb{R}^n \to \mathbb{R}$, while decision x needs to belong to a local constraint set that encodes limitations/conditions that agent i is subject to. These limitations/conditions depend on the uncertainty vector δ; since agent i's uncertainty knowledge is defined by the scenarios in the dataset S_i, we denote the constraint set of agent i by $\bigcap_{\delta \in S_i} X_i(\delta)$, where $X_i(\delta) \subseteq \mathbb{R}^n$ incorporates all restrictions imposed by agent i to the decision vector for the scenario δ, including for example constraints expressed by inequalities of the form $h_i(x, \delta) \leq 0$.

The decision coupled program is then encoded by the optimization problem P_{decision} below:

$$\begin{aligned} \mathrm{P}_{\text{decision}}: \ & \min_{x \in \mathbb{R}^n} \sum_{i=1}^{m} f_i(x) \\ & \text{subject to } x \in \bigcap_{i=1}^{m} \bigcap_{\delta \in S_i} X_i(\delta). \end{aligned}$$

Note that local objective functions and constraint sets, as well as the local datasets, constitute agent-private information that is not to be shared with the entire population. The population's objective is then to determine an optimal consensus decision (a minimizer of P_{decision}), while preserving privacy of the local information.

2. *Constraint coupled problems:* In this case, agents do not seek to determine a common decision vector; in contrast, each agent i, $i = 1, \dots, m$, has a local decision vector $x_i \in \mathbb{R}^{n_i}$, a local objective function $f_i(\cdot) : \mathbb{R}^{n_i} \to \mathbb{R}$, and a local constraint set $\widetilde{X}_i(\delta) \subseteq \mathbb{R}^{n_i}$ (note that we use the notation $\widetilde{X}_i$ as opposed to X_i used in decision coupled problems for notation reasons that will become clear in the sequel). Agents cooperate in this case to solve a resource sharing problem. The latter is summarized by $P_{\text{constraint}}$ below:

$$\begin{aligned} \mathrm{P}_{\text{constraint}}: \ & \min_{\{x_i \in \mathbb{R}^{n_i}\}_{i=1}^m} \sum_{i=1}^{m} f_i(x_i) \\ & \text{subject to } x_i \in \bigcap_{\delta \in S_i} \tilde{X}_i(\delta), \ i = 1, \dots, m \\ & \sum_{i=1}^{m} g_i(x_i) \leq 0, \end{aligned}$$

where the term $g_i(x_i) : \mathbb{R}^{n_i} \to \mathbb{R}^p$ quantifies the amount of p resources that are required by agent i to implement its decision x_i. The coupling among the agents' decisions is due to the constraint $\sum_{i=1}^{m} g_i(x_i) \leq 0$, which in turn encodes a resource/budget constraint.

As with decision coupled problems, agents seek to determine optimal local decisions (minimizers of $P_{\text{constraint}}$), without sharing information considered as private like f_i, $\tilde{X}_i$, and their contribution g_i to the coupling constraint. Moreover, each agent only has access to its own dataset S_i that is not to be shared with other agents.

Depending on the physical and communication network structure according to which agents exchange information with other agents, different privacy-preserving algorithms exist for both classes of problems. In Section 4 we present for each problem one such distributed implementation that is general enough to encompass several characteristics that appear in alternative algorithmic implementations as well, and thus provide a general architecture that pertains iterative methods that solve such problems in a distributed manner. It should be noted, however, that depending on the application at hand, the network structure and the information that is considered as private, other algorithms may be applicable/preferable. We provide pointers to such alternatives in Section 4.

2.2 Unifying problem statement

Let $S = \bigcup_{i=1}^{m} S_i$ be the collection of the scenarios in all agents' datasets, and denote by N their number, i.e., $N = \sum_{i=1}^{m} N_i$. Consider the following optimization problem:

$$\begin{aligned} \mathrm{P}_N: \ & \min_{x \in X} f(x) \\ & \text{subject to } x \in \bigcap_{i=1}^{m} \bigcap_{\delta \in S_i} X_i(\delta), \end{aligned} \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is an implicit domain, encoding deterministic constraints, i.e., constraints that do not depend on the uncertain parameter δ. Note that we introduce the subscript N to denote that such a problem, as well as its solution, depends collectively on all data used.

P_N includes the decision coupled problem $\mathrm{P}_{\text{decision}}$, and the constraint $\mathrm{P}_{\text{constraint}}$, as special instances. To see this, consider the following assignments.

1. From $\mathrm{P}_{\text{decision}}$ to P_N: Set $X = \mathbb{R}^n$, and $f(\cdot) = \sum_{i=1}^m f_i(\cdot)$, we directly observe that $\mathrm{P}_{\text{decision}}$ is in the format of P_N.

2. From $\mathrm{P}_{\text{constraint}}$ to P_N: Let $n = \sum_{i=1}^m n_i$ and set $x = [x_1^\top \cdots x_m^\top]^\top$, $X = \{x \in \mathbb{R}^n : \sum_{i=1}^m g_i(x_i) \leq 0\}$, $f(\cdot) = \sum_{i=1}^m f_i(\cdot)$, and
$$X_i(\delta) = \mathbb{R}^{n_1} \times \cdots \times \tilde{X}_i(\delta) \times \cdots \times \mathbb{R}^{n_m}, \text{ for all } i = 1, \dots, m.$$
Under these assignments $\mathrm{P}_{\text{constraint}}$ can be written in the format of P_N.

We impose the following structural assumption.

Assumption 2 (convexity and well-posedness)**.** We assume that

1. the function $f(\cdot)$ and the set X are convex;
2. for every $i = 1, \dots, m$ and $\delta \in \Delta$, $X_i(\delta)$ is convex;
3. for every $i = 1, \dots, m$ and every finite set S_i, the set $\cap_{\delta \in S_i} X_i(\delta)$ is compact;
4. for every collection $\{S_i\}_{i=1}^m$, the set $\left(\bigcap_{i=1}^m \bigcap_{\delta \in S_i} X_i(\delta)\right) \cap X$ has a non-empty interior.

Under Assumption 2, the set of minimizers of P_N is non-empty. We assume here that P_N (for any set of scenarios S) admits a unique solution; if this is not the case, we assume that a deterministic, convex tie-break rule is adopted to single-out a particular minimizer.

This unified representation of decision coupled and constraint coupled problems by means of P_N allows us to analyze simultaneously their properties as learning algorithms. Specifically, P_N could be thought of as a learning algorithm that implicitly maps data (the scenarios in S) to decisions (the minimizer $x_N^\star$ of P_N). We seek to analyze the generalization properties of these learned decisions, namely, we want to provide *a priori* guarantees on how likely it is that $x_N^\star$ becomes infeasible for the agents' population constraints, when it comes to a new realization $\delta \in \Delta$ that is not includes in S. Since $x_N^\star$ depends on the data used, it constitutes a random quantity, and as a result the probability that it becomes infeasible is random as well. To this end, we can only provide guarantees on the probability that $x_N^\star$ becomes infeasible with certain confidence. We formalize this in the next problem.

Problem 1. *Fix* $\beta \in (0, 1)$*. Determine* $\varepsilon \in (0, 1)$ *such that*
$$\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin \bigcap_{i=1}^m X_i(\delta)\Big\} \leq \varepsilon\Big\} \geq 1 - \beta. \tag{2}$$

In words, we seek to determine a violation level threshold $\varepsilon \in (0, 1)$, such that the probability that $x_N^\star$ violates the agents' constraints $\bigcap_{i=1}^m X_i(\delta)$ corresponding to a new realization of the uncertainty δ, is at most equal to ε. Since $x_N^\star$ is a random quantity that depends on the multi-sample S, the violation assessment can only hold true probabilistically, with a confidence of at least $1 - \beta$, measured with respect to the product probability $\mathbb{P}^N$ as all scenarios in S are independent by Assumption 1. Statements of that form constitute generalization results within the realm of probably approximately correct (PAC) learning Vidyasagar (2002); Margellos et al. (2015).

In the next section we provide an answer to Problem 1 and offer a characterization of ε that allows for such generalization statements. Such expressions for ε would be a function of the pre-specified confidence level β, the cardinalities of the agents' datasets $\{N_i\}_{i=1}^m$, and a quantity related to data compression properties that will be specified in the sequel. Due to these dependencies the obtained expressions for ε would also depend on the number of agents m. We will provide different bounds and analyze their complexity properties with respect to the number of agents m. It is worth mentioning that, interestingly, these expressions for ε do not depend on any of the problem P_N features, namely the objective function and the form of the constraint sets, and hold true in a distribution-free fashion, i.e., irrespective of the underlying probability distribution $\mathbb{P}$ according to which data are generated.

We refer to violation levels ε as probabilistic certificates, as they accompany a learned optimizer $x_N^\star$ carrying information on its reliability level as far as constraint satisfaction is concerned.

3 Probabilistic certificates

Our probabilistic certificates rely on the notion of minimal support set of the solution $x_N^\star$ of P_N (see Campi et al. (2018); Garatti and Campi (2025)), which is related to data compression properties. For a given optimization program, a minimal support set is a minimal cardinality subset of constraints such that solving the optimization problem only with these constraints yields the same solution with the one that is obtained when all the constraints are employed. If there are multiple minimal support sets, a unique one is supposed to be picked according to any selection rule. In a sense, the constraints that are not in the minimal support set are inessential since removing all of them leaves the solution unchanged. It is well known that for convex optimization programs the cardinality of the minimal support set is always no bigger than the number n of decision variables, see Calafiore and Campi (2006). In some cases the cardinality of the minimal support set can be strictly smaller than n and improved bounds can be obtained, see e.g., Schildbach et al. (2013). We denote by $d \leq n$ any available upper-bound to the cardinality of the minimal support set of P_N.

Based on the aforementioned description of the minimal support set, it follows directly that it is related to data compression properties. To see this, for each agent i, $i = 1, \dots, m$, and for each S_i, let $d_{i,N}(S)$ denote how many scenarios in the minimal support set belong to S_i

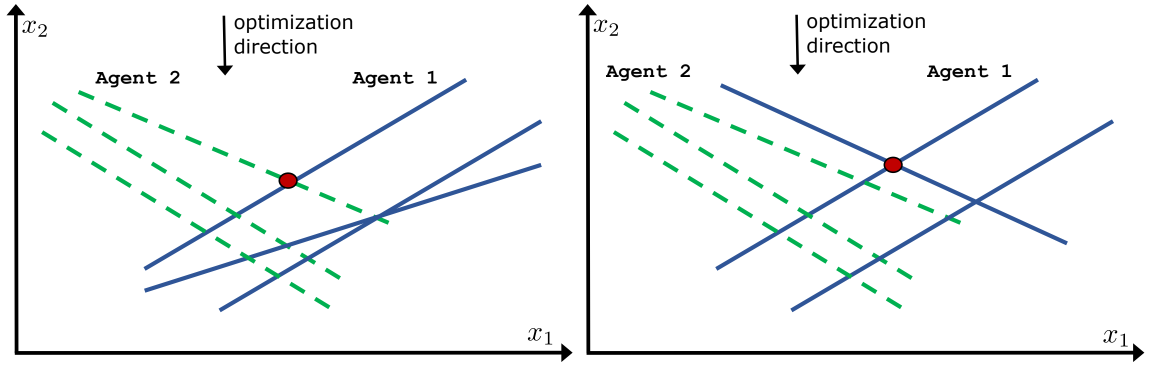

Fig. 1 Minimal support sets for a problem with two agents. The problem involves two decision variables x_1, x_2, and we seek to minimize x_2 (indicated by the optimization direction). For the instance in the left we have $d_{1,N} = 1$ and $d_{2,N} = 1$ (one dashed and one solid constraint support the solution), while for the one in the right we have $d_{1,N} = 2$ and $d_{2,N} = 0$ (two solid constraints support the solution).

(the dataset of agent i). Denote then the set that contains these scenarios by C_i, where $\|C_i\| = d_{i,N}(S)$. Notice that $d_{i,N}(S)$ can possibly be also zero (and hence C_i could be empty) as the way that constraints in the minimal support set split among agents varies according to the extracted S. Still, irrespective of $S \in \Delta^N$, it clearly holds that $\sum_{i=1}^m d_{i,N}(S) \leq d$, where $d \leq n$ is the bound on the cardinality of the minimal support set.

Solving (1) by replacing its constraint by $x \in \bigcap_{i=1}^m \bigcap_{\delta\in C_i} X_i(\delta)$ we obtain a solution $x_d^\star$ (we introduce the subscript d as this is a bound on the number of scenarios in $\bigcup_{i=1}^m C_i$). By the properties of the minimal support set, and since all scenarios that give rise to constraints in the minimal support set are included as constraints in $\bigcap_{i=1}^m \bigcap_{\delta\in C_i} X_i(\delta)$, we directly have that $x_N^\star = x_d^\star$. In other words, $\bigcup_{i=1}^m C_i$ serves as a compression of the dataset S in the sense that it leads to the same solution.

A pictorial illustration of this fact is given in Fig. 1 for a problem with two agents. The problem involves two decision variables, x_1, x_2 and we seek to minimize x_2 (indicated by the optimization direction). Scenarios give rise to different type of constraints: the solution must stay above solid lines for agent 1, above dashed lines for agent 2. In Fig. 1, two different scenario extractions are represented, corresponding to $d_{1,N} = 1$ and $d_{2,N} = 1$ (left panel, one dashed and one solid constraint support the solution) and to $d_{1,N} = 2$ and $d_{2,N} = 0$ (right panel, two solid constraints support the solution).

To ease notation, we will write in the sequel $d_{i,N}$ in place of $d_{i,N}(S)$ and make the dependency on S explicit only when necessary.

3.1 A subadditivity bound

We first provide a direct approach to accompany $x_N^\star$ with a probabilistic feasibility certificate. This result is the more straightforward out of the certificates provided in this section, however, besides being interesting per se, it sets the stage for the subsequent developments.

Since it always holds that $d_{i,N} \leq d$ for each individual $i = 1, \dots, m$, one can apply either the main result in Calafiore and Campi (2006) or in Campi and Garatti (2008) conditionally to the scenarios of all other agents and then, integrating with respect to the realizations of these scenarios of the other agents, one obtains a feasibility guarantee that holds locally, i.e. for the constraints of agent i only. For instance, using the result in Calafiore and Campi (2006), letting $\beta_i \in (0, 1)$ and

$$\widetilde{\varepsilon}_i = 1 - \sqrt[N_i-d]{\frac{\beta_i}{\binom{N_i}{d}}}; \tag{3}$$

then, it holds that

$$\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\} \leq \widetilde{\varepsilon}_i\Big\} \geq 1 - \beta_i. \tag{4}$$

This results along with the subadditivity of $\mathbb{P}^N$ and $\mathbb{P}$ can be used to establish the following theorem on the probabilistic feasibility of $x_N^\star$ for the global constraint $\bigcap_{i=1}^m X_i(\delta)$.

Theorem 1. *Consider Assumptions 1 and 2. Given $\beta \in (0, 1)$, let β_i, $i = 1, \dots, m$, be such that $\sum_{i=1}^m \beta_i = \beta$. For each $i = 1, \dots, m$, let $\widetilde{\varepsilon}_i$ be as in* (3) *and $\widetilde{\varepsilon} = \sum_{i=1}^m \widetilde{\varepsilon}_i$. Then, it holds that*

$$\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin \bigcap_{i=1}^m X_i(\delta)\Big\} \leq \widetilde{\varepsilon}\Big\} \geq 1 - \beta. \tag{5}$$

Proof. The following chain of inequalities using the subadditivity properties of the underlying measure hold; a similar argument has been also used in Kariotoglou et al. (2016).

$$
\begin{aligned}
&\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin \bigcap_{i=1}^m X_i(\delta)\Big\} \le \sum_{i=1}^m \widetilde{\varepsilon}_i\Big\} = \mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : \exists i \in \{1,\dots,m\} \text{ such that } x_N^\star \notin X_i(\delta)\Big\} \le \sum_{i=1}^m \widetilde{\varepsilon}_i\Big\} \\
&\quad = \mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\bigcup_{i=1}^m \Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\}\Big\} \le \sum_{i=1}^m \widetilde{\varepsilon}_i\Big\} \ge \mathbb{P}^N\Big\{S \in \Delta^N : \sum_{i=1}^m \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\} \le \sum_{i=1}^m \widetilde{\varepsilon}_i\Big\} \\
&\quad \ge \mathbb{P}^N\Big\{\bigcap_{i=1}^m \Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\} \le \widetilde{\varepsilon}_i\Big\}\Big\} \ge 1 - \sum_{i=1}^m \mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\} > \widetilde{\varepsilon}_i\Big\} \\
&\quad \ge 1 - \sum_{i=1}^m \beta_i, \qquad (6)
\end{aligned}
$$

where the last step follows from (4). This concludes the proof. □

In words, Theorem 1 establishes the fact that with confidence at least $1-\beta$, $\widetilde{\varepsilon}$ constitutes a certificate for $x_N^\star$, that bounds the probability that it becomes infeasible when it comes to a new realization of $\delta \in \Delta$. However, the certificate $\widetilde{\varepsilon}$ offered by Theorem 1 tends to be very conservative when the number of agents is large, which in turn hampers its practical applicability. To see this, consider the specific case where scenarios are common across the agents, i.e., $N_1 = \cdots = N_m$ and $\beta_i = \beta/m$, for all $i = 1, \dots, m$. In this case, (3) leads to all $\widetilde{\varepsilon}_i$, $i = 1, \dots, m$, being equal say to some $\bar{\varepsilon}$. As a result, the probabilistic certificate offered by Theorem 1 is $\widetilde{\varepsilon} = \sum_{i=1}^m \widetilde{\varepsilon}_i = m\bar{\varepsilon}$, i.e., it increases (approximately) linearly with the number of agents m. We use the term "approximately" as in (3) we have $\beta_i = \beta/m$, and as a result $\bar{\varepsilon}$ also depends on m via β_i, however, this dependence is provably logarithmic and has a negligible effect. Numerical calculations verify this claim as it will be discussed in Section 3.4 and Fig. 2.

Remark 1. Following Campi and Garatti (2008) instead of Calafiore and Campi (2006), a result similar to can be given by replacing $\widetilde{\varepsilon}_i$, $i = 1, \dots, m$, in (3) with the solution of the equation $\sum_{k=0}^{d-1} \binom{N_i}{k} \widetilde{\varepsilon}_i^k (1-\widetilde{\varepsilon}_i)^{N_i-k} = \beta_i$. which can be numerically computed via bisection. This results in a much improved $\widetilde{\varepsilon}$ and, therefore, in a tighter certification. For simplicity, we use (3) as this constitutes an explicit expression and facilitates the subsequent comparison of different certificates in terms of their complexity.

3.2 A tighter bound

The conservative behavior of Theorem 1 in terms of the way the associated certificate scales with the number of agents m is due to the fact that it considers all $d_{i,N}$, $i = 1, \dots, m$, equal to the (pessimistic) upper-bound d. In fact, $\sum_{i=1}^m d_{i,N} \le d$ shows that when $d_{i,N} = d$ for some i, then $d_{j,N}$ must be 0 for all other $j \neq i$. In this subsection, we want to exploit the inequality $\sum_{i=1}^m d_{i,N} \le d$ to reduce the conservatism of Theorem 1. To achieve this we capitalize on the results of Campi et al. (2015, 2018) that are based on a wait-and-judge, a posteriori, perspective. Following Theorem 1 and Remark 4 in Campi et al. (2018), for each $i = 1, \dots, m$, fix $\beta_i \in (0,1)$ and consider function $\varepsilon_i(\cdot)$ defined as follows:

$$\varepsilon_i(k) = 1 - \sqrt[N_i-k]{\frac{\beta_i}{(d+1)\binom{N_i}{k}}}, \text{ for all } k = 0, \dots, d. \qquad (7)$$

Besides k, $\varepsilon_i(\cdot)$ depends on N_i, β_i and d as well, but this dependency is not explicitly indicated to ease the notation. By focusing on a given agent i, $i = 1, \dots, m$, an application of Theorem 1 of Campi et al. (2018) conditional to the scenarios of all other agents $S \setminus S_i$ yields

$$\mathbb{P}^{N_i}\Big\{S \in \Delta^N : \mathbb{P}\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\} \le \varepsilon_i(d_{i,N}) \,\Big|\, \{S \setminus S_i \in \Delta^{N-N_i}\}\Big\} \ge 1 - \beta_i. \qquad (8)$$

Integrating (8) with respect to the probability of realizing the scenarios $S \setminus S_i$, we then have that

$$\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin X_i(\delta)\Big\} \le \varepsilon_i(d_{i,N})\Big\} \ge 1 - \beta_i. \qquad (9)$$

That is, for each agent $i = 1, \dots, m$, with confidence at least $1 - \beta_i$, we have that $x_N^\star$ violates the constraint set $X_i(\delta)$ of agent i (for a new uncertainty realization $\delta \in \Delta$) with probability no bigger than $\varepsilon_i(d_{i,N})$. Though (9) may resemble (4), note that there is a big difference in that $d_{i,N}$ in (9) depends on the seen scenarios and hence is not a-priori known. The inequality (9) can be used in place of (4) in the subadditivity-based proof of Theorem 1 (see (6)) to obtain the following characterization of the global feasibility of $x_N^\star$ for new values of δ:

$$\mathbb{P}^N\Big\{S \in \Delta^N : \mathbb{P}\Big\{\delta \in \Delta : x_N^\star \notin \bigcap_{i=1}^m X_i(\delta)\Big\} \le \sum_{i=1}^m \varepsilon_i(d_{i,N})\Big\} \ge 1 - \sum_{i=1}^m \beta_i. \qquad (10)$$

Differently from (15) and (1), the assessment of the violation probability level in (10) is *a posteriori* because $\varepsilon_i(d_{i,N})$ is a function of the seen scenarios.

To provide an *a priori* assessment we capitalize on this *a posteriori* result and compute the worst-case allocation of $d_{i,N}$, $i = 1, \dots, m$ across agents (recall that these quantities depend on S) that results in the maximum value of $\sum_{i=1}^m \varepsilon_i(d_{i,N})$, while satisfying $\sum_{i=1}^m d_{i,N} \le d$.

This is encoded by the following optimization program:

$$\varepsilon = \max_{\{d_i \in \mathbb{N}_+\}_{i=1}^m} \sum_{i=1}^m \varepsilon_i(d_i), \tag{11}$$
$$\text{subject to } \sum_{i=1}^m d_i \leq d,$$

which is an integer maximization program that can be solved numerically. Notice that $\{d_i\}_{i=1}^m$ in (11) are integer optimization variables, which should not be confused with $\{d_{i,N}\}_{i=1}^m$. We then have the following theorem.

Theorem 2. *Consider Assumptions 1 and 2. Fix $\beta \in (0,1)$ and choose β_i, $i = 1,\ldots,m$, such that $\sum_{i=1}^m \beta_i = \beta$. Set ε according to (11). We then have that*

$$\mathbb{P}^N \Big\{ S \in \Delta^N : \ \mathbb{P}\Big\{ \delta \in \Delta : \ x_N^\star \notin \bigcap_{i=1}^m X_i(\delta) \Big\} \leq \varepsilon \Big\} \geq 1 - \beta. \tag{12}$$

Proof. For any set S of scenarios it holds that $\sum_{i=1}^m d_{i,N}(S) \leq d$, which means that $\{d_{i,N}(S)\}_{i=1}^m$ is feasible for (11). Thus $\sum_{i=1}^m \varepsilon_i(d_{i,N}(S)) \leq \varepsilon$, as ε is the maximum value of the optimization problem in (11). Using $\sum_{i=1}^m \varepsilon_i(d_{i,N}(S)) \leq \varepsilon$ in (10) gives (12). □

Enforcing the condition $\sum_{i=1}^m d_{i,N} \leq d$ when determining ε in (11), results in Theorem 2 offering a tighter estimate for the certificate ϵ with respect to that (denoted as $\widetilde{\varepsilon}$) in Theorem 1. This is verified in Section 3.4 and Fig. 2 where we compare the various certificates provided. When the number of agents is very large and/or there are few scenarios available, ε may still exceed 1, making the result of Theorem 2 trivial. Note that Theorem 2 can be reversed to compute the number of scenarios N_i that need to be extracted by agent i, $i = 1,\ldots,m$, for given values of $\varepsilon, \beta \in (0,1)$. This can be achieved by numerically seeking for values of N_i, $i = 1,\ldots,m$, that lead to a solution of (11) that attains the desired ε.

Remark 2. As a follow-up to Remark 1, we observe that an improved result can also be achieved in the present case without conceptual changes. This can be done by utilizing the more recent wait-and-judge certifications of Campi and Garatti (2018b); Garatti and Campi (2022, 2025) rather than those of Campi et al. (2018) used previously. In particular, according to Garatti and Campi (2025), enhanced expressions for $\epsilon_i(\cdot)$ can be obtained for which the result (8) remains valid. This ultimately leads to tighter certifications in (10) and in (12), while the conceptual significance of Theorem 2 remains unaltered. However, similarly to the $\widetilde{\epsilon}_i$ in Remark 1, the enhanced $\epsilon_i(\cdot)$ as functions of k, N_i, β_i and d are not provided in closed form, but their value needs to be numerically computed by solving certain polynomial equations. See Garatti and Campi (2025) for details.

3.3 The case of common datasets

We consider now the specific case where all agents have access to the same dataset. This common information structure allows to provide sharper certificates and serves as a benchmark when compared with the certificates derived in the previous subsections.

To this end, for all $i = 1,\ldots,m$, let $S_i = \bar{S}$ where $\bar{S} \subset \Delta$ is a set of $\bar{N} \in \mathbb{N}_+$ scenarios independently extracted from Δ according to $\mathbb{P}$, and are available to all agents. The optimization program P_N in (1) then takes the form

$$\mathrm{P}_{\bar{N}}: \ \min_{x \in X} f(x) \tag{13}$$
$$\text{subject to } x \in \bigcap_{i=1}^m \bigcap_{\delta \in \bar{S}} X_i(\delta),$$

where we changed the subscript from N to $\bar{N}$ to emphasize the fact that there are $\bar{N}$ common scenarios. Let us denote by $x_{\bar{N}}^\star$ the solution of $\mathrm{P}_{\bar{N}}$ (which is well-defined based on Assumption 2); we assume again that this solution is unique, while in the opposite case a deterministic, convex tie-break rule can be employed to single-out a solution.

We can now accompany $x_{\bar{N}}^\star$ with a probabilistic certificate regarding its feasibility properties. In the present context, the theory of the scenario approach developed in Calafiore and Campi (2006); Campi and Garatti (2008) provides a direct characterization, showing that $x_{\bar{N}}^\star$ is feasible for $\bigcap_{i=1}^m \bigcap_{\delta \in \bar{S}} X_i(\delta)$ when it comes to a new realization of the uncertainty $\delta \in \Delta$ up to an explicitly quantified probabilistic level $\bar{\varepsilon}$. To establish this, notice that under Assumption 2, $\mathrm{P}_{\bar{N}}$ is a convex optimization program; let $d \in \mathbb{N}_+$ be any available upper-bound to the cardinality of its minimal support set (recall that $d \leq n$ due to Calafiore and Campi (2006)). The following theorem is a direct consequence of the results of Calafiore and Campi (2006).

Theorem 3. *Consider Assumptions 1 and 2. Fix $\beta \in (0,1)$ and let*

$$\bar{\varepsilon} = 1 - \sqrt[\bar{N}-d]{\frac{\beta}{\binom{\bar{N}}{d}}}. \tag{14}$$

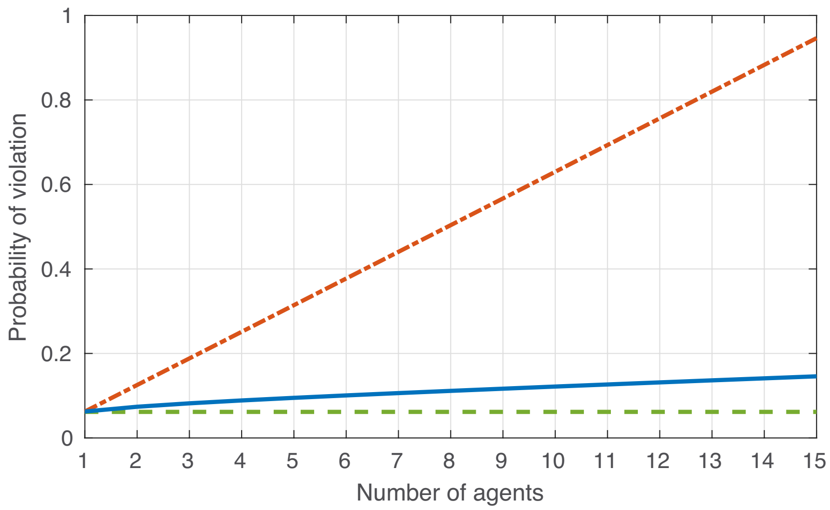

Fig. 2 Dashed green line: $\bar{\varepsilon}$ in Theorem 3; dotted-dashed red line: $\widetilde{\varepsilon}$ in Theorem 1; solid blue line: ε in Theorem 2.

We then have that

$$\mathbb{P}^{\bar{N}}\Big\{\bar{S} \in \Delta^{\bar{N}} : \mathbb{P}\Big\{\delta \in \Delta : x^\star_{\bar{N}} \notin \bigcap_{i=1}^{m} X_i(\delta)\Big\} \leq \bar{\varepsilon}\Big\} \geq 1 - \beta. \tag{15}$$

Theorem 3 implies that with confidence at least $1-\beta$, $\bar{\varepsilon}$ acts as a probabilistic certificate for the infeasibility properties of $x^\star_{\bar{N}}$. If $\bar{N}$ is too small, it may be that $\bar{\varepsilon}$ is larger than 1 and the theorem is not of practical interest. In this case, one may want to fix $\bar{\varepsilon}, \beta \in (0,1)$ and use Theorem 3 the other way around to determine how many scenarios are needed for (15) to hold. This amounts to solving (14) with respect to $\bar{N}$. See (Calafiore and Campi, 2006, Theorem 1).

Note that, as in Remark 1, following the tight bound provided in Campi and Garatti (2008), an improved result could be given by replacing $\bar{\varepsilon}$ in (14) with the solution of the equation $\sum_{k=0}^{d-1} \binom{\bar{N}}{k} \bar{\varepsilon}^k (1-\bar{\varepsilon})^{\bar{N}-k} = \beta$.

3.4 Comparison of bounds

In this subsection we compare the certificate $\widetilde{\varepsilon}$ provided in Theorem 1 using a subadditivity approach, the tighter certificate ε established in Theorem 2, and the certificate $\bar{\varepsilon}$ of Theorem 3. The last one serves as a benchmark, corresponding to the specific case where all agents have a common dataset. The comparison investigates how the obtained certificates scale with respect to the number of agents m.

Fig. 2 illustrates pictorially this comparison, where $\widetilde{\varepsilon}$ is shown with dotted-dashed red line, ε with solid blue line, and $\bar{\varepsilon}$ with dashed green line. We consider $\beta = 10^{-5}$, $N_i = \bar{N} = 4500$, $\beta_i = \beta/m$, $i = 1, \ldots, m$, and $d = 50$. As it appears, $\widetilde{\varepsilon}$ grows (approximately) as $m \cdot \bar{\varepsilon}$ (see discussion above Remark 1), while ε is only moderately increasing with m.

4 Distributed solution algorithms

In this section we return back to the class of decision coupled $\mathrm{P}_{\text{decision}}$ and constraint coupled $\mathrm{P}_{\text{constraint}}$ optimization problems. For each case we present a general purpose architecture, within which agents can exchange tentative information – keeping their local data sets, objective function and constraint set private – while solving the underlying problem in a distributed manner. In each case a wide range of algorithms is applicable. We do not provide an exhaustive analysis but rather outline one such algorithm per problem class to better illustrate our architecture and highlight the assumptions required and the underlying convergence statements. We then provide pointers to alternative algorithms that could be generalized to this data driven counterpart in an analogous way.

4.1 Decision coupled problems

Consider problem $\mathrm{P}_{\text{decision}}$ as presented in Section 2. Note that under the variable assignments in Section 2.1, we postulate that Assumption 2 holds for $\mathrm{P}_{\text{decision}}$. Agents' objective is to reach consensus on common decision; in particular, we would like this decision to be the optimizer of the cooperative problem in $\mathrm{P}_{\text{decision}}$. We discuss a distributed architecture that can be followed by agents in an iterative manner, so that they converge to such a decision. To this end, we will present a communication-computation algorithm, where at each step agents will be exchanging information, maintaining a copy of the vector they seek to reach consensus on. Denote by $k \in \mathbb{N}$ the iteration index, and let $x_i(k)$ be a copy/estimate of the common decision vector agents seek to agree upon, maintained by agent i, $i = 1, \ldots, m$, at iteration k.

Information vector: At iteration k, each agent $i = 1, \dots, m$, constructs the information vector

$$z_i(k) = \sum_{j=1}^{m} a_j^i(k) x_j(k).$$

This can be thought of as a weighted average of the tentative decisions $x_j(k)$, $j = 1, \dots, m$, of various agents at iteration k of the algorithm. The mixing weights in this step $a_j^i(k)$ dictate how agent i weighs information received by agent j, at iteration k of the process. This quantity constitutes an (aggregate) information vector that is available to each agent, at a given iteration. We impose the following assumption on the mixing weights.

Assumption 3 (Weight coefficients)**.** There exists $\eta \in (0, 1)$ such that for all $i, j \in \{1, \dots, m\}$ and all $k \geq 0$, $a_j^i(k) \in \mathbb{R}_+ \cup \{0\}$, $a_i^i(k) \geq \eta$, and $a_j^i(k) > 0$ implies that $a_j^i(k) \geq \eta$. Moreover, for all $k \geq 0$,

1. $\sum_{j=1}^m a_j^i(k) = 1$ for all $i = 1, \dots, m$,
2. $\sum_{i=1}^m a_j^i(k) = 1$ for all $j = 1, \dots, m$.

According to Assumption 3, we allow some lack of information at specific iterations of the algorithm. To see this, notice that if $a_j^i(k) = 0$, then agents i and j do not exchange information at iteration k of the process. Therefore, this mixing step can be executed in a fully distributed manner, with agents exchanging information according to an underlying communication protocol.

Such an assumption is rather standard in the distributed optimization literature; see Nedíc et al. (2009, 2010); Olshevsky and Tsitsiklis (2011); Zhu and Martínez (2012). The interpretation of having a uniform lower bound η, independent of k, for the coefficients $a_j^i(k)$ in Assumption 3 is that it ensures that each agent is mixing information received by other agents at a non-diminishing rate in time, Nedíc et al. (2010). Moreover, points 1 and 2 in Assumption 3 ensure that this mixing is a convex combination of the other agent estimates, assigning a non-zero weight to its local one since $a_i^i(k) \geq \eta$. Note that satisfying Assumption 3 requires agents to agree on an infinite sequence of doubly stochastic matrices (double stochasticity arises due to conditions 1 and 2 in Assumption 3), where $a_j^i(k)$ would be element (i, j) of the matrix at iteration k. This agreement should be performed prior to the execution of the algorithm in a centralized manner, and the resulting matrices have to be communicated to all agents via some consensus scheme; this is standard in distributed optimization algorithms of this type (see also Nedíc et al. (2010); Olshevsky and Tsitsiklis (2011); Zhu and Martínez (2012)).

Decision update: At iteration k, once the information vector $z_i(k)$ is computed, each agent i, $i = 1, \dots, m$, updates its local estimate from $x_i(k)$ to $x_i(k+1)$ by performing the following update

$$x_i(k+1) = \arg \min_{x_i \in \bigcap_{\delta \in S_i} X_i(\delta)} f_i(x_i) + \frac{1}{2c(k)} \|z_i(k) - x_i\|^2,$$

where $c(k)$ is a positive coefficient. Note that due to presence of the quadratic term this problem is strongly convex and the set of minimizers is a singleton; hence, $x_i(k+1)$ is always uniquely defined. This is a local minimization problem, involving only the local objective function $f_i(x_i)$ and the local constraint set $\bigcap_{\delta \in S_i} X_i(\delta)$ of agent i. Notice that the latter depends only on the data in S_i, which constitutes private information of agent i. This update step involves trading between minimizing the local objective function f_i and minimizing disagreement from the information vector $z_i(k)$, where the latter captures the aggregate behavior of the rest of the network. The way these are traded is through the iteration-varying proxy coefficient $c(k)$. Hence we impose the following assumption on the proxy coefficient.

Assumption 4 (Coefficient $\{c(k)\}_{k \geq 0}$)**.** For all $k \geq 0$, $c(k) \in \mathbb{R}_+$ and $\{c(k)\}_{k \geq 0}$ is a non-increasing sequence, i.e., $c(k) \leq c(r)$ for all $k \geq r$, with $r \geq 0$. Moreover,

1. $\sum_{k=0}^\infty c(k) = \infty$,
2. $\sum_{k=0}^\infty c(k)^2 < \infty$.

Notice that point 2 of the assumption implies that $\lim_{k \to \infty} c(k) = 0$. Therefore, we give progressively more importance to minimizing disagreement. However, the rate at which happens is important. This is in turn captured by the last two conditions on the series sum, which prevent $c(k)$ from decreasing too fast, as this may prevent consensus.

The pseudo-code capturing the main steps of this iterative approach as outlines above is given in Algorithm 1. The basic steps of the proposed approach are summarized in Algorithm 1. Notice that at initialization, each agent i, $i = 1, \dots, m$, starts with some tentative value $x_i(0)$ which belongs to the local constraint set $\bigcap_{\delta \in S_i} X_i(\delta)$ of agent i, but not necessarily to $\bigcap_{i=1}^m \bigcap_{\delta \in S_i} X_i(\delta)$. One sensible choice for $x_i(0)$ is to set it such that $x_i(0) \in \arg\min_{x_i \in \bigcap_{\delta \in S_i} X_i(\delta)} f_i(x_i)$.

We will show that under suitable assumptions on the underlying communication protocol, Algorithm 1 converges, and agents reach consensus to an optimal solution of $\mathrm{P}_{\text{decision}}$. Note that under Assumption 2, by the Weierstrass' theorem (Proposition A.8, p. 625 in Bertsekas and Tsitsiklis (1989)), problem $\mathrm{P}_{\text{decision}}$ admits a non-empty set of minimizers; however, this set is not necessarily a singleton, and Algorithm 1 will be shown to return one element in the set of optimal solutions.

To impose these communications assumptions, notice that for each $k \geq 0$, the information exchange between the m agents can be represented by a directed graph (V, E_k), where the nodes $V = \{1, \dots, m\}$ are the agents and the set E_k of directed edges (j, i), indicating that at time k agent i receives information from agent j, is given by

$$E_k = \{(j, i) : \; a_j^i(k) > 0\}. \tag{16}$$

Algorithm 1 Distributed algorithm – Decision coupled problems

Initialization
  Set $\{a^i_j(k)\}_{k\geq 0}$, for all $i, j = 1, \dots, m$
  Collect dataset S_i, for all $i = 1, \dots, m$
  Choose $\{c(k)\}_{k\geq 0}$
  $k = 0$
  Consider $x_i(0) \in \bigcap_{\delta\in S_i} X_i(\delta)$, for all $i = 1, \dots, m$
For $i = 1, \dots, m$ **repeat until convergence**
  $z_i(k) = \sum_{j=1}^m a^i_j(k) x_j(k)$ ▷ Information vector
  $x_i(k+1) = \arg\min_{x_i \in \bigcap_{\delta\in S_i} X_i(\delta)} f_i(x_i) + \frac{1}{2c(k)} \|z_i(k) - x_i\|^2$ ▷ Decision update
  $k \leftarrow k + 1$

From (16), we set $a^i_j(k) = 0$ in the absence of communication. If $(j, i) \in E_k$ we say that j is a neighboring agent of i at time k. Under this set-up, Algorithm 1 provides a fully distributed implementation, where at iteration k each agent $i = 1, \dots, m$ receives information only from neighboring agents. Moreover, this information exchange is time-varying and may be occasionally absent. However, the following connectivity and communication assumption is made, where $E_\infty = \{(j, i) : (j, i) \in E_k$ for infinitely many $k\}$ denotes the set of edges (j, i) representing agent pairs that communicate directly infinitely often.

Assumption 5 (Connectivity and Communication)**.** The graph (V, E_∞) is strongly connected, i.e., for any two nodes there exists a path of directed edges that connects them. Moreover, there exists $T \geq 1$ such that for every $(j, i) \in E_\infty$, agent i receives information from a neighboring agent j at least once every consecutive T iterations.

Assumption 5 guarantees that any pair of agents communicates directly infinitely often, and the intercommunication interval is bounded. For further details on the interpretation of the imposed network structure the reader is referred to Nedíc and Ozdaglar (2009); Nedíc et al. (2010).

The following result shows that under the imposed structural and communication assumptions, Algorithm 1 converges and agents reach consensus, in the sense that their local estimates $x_i(k)$, $i = 1, \dots, m$, converge to some minimizer of problem $\mathrm{P}_{\mathrm{decision}}$.

Theorem 4. *Consider Assumptions 2 (with the additional requirement that also all $f_i(\cdot)$'s are convex), 3, 4, 5, and Algorithm 1. We have that*

$$\lim_{k\to\infty} \|x_i(k) - x^\star\| = 0, \text{ for all } i = 1, \dots, m, \tag{17}$$

where $x^\star$ is some minimizer of $\mathrm{P}_{\mathrm{decision}}$.

From an implementation point of view, agent i, $i = 1, \dots, m$, will terminate its update process if the absolute difference (relative difference can also be used) between two consecutive iterates $\|x_i(k+1) - x_i(k)\|$ keeps below some user-defined tolerance for a number of iterations equal to T (see Assumption 5) times the diameter of the graph (i.e., the greatest distance between any pair of nodes connected via an edge in E_∞). This is the worst case number of iterations required for an agent to (indirectly) receive some information from all others in the network; note that if an agent terminated the process at the first iteration where the desired tolerance is met, then convergence would not be guaranteed since its solution may still change as an effect of other agents updating their solutions.

It should be highlighted that the solution $x^\star$ returned by Algorithm 1 upon convergence (due to Theorem 4) is accompanied with the probabilistic certificates of Theorem 2. Therefore, we have constructed a distributed mechanism to determine the optimizer of a multi-agent problem, while keeping the dataset S_i, $i = 1, \dots, m$, of each agent private.

Note that algorithm presented in Algorithm 1 is the so called distributed proximal minimization algorithm, and the convergence result of Theorem 4 is due to Margellos et al. (2018). Other algorithms could be considered in place of that one, while still accompanying the resulting solution with the certificates of Section 3. This flexibility stems from the fact that, once we use data/scenarios for the uncertainty, any distributed consensus algorithm that is able to solve decision couple problems and allows for local constraint sets can be invoked, with the local constraint set of each agent i, $i = 1, \dots, m$, replaced by $\bigcap_{\delta\in S_i} X_i(\delta)$. As an example we refer to the distributed, projected subgradient algorithms of Nedíc et al. (2010); Zhu and Martínez (2012); these could be directly implemented in the proposed framework, by replacing the proximal minimization in the decision update step of Algorithm 1 with a projected subgradient one. More recent distributed algorithms employing a constant coefficient $c(k) = c$, $k \geq 0$, to improve convergence, Nedic et al. (2017), and able to handle local constraints, like those in Falsone and Prandini (2022); Shi et al. (2015); Li et al. (2019), can also be applied.

4.2 Constraint coupled problems

Now we turn our attention to constraint-coupled problems. Recall problem $\mathrm{P}_{\mathrm{constraint}}$, as presented in Section 2, and impose that Assumption 2 holds for $\mathrm{P}_{\mathrm{constraint}}$ under the variable assignments in Section 2.1. Similarly to the previous section each agents has its own objective function and its own local constraint set, but, differently from the previous section, each agent has also its own decision, which can be different from that of the other agents, but the decision of all the agents must collectively satisfy the coupling constraints $\sum_{i=1}^m g_i(x_i) \leq 0$

besides being feasible for the local ones and minimizing the sum of the local cost functions. We will discuss a distributed architecture that can be followed by agents in an iterative manner, so that the decisions the agents converge to satisfy these properties. However, before delving into the distributed resolution scheme, we need to introduce some notions of duality theory that are key for the subsequent developments.

Looking again at problem $P_{\text{constraint}}$, it is easy to see that $\sum_{i=1}^{m} g_i(x_i) \leq 0$ is somehow a complicating constraint. Indeed, removing it from the constraints and penalizing it into the objective function, would turn $P_{\text{constraint}}$ into m completely separable (smaller) optimization problems. This is the typical set-up where duality theory can be used to deal with the complicating constraint. To this end, let us introduce a vector $\lambda \in \mathbb{R}^p$ of p non-negative Lagrange multipliers and let us define a modified cost function

$$L(x_1, \ldots, x_m, \lambda) = \sum_{i=1}^{m} f_i(x_i) + \lambda^\top \left(\sum_{i=1}^{m} g_i(x_i) \right) = \sum_{i=1}^{m} f_i(x_i) + \lambda^\top g_i(x_i) \tag{18}$$

called Lagrangian, where the value of the complicating constraint is penalized through $\lambda \geq 0$. The hope is that, by properly selecting $\lambda \geq 0$, then the optimal solution to

$$\min_{\{x_i \in \mathcal{X}_i\}_{i=1}^m} L(x_1, \ldots, x_m, \lambda) = \min_{\{x_i \in \mathcal{X}_i\}_{i=1}^m} \sum_{i=1}^{m} f_i(x_i) + \lambda^\top g_i(x_i) = \sum_{i=1}^{m} \min_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda^\top g_i(x_i) \tag{19}$$

will also be optimal for problem $P_{\text{constraint}}$, where we used $\mathcal{X}_i = \bigcap_{\delta \in S_i} \tilde{X}_i(\delta)$ to ease the notation. Note that the optimal value of (19) is a function of $\lambda \geq 0$:

$$\varphi(\lambda) = \sum_{i=1}^{m} \varphi_i(\lambda) = \sum_{i=1}^{m} \min_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda^\top g_i(x_i) \tag{20}$$

and is named dual function. Each $\varphi_i(\lambda)$ is finite (as an effect of $\mathcal{X}_i$ being compact), concave (since $\varphi_i(\lambda)$ is the point-wise minimum of affine functions of λ), and contains all (and only!) information related to agent i. To find the appropriate value of $\lambda \geq 0$ we need to solve the so-called dual problem

$$\max_{\lambda \geq 0} \varphi(\lambda) = \max_{\lambda \geq 0} \sum_{i=1}^{m} \varphi_i(\lambda), \tag{21}$$

which is a convex problem since it aims at maximizing a concave function over a convex set. Under Assumption 2, an optimal solution $\lambda^\star \geq 0$ is guaranteed to exists (Bertsekas, 2015, Proposition 1.1.3), and any optimal solution to problem $P_{\text{constraint}}$ is also an optimal solution to (19) with $\lambda = \lambda^\star$ (Bertsekas, 2015, Proposition 1.1.2). Unfortunately, the converse is true if all $f_i(x_i)$'s are strictly convex, but it is not true in general where there may be optimal solutions to (19) with $\lambda = \lambda^\star$ which are not necessarily optimal (or not even feasible) for problem $P_{\text{constraint}}$. Luckily there are procedures to recover an optimal solution to problem $P_{\text{constraint}}$ while solving (21) with iterative schemes, see, e.g., (Shor, 1985, p. 117), when the $f_i(x_i)$'s are not strictly convex.

Up to now we have shown that we can construct a new optimization problem (the dual problem in (21)) which can be leveraged to obtain an optimal solution to the original problem $P_{\text{constraint}}$. However, problem (21), despite having a cost function that is separable across the agents, is still coupled because the agents have to agree on a common $\lambda \geq 0$. Luckily, problem (21) is a decision coupled problem, similar to problem P_{decision} introduced in Section 2, but with local constraint sets being deterministic and equal to $\{\lambda \in \mathbb{R}^p : \lambda \geq 0\}$ for all agents and with a maximization of a concave function in place of a minimization of a convex one. We can thus devise a distributed architecture, similar to the one introduced in Section 4.1, that can be followed by agents in an iterative manner to agree on a common solution to (21) and, at the same time, use that to solve the original problem $P_{\text{constraint}}$. Denote again by $k \in \mathbb{N}$ the iteration index, and let $\lambda_i(k)$ be a copy/estimate of the common dual vector λ agents seek to agree upon, maintained by agent i, $i = 1, \ldots, m$, at iteration k.

Information vector: At iteration k, each agent $i = 1, \ldots, m$, constructs the information vector

$$\ell_i(k) = \sum_{j=1}^{m} a_j^i(k) \lambda_j(k),$$

which can again be thought of as a weighted average of the estimates $\lambda_j(k)$, $j = 1, \ldots, m$, of the other agents at iteration k. The weights $a_j^i(k)$ have the exact same meaning of the corresponding weights in Section 4.1 and must also satisfy Assumption 3.

Decision update: At iteration k, after the information vector $\ell_i(k)$ has been computed, each agent i, $i = 1, \ldots, m$, should update its local estimate from $\lambda_i(k)$ using the following update

$$\lambda_i(k+1) = \arg\max_{\lambda_i \geq 0} \varphi_i(\lambda_i) - \frac{1}{2c(k)} \|\ell_i(k) - \lambda_i\|^2 = \arg\max_{\lambda_i \geq 0} \min_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda_i^\top g_i(x_i) - \frac{1}{2c(k)} \|\ell_i(k) - \lambda_i\|^2, \tag{22}$$

which, however, differently from Section 4.1 involves solving a max-min. To avoid such a max-min optimization at each step, one can approximate (22) as the following two steps

$$x_i(k+1) \in \arg\min_{x_i \in \mathcal{X}_i} f_i(x_i) + \ell_i(k)^\top g_i(x_i), \tag{23a}$$

$$\lambda_i(k+1) = \arg\max_{\lambda_i \geq 0} \lambda^\top g_i(x_i(k+1)) - \frac{1}{2c(k)} \|\ell_i(k) - \lambda_i\|^2, \tag{23b}$$

where, in (22), minimization over $x_i \in \mathcal{X}_i$ is carried out first, fixing $\lambda_i = \ell_i(k)$, and the maximization is carried out next, fixing $x_i = x_i(k+1)$. Note that the optimization problem in (23b) is very structured and it admits an explicit solution as

$$\lambda_i(k+1) = \max\{\ell_i(k) + c(k) g_i(x_i(k+1)), 0\}, \tag{24}$$

where $\max\{v, 0\}$ between a vector v and zero has to be intended as component-wise. Similarly to the discussion in Section 4.1, this way of updating $\lambda_i(k)$ trades between the maximization of agent i dual function $\varphi_i(\lambda_i)$ and the minimization of the disagreement between λ_i and the information vector $\ell_i(k)$ via the iteration-varying coefficient $c(k)$, which, also in this case, must satisfy Assumption 4, with the exact same interpretation.

Solution recovery: In case the agents objective functions $f_i(x_i)$'s are not strictly convex, the $x_i(k)$'s generated by the proposed scheme are not guaranteed to asymptotically form an optimal solution to problem $\mathrm{P}_{\text{constraint}}$, but, as briefly mentioned before, there exist a way of constructing an optimal solution to problem $\mathrm{P}_{\text{constraint}}$ starting from the $x_i(k)$'s generated by the proposed scheme. This involves the agents to maintain an auxiliary sequence $\hat{x}_i(k)$ and update it according to

$$\hat{x}_i(k+1) = \hat{x}_i(k) + \frac{c(k)}{\sum_{s=0}^{k} c(s)} (x_i(k+1) - \hat{x}_i(k)) = \frac{\sum_{s=0}^{k} c(s) x_i(s+1)}{\sum_{s=0}^{k} c(s)}, \tag{25}$$

where the first expression provides a nice recursive update for the proposed architecture, while the second expression provides an interpretation of $\hat{x}_i(k+1)$ as a $c(k)$-weighted convex combination of all $x_i(k)$'s explored by agent i across the iterations.

Algorithm 2 Distributed algorithm – Constraint coupled problems

Initialization
 Set $\{a^i_j(k)\}_{k\geq 0}$, for all $i, j = 1, \dots, m$
 Collect dataset S_i, for all $i = 1, \dots, m$
 Choose $\{c(k)\}_{k\geq 0}$
 $k = 0$
 Consider $\lambda_i(0) \geq 0$, for all $i = 1, \dots, m$
For $i = 1, \dots, m$ **repeat until convergence**
 $\ell_i(k) = \sum_{j=1}^{m} a^i_j(k) \lambda_j(k)$ ▹ Information vector
 $x_i(k+1) \in \arg\min_{x_i \in \bigcap_{\delta\in S_i} \tilde{\mathcal{X}}_i(\delta)} f_i(x_i) + \ell_i(k)^\top g_i(x_i)$ ▹ Decision x_i-update
 $\lambda_i(k+1) = \max\{\ell_i(k) + c(k) g_i(x_i(k+1)), 0\}$ ▹ Decision λ_i-update
 $\hat{x}_i(k+1) = \hat{x}_i(k) + \frac{c(k)}{\sum_{s=0}^{k} c(s)} (x_i(k+1) - \hat{x}_i(k))$ ▹ Recovery update
 $k \leftarrow k+1$

We report in Algorithm 2 the main steps of the distributed iterative scheme for the solution of constraint coupled problems presented in this section. Note that, during the initialization step, each agent chooses a value for $\lambda_i(0)$ which satisfies the non-negativity constraint, but is not necessarily the same of the other agents. A sensible choice for $\lambda_i(0)$ is to set $\lambda_i(0) = 0$ for all $i = 1, \dots, m$, so that, at $k = 0$ each agent sets $x_i(1) \in \arg\min_{x_i \in \bigcap_{\delta\in S_i} \tilde{\mathcal{X}}_i(\delta)} f_i(x_i)$ and if $g_i(x_i(1)) \leq 0$ for all $i = 1, \dots, m$, then the coupling constraint is already satisfied and $[x_1(1) \cdots x_m(1)]^\top$ is optimal for problem $\mathrm{P}_{\text{constraint}}$. Moreover, $\lambda_i(1) = 0$ for all $i = 1, \dots, m$ and, hence, $[x_1(k) \cdots x_m(k)]^\top$ is optimal for problem $\mathrm{P}_{\text{constraint}}$ for every $k \geq 0$. Let us highlight that, if the $f_i(x_i)$'s are known to be strictly convex, then the $\hat{x}_i(k)$'s are not needed.

Similarly to Section 4.1, for Algorithm 2 to work, we need to impose some requirements on the communication graph modeling the interactions among the agents. Specifically, also in this case, we require Assumption 5 to hold to ensure that any pair of agents communicate (directly or indirectly, through other agents) infinitely often, and that the intercommunication interval is bounded. The following result shows that under the imposed structural and communication assumptions, agents reach consensus on a common vector of Lagrange multipliers and are able to construct an optimal solution to problem $\mathrm{P}_{\text{constraint}}$.

Theorem 5. *Consider Assumptions 2 (with the additional requirement that also all $f_i(\cdot)$'s are convex), 3, 4, 5, and Algorithm 2. We have that*

$$\lim_{k\to\infty} \|\lambda_i(k) - \lambda^\star\| = 0, \text{ for all } i = 1, \dots, m, \tag{26}$$

where $\lambda^\star$ is an optimal solution to problem (21)*, and all limit points of vector $[\hat{x}_1(k) \cdots \hat{x}_m(k)]^\top$ (or $[x_1(k) \cdots x_m(k)]^\top$ in case the $f_i(x_i)$'s are strictly convex) are optimal solutions to problem* $\mathrm{P}_{\text{constraint}}$.

A termination criterion for Algorithm 2 would be similar to that of Algorithm 1, but would monitor the absolute/relative difference of the quantity $\|\lambda_i(k+1) - \lambda_i(k)\|$ across consecutive iterates.

Similarly to its counterpart in Section 4.1, any optimal solution obtained running Algorithm 2 is accompanied with the probabilistic certificates of Theorem 2. Therefore, we have introduced a distributed scheme to compute the optimal solution of a multi-agent constraint coupled problem while keeping the dataset S_i, $i = 1, \dots, m$, of each agent private.

Algorithm 2 presented above can be thought of as the distributed counterpart of the dual subgradient algorithm typically employed in the resolution of dual problems in a centralized set-up. The convergence property stated in Theorem 5 is rigorously proved in Falsone et al. (2017a) under slightly milder version of point 4 of Assumption 2. Other algorithms can be considered in place of Algorithm 2, while still accompanying the resulting solution with the certificates of Section 3, as they do not depend of the specific algorithmic strategy employed to get the optimal solution of the corresponding scenario problem. Indeed, by the time we use data/scenarios for the uncertainty, any distributed algorithm for constraint coupled problems that allows for local constraint sets can be invoked. For example, the algorithm proposed in Falsone and Prandini (2020) employs a similar scheme, but proposes a two-step update to solve (22) exactly when the coupling constraints are linear equalities, and not approximately as in Algorithm 2 and will inherit the probabilistic guarantees. Also the duality and relaxation-based approach proposed in Notarnicola and Notarstefano (2019) can be applied. More recent schemes, like Falsone et al. (2020); Falsone and Prandini (2023), leverage additional quantities to be maintained and exchanged among the agents but are able to use a constant penalty parameter $c(k) = c$, $k \geq 0$, instead of a vanishing one, which typically enhances the convergence speed, and will all inherit the probabilistic guarantees provided by our framework.

5 Concluding remarks

We considered cooperative optimization problems over networks, where agents seek to reach consensus on a common decision or the share of a common resource. We encompassed such problems under the setting of decision coupled and constraint coupled optimization problems, and provided an analysis framework for the case where the underlying environment is uncertain. In particular, we viewed the problem under a data driven lens, and considered data as a finite and private resource, that is not to be exchanged or shared among agents. We showed how to accompany agents' decision with probabilistic certificate on their feasibility properties, and discussed how such decisions can be computed in a distributed manner. To this end, we presented a general distributed architecture as a multi-agent information exchange mechanism, and illustrated it by outlining specific algorithms to solve decision and constraint couple optimization problems. The proposed framework is based on a minimal set of assumptions, and could be considered in tandem to other distributed optimization algorithms. Current research concentrates towards relaxing the data independence assumption across agents, thus broadening the applicability of the proposed results.

References

Baingana B, Mateos G and Giannakis G (2014). Proximal-gradient algorithms for tracking cascades over social networks. *IEEE Journal of Selected Topics in Signal Processing* 8 (4): 563–575.
Bertsekas D (2015). Convex optimization algorithms, Athena Scientific.
Bertsekas D and Tsitsiklis J (1989). Parallel and distributed computation: Numerical methods, Athena Scientific (republished in 1997).
Bolognani S, Carli R, Cavraro G and Zampieri S (2015). Distributed reactive power feedback control for voltage regulation and loss minimization. *IEEE Transactions on Automatic Control* 60 (4): 966–981.
Boyd S, Parikh N, Chu E, Peleato B and Eckstein J (2010). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3 (1): 1–122.
Calafiore G and Campi M (2006). The scenario approach to robust control design. *IEEE Transactions on Automatic Control* 51 (5): 742–753.
Campi M and Garatti S (2008). The exact feasibility of randomized solutions of uncertain convex programs. *SIAM Journal on Optimization* 19 (3): 1211–1230.
Campi M and Garatti S (2011). A sampling-and-discarding approach to chance-constrained optimization: feasibility and optimality. *Journal of Optimization Theory and Applications* 148 (2): 257–280.
Campi M and Garatti S (2018a). Introduction to the Scenario Approach, SIAM Series on Optimization.
Campi M and Garatti S (2018b). Wait-and-judge scenario optimization. *Math. Programming* 167 (1): 155–189.
Campi MC and Garatti S (2023). Compression, generalization and learning. *Journal of Machine Learning Research* 24 (339): 1–74.
Campi M, Garatti S and Prandini M (2009). The scenario approach for systems and control design. *Annual Reviews in Control* 33 (2): 149 – 157.
Campi M, Garatti S and Ramponi F (2015), Non-convex scenario optimization with application to system identification, Proceedings of the 54th IEEE Conference on Decision and Control, Osaka, Japan, 4023–4028.
Campi M, Garatti S and Ramponi F (2018). A general scenario theory for non-convex optimization and decision making. *IEEE Transactions on Automatic Control* 63 (12): 4067–4078.
Carlone L, Srivastava V, Bullo F and Calafiore GC (2014). Distributed random convex programming via constraints consensus. *SIAM Journal on Control and Optimization* 52 (1): 629–662.
Casagrande V and Boem F (2022), A distributed scenario-based stochastic MPC for fault-tolerant microgrid energy management, Proceedings of the 11th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes, Pafos, Cyprus, 704–709.
Chamanbaz M, Notarstefano G and Bouffanais R (2017), Randomized constraints consensus for distributed robust linear programming, Proceedings of the 20th IFAC World Congress, Toulouse, France, 5124–5129.
Falsone A and Prandini M (2020). A distributed dual proximal minimization algorithm for constraint-coupled optimization problems. *IEEE Control Systems Letters* 5 (1): 259–264.
Falsone A and Prandini M (2022). Distributed decision-coupled constrained optimization via proximal-tracking. *Automatica* 135: 109938.
Falsone A and Prandini M (2023). Augmented lagrangian tracking for distributed optimization with equality and inequality coupling constraints. *Automatica* 157: 111269.
Falsone A, Margellos K, Garatti S and Prandini M (2017a). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica* 84: 149 – 158. ISSN 0005-1098.
Falsone A, Margellos K, Garatti S and Prandini M (2017b). Dual decomposition for multi-agent distributed optimization with coupling constraints. *Automatica* 84: 149–158.
Falsone A, Notarnicola I, Notarstefano G and Prandini M (2020). Tracking-admm for distributed constraint-coupled optimization. *Automatica* 117: 108962.

Garatti S and Campi M (2013). Modulating Robustness in Control Design. *IEEE Control Systems* 33 (2): 36 – 51.

Garatti S and Campi M (2022). Risk and complexity in scenario optimization. *Math. Programming* 191 (1): 243 – 279.

Garatti S and Campi M (2025). Non-convex scenario optimization. *Math. Programming* 209: 557 – 608.

Jadbabaie A, Lin J and Morse S (2003). Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on Automatic Control* 48 (6): 988–1001.

Kariotoglou N, Margellos K and Lygeros J (2016). On the computational complexity and generalization properties of multi-stage and stage-wise coupled scenario programs. *Systems and Control Letters* 94: 63–69.

Krishnamoorthy D, Foss B and Skogestad S (2018), A distributed algorithm for scenario-based model predictive control using primal decomposition, Proceedings of the 10th IFAC Symposium on Advanced Control of Chemical Processes, Shenyang, China, 351–356.

Lee S and Nedic A (2013). Distributed random projection algorithm for convex optimization. *IEEE Journal of Selected Topics in Signal Processing* 7 (2): 221–229.

Lee S and Nedic A (2016). Asynchronous gossip-based random projection algorithms over networks. *IEEE Transactions on Automatic Control* 61 (4): 953–968.

Li Z, Shi W and Yan M (2019). A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing* 67 (17): 4494–4506.

Margellos K, Prandini M and Lygeros J (2015). On the connection between compression learning and scenario based single-stage and cascading optimization problems. *IEEE Transactions on Automatic Control* 60 (10): 2716–2721.

Margellos K, Falsone A, Garatti S and Prandini M (2018). Distributed constrained optimization and consensus in uncertain networks via proximal minimization. *IEEE Transactions on Automatic Control* 63 (5): 1372–1387.

Martínez S, Bullo F, Cortés J and Frazzoli E (2007). On synchronous robotic networks - Part I: Models, tasks, and complexity. *IEEE Transactions on Automatic Control* 52 (12): 2199–2213.

Mateos G and Giannakis G (2012). Distributed recursive least-squares: Stability and performance analysis. *IEEE Transactions on Signal Processing* 60 (7): 3740–3754.

Nedíc A and Ozdaglar A (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control* 54 (1): 48–61.

Nedíc A, Olshevsky A, Ozdaglar A and Tsitsiklis J (2009). On distributed averaging algorithms and quantization effects. *IEEE Transactions on Automatic Control* 54 (11): 2506–2517.

Nedíc A, Ozdaglar A and Parrilo P (2010). Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control* 55 (4): 922–938.

Nedic A, Olshevsky A and Shi W (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization* 27 (4): 2597–2633.

Notarnicola I and Notarstefano G (2017). Asynchronous distributed optimization via randomized dual proximal gradient. *IEEE Transactions on Automatic Control* 62 (5): 2095–2106.

Notarnicola I and Notarstefano G (2019). Constraint-coupled distributed optimization: A relaxation and duality approach. *IEEE Transactions on Control of Network Systems* 7 (1): 483–492.

Notarstefano G, Notarnicola I and Camisa A (2019). Distributed optimization for smart cyber-physical networks. *Foundations and Trends® in Systems and Control* 7 (3): 253–383. ISSN 2325-6818.

Olfati-Saber R and Murray R (2004). Consensus problems in networks of agents with switching topology and time delays. *IEEE Transactions on Automatic Control* 49 (9): 1520–1533.

Olshevsky A and Tsitsiklis J (2011). Convergence speed in distributed convergence and averaging. *SIAM Review* 53 (4): 747–772.

Sayin MO, Vanli ND, Kozat SS and Basar T (2017). Stochastic subgradient algorithms for strongly convex optimization over distributed networks. *IEEE Transactions on Network Science and Engineering* 4 (4): 248–260.

Schildbach G, Fagiano L and Morari M (2013). Randomized Solutions to Convex Programs with Multiple Chance Constraints. *SIAM Journal on Optimization* 23 (4): 2479 – 2501.

Shi W, Ling Q, Wu G and Yin W (2015). A proximal gradient algorithm for decentralized composite optimization. *IEEE Transactions on Signal Processing* 63 (22): 6013–6023.

Shor NZ (1985). Minimization methods for non-differentiable functions, Springer.

Towfic ZJ and Sayed AH (2014). Adaptive penalty-based distributed stochastic convex optimization. *IEEE Transactions on Signal Processing* 62 (15): 3924–3938.

Tsitsiklis J, Bertsekas D and Athans M (1986). Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Transactions on Automatic Control* 31 (9): 803–812.

Vidyasagar M (2002). A Theory of Learning and Generalization, 2nd ed. London, U.K.: Springer.

Zhang Y and Giannakis G (2016). Distributed stochastic market clearing with high-penetration wind power. *IEEE Transactions on Power Systems* 31 (2): 895–906.

Zhu M and Martínez S (2012). On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control* 57 (1): 151–164.