Distributed Computational Framework for Large-Scale Stochastic Convex Optimization

Vahab Rostampour [1,*] and **Tamás Keviczky** [2]

1   Engineering and Technology Institute Groningen (ENTEG), University of Groningen, Nijenborgh 4, 9747 AG Groningen, The Netherlands

2   Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands; T.Keviczky@tudelft.nl

*   Correspondence: v.rostampour@rug.nl; Tel.: +31-50-36-34484

Abstract: This paper presents a distributed computational framework for stochastic convex optimization problems using the so-called scenario approach. Such a problem arises, for example, in a large-scale network of interconnected linear systems with local and common uncertainties. Due to the large number of required scenarios to approximate the stochasticity of these problems, the stochastic optimization involves formulating a large-scale scenario program, which is in general computationally demanding. We present two novel ideas in this paper to address this issue. We first develop a technique to decompose the large-scale scenario program into distributed scenario programs that exchange a certain number of scenarios with each other to compute local decisions using the alternating direction method of multipliers (ADMM). We show the exactness of the decomposition with a-priori probabilistic guarantees for the desired level of constraint fulfillment for both local and common uncertainty sources. As our second contribution, we develop a so-called soft communication scheme based on a set parametrization technique together with the notion of probabilistically reliable sets to reduce the required communication between the subproblems. We show how to incorporate the probabilistic reliability notion into existing results and provide new guarantees for the desired level of constraint violations. Two different simulation studies of two types of interconnected network, namely dynamically coupled and coupling constraints, are presented to illustrate advantages of the proposed distributed framework.

Keywords: stochastic optimization; scenario convex program; distributed computation; distributed stochastic systems; distributed scenario program; decentralized scenario program; plug-and-play framework

1. Introduction

Stochastic optimization has attracted significant attention in the recent control literature, due to its ability to provide an alternative, often less conservative way to handle uncertainty in dynamical systems. Stochastic optimization takes into account the stochastic characteristics of the uncertainties and thereby the system constraints are treated in a probabilistic sense, i.e., using chance constraints [1]. Using stochastic optimization in the so-called model predictive control (MPC), one computes an optimal control sequence that minimizes a given objective function subject to the uncertain system dynamics model and chance constraints in a receding horizon fashion [2]. Chance constraints enable stochastic optimization to offer an alternative approach to achieve a less conservative solution compared to robust optimization [3], since it directly incorporates the trade-off between constraint feasibility and performance of objective function.

Distributed optimization has been an active research area in recent decades, due to its applicability to handle large-scale problems with constraints. In distributed optimization one replaces large-scale problems stemming from centralized optimization with several

smaller-scale problems that can be solved in parallel. These problems make use of information from other subproblems to formulate a local optimization problem. In the presence of uncertainties, however, the main challenge in the formulation of distributed optimization is how one should exchange information through a communication scheme among subproblems (see, e.g., [4], and references therein). This highlights the necessity of developing distributed strategies to cope with the uncertainties in subproblems while at the same time minimizing information exchange through a communication framework.

To handle uncertainties in distributed setting, some approaches are based on robustification [5]. Assuming that the uncertainty is bounded, a robust optimization problem is solved, leading to a decision that satisfies the constraints for all admissible values of the uncertainty. The resulting solution using such an approach tends to be conservative in many cases. Tube-based robust optimization, see for example [6] and the references therein, was considered in a decentralized setup (plug-and-play) in [7], and it has been recently extended to distributed control systems [8] for a collection of linear stochastic subsystems with independent dynamics.

1.1. Related Works

Although in [8] coupled chance constraints were considered separately at each sampling time, in this paper we consider a chance constraint on the feasibility of trajectories of dynamically coupled subsystems. Our approach is motivated by [7] to reduce the conservativeness of the control design. Other representative approaches for stochastic optimization of a single stochastic system include affine parametrization of the control policy [9], the randomized (scenario) approach [10–13], and the combined randomized and robust approach [14–16]. None of these approaches, to the best of our knowledge, have been considered in a distributed control setting.

This paper aims to develop a systematic approach to distributed stochastic optimization using the scenario-based technique. Scenario optimization approximates stochastic problem via the so-called scenario (sample) approach [17], and if the underlying optimization problem is convex with respect to the decision variables, finite sample guarantees can be provided. Following such an approach, the computation time for a realistic large-scale problem of interest becomes prohibitive, because the number of samples to be extracted tends to be high, and consequently leads to many constraints in the resulting optimization problem.

To overcome the computational burden caused by the large number of constraints, in [18,19] a heuristic sample-based approach was used in an iterative distributed fashion via dual decomposition such that all subsystems collaboratively optimize a global performance index. In another interesting work [20], a multi-agent consensus algorithm was presented to achieve consensus on a common value of the decision vector subject to random constraints such that a probabilistic bound on the tails of the consensus violation was also established. However, in most of the aforementioned references the aim to reduce communication among subsystems, which we refer to as agents, has not been addressed.

1.2. Contributions

Our work in this paper differs from the above references in two important aspects which have not been, to the best of our knowledge, considered in the literature. We develop two different distributed scenario optimization frameworks. In the first version, each agent generates its own scenarios of the local uncertainty sources, and takes into consideration an estimation of the neighboring possible state trajectories, such that using the so-called alternating direction method of multipliers (ADMM), all the agents start to improve their local estimation until they agree on the consistency of exchange scenarios in the proposed distributed setting. In the second version, we develop a new set-based communication setup together with the notion of probabilistically reliable information to reduce the communications between agents required by the proposed first version of distributed scenario optimization framework. To quantify the error introduced by such a new communication

scheme, we incorporate the probabilistic reliability notion into existing results and provide new guarantees for the desired level of constraint violations. To summarize the main contributions of this paper are as follows:

(a) By reformulating a centralized scenario optimization problem into a decomposable scenario program, we first present the exactness of decomposition and then, quantify the level of robustness of resulting solutions by providing new a-priori probabilistic guarantees for the desired level of constraint fulfillment under some mild conditions.

(b) To solve the proposed decomposable scenario program, we develop a so-called distributed scenario exchange scheme between subproblems (neighboring agents) using the alternating direction method of multipliers (ADMM). Such a new scheme is used to provide a distributed scenario optimization framework to handle two types of large-scale network problems in control literature, namely dynamically coupled and coupling constraints network of agents.

(c) To reduce the communication in the proposed distributed setting, we develop a novel inter-agent soft communication scheme based on a set parametrization technique together with the notion of probabilistically reliable set to reduce the required communication between the subproblems. We show how to incorporate the probabilistic reliability notion into existing results and provide new guarantees for the desired level of constraint violations.

(d) Using the so-called soft communication scheme, we present a distributed framework such that the neighboring agents just need to communication a set together with the level of reliability of the set. Each agent then solves a *local robust-communication scenario program* at each sampling time. We also establish a practical plug-and-play (PnP) distributed framework that considers network changes by agents which want to join or leave the network.

The structure of this paper is as follows. Section 2 describes a problem formulation leading to a large-scale scenario optimization problem. In Section 3, we first reformulate the large-scale scenario problem into a decomposable scenario problem. In Section 4, we then provide the results on equivalent relations and analyze robustness of the obtained solutions using the proposed distributed scenario exchange scheme. Section 5 introduces a novel inter-agent communication scheme between the subproblems, namely *soft communications*, and then proceeds to quantify the robustness of the proposed schemes. Using the proposed soft communication scheme, in Section 6 we establish a practical PnP distributed framework. Section 7 presents two different simulation studies, whereas in Section 8, we conclude this paper with some remarks and future work.

2. Problem Formulation

Without loss of generality, this section presents a mathematical problem formulation in the context of advanced control problem, namely stochastic model predictive control (SMPC), leading to a large-scale scenario optimization problem at each time step. Please note that in this paper we focus on the large-scale scenario problem instance from the optimization point of view and derive probabilistic guarantees for constraint fulfillment in a distributed setting. Analyzing the closed-loop asymptotic behavior of SMPC (e.g., stability and recursive feasibility) are not in the scope of this paper.

Consider a discrete-time uncertain linear system with additive disturbance in a compact form as follows:

$$x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k + C(\delta_k)w_k \,, \tag{1}$$

with a given initial condition $x_0 \in \mathbb{R}^n$. Here $k \in \mathcal{T} := \{0, 1, \cdots, T-1\}$ denotes the time instance, $x_k \in \mathcal{X} \subset \mathbb{R}^n$ and $u_k \in \mathcal{U} \subset \mathbb{R}^m$ correspond to the state and control input, respectively, and $w_k \in \mathbb{R}^p$ represents an additive disturbance. The system matrices $A(\delta_k) \in \mathbb{R}^{n \times n}$ and $B(\delta_k) \in \mathbb{R}^{n \times m}$ as well as $C(\delta_k) \in \mathbb{R}^{n \times p}$ are random, since they are known functions of an uncertain variable δ_k that influences the system parameters at each time step k. It is important to mention that given the initial measured state variable x_0,

the predicted future states are modeled using (1) at each sampling time k. For avoiding crowded notation, when there is no ambiguity, in the rest of the paper we will drop the conventional index for the predicted time steps, i.e., $k + \ell|k$, and simply use $k + \ell$. We also define $\mathcal{T}_+ := \{1, \cdots, T\}$ to denote the future predicted time steps.

Throughout this paper, there are three technical assumptions. Assumption 1 is a classical assumption where we define the probability spaces associated with the random variables and highlight the fact that in this paper we only require the availability of a "sufficient number" of samples of such random variables. Using such samples to formulate an optimization problem, Assumption 2 is given as a technical requirement to be able to use a so-called randomization technique and to guarantee a nonempty feasibility domain of such an optimization problem. Assumption 3 is focused on the structure of large-scale system dynamics of interest to be able to explore and exploit its structure for decomposition into an interconnection of smaller subsystems.

Assumption 1. *Random variables $w := \{w_k\}_{k \in \mathcal{T}}$ and $\delta := \{\delta_k\}_{k \in \mathcal{T}}$ are defined on probability spaces $(\mathcal{W}, \mathfrak{B}(\mathcal{W}), \mathbb{P}_w)$ and $(\Delta, \mathfrak{B}(\Delta), \mathbb{P}_\delta)$, respectively. w and δ are two independent random processes, where $\mathbb{P}_w$ and $\mathbb{P}_\delta$ are two different probability measures defined over $\mathcal{W}$ and Δ, respectively, and $\mathfrak{B}(\cdot)$ denotes a Borel σ-algebra. The support sets $\mathcal{W}$ and Δ of w and δ, respectively, together with their probability measures $\mathbb{P}_w$ and $\mathbb{P}_\delta$ are entirely generic. In fact, $\mathcal{W}$, Δ and $\mathbb{P}_w$, $\mathbb{P}_\delta$ do not need to be known explicitly. Instead, the only requirement is availability of a "sufficient number" of samples, which will become concrete in later parts of the paper. Such samples can be for instance obtained by a model learned from available historical data [21].*

The system in (1) is subject to constraints on the system state trajectories and control input. Consider the state and control input constraint sets to be compact convex in the following form

$$\mathcal{X} := \{x \in \mathbb{R}^n : Gx \leq g\}, \mathcal{U} := \{u \in \mathbb{R}^m : Hu \leq h\}, \tag{2}$$

where $G \in \mathbb{R}^{q \times n}$, $g \in \mathbb{R}^q$, and $H \in \mathbb{R}^{r \times m}$, $h \in \mathbb{R}^r$. Keeping the state inside a feasible set $\mathcal{X} \subset \mathbb{R}^n$ for the entire prediction horizon may be too conservative and result in loss of performance. In particular, this is the case when the best performance is achieved close to the boundary of $\mathcal{X}$, and thus, constraint violations will be unavoidable due to the fact that the system parameters in (1) are imperfect and uncertain. To tackle such a problem, we will consider chance constraints on the state trajectories to avoid violation of the state variable constraints probabilistically even if the disturbance w or uncertainty δ has unbounded support. Notice that a robust problem formulation [3] cannot cope with problems having an unbounded disturbance set.

Following the traditional MPC approach to find a stabilizing full-information controller that leads to admissible control inputs $u := \{u_k\}_{k \in \mathcal{T}}$ and satisfies the state constraints, one can rely on the standard assumption of the existence of a suitable pre-stabilizing control law, see, e.g., (Proposition 1, [7]). Based on [9], one can also employ a so-called parametrized feedback policy and split the control input to cope with the state prediction under uncertainties: $u_k = Kx_k + v_k$ with $v_k \in \mathbb{R}^m$ as a free correction input variable to compensate for disturbances and K is a given pre-stabilizing feedback gain for the nominal (uncertain-free) system (1).

The control objective is to minimize a cumulative quadratic stage cost of a finite horizon cost $J(\cdot) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ that is defined as follows:

$$J(x, u) = \mathbb{E}\left[\sum_{k=0}^{T-1}\left(x_k^\top Q x_k + u_k^\top R u_k\right) + x_T^\top P x_T\right], \tag{3}$$

with $Q \in \mathbb{R}^{n \times n}_{\succeq 0}$, and $R \in \mathbb{R}^{m \times m}_{\succ 0}$. Consider $x := \{x_k\}_{k \in \mathcal{T}}$, $(A, Q^{\frac{1}{2}})$ to be detectable and P to be the solution of the discrete-time Lyapunov equation:

$$\mathbb{E}[\, A_{cl}(\delta_k)^\top P A_{cl}(\delta_k) \,] + Q + K^\top R K - P \preceq 0, \tag{4}$$

for the closed-loop system with $A_{cl}(\delta_k) = A(\delta_k) + B(\delta_k)K$. Each stage cost term is taken in expectation $\mathbb{E}[\cdot]$, since the argument x_k is a random variable. Using $v = \{v_k\}_{k \in \mathcal{T}}$, consider now the following stochastic control (optimization) problem:

$$\min_{v \in \mathbb{R}^{Tm}} \quad J(x, u) \tag{5a}$$

$$\text{s.t.} \quad x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k + C(\delta_k)w_k, \tag{5b}$$

$$\mathbb{P}[\, x_{k+\ell} \in \mathcal{X}, \ell \in \mathcal{T}_+ \,] \geq 1 - \varepsilon, \tag{5c}$$

$$u_k = Kx_k + v_k \in \mathcal{U}, \quad \forall k \in \mathcal{T}, \tag{5d}$$

where x_0 is initialized based on the measured current state, and $\varepsilon \in (0, 1)$ is the admissible state constraint violation parameter of the large-scale system (1). The state trajectory $x_{k+\ell}, \forall \ell \in \mathcal{T}_+$, has a dependency on the random variables w and δ, and thus, the chance constraint can be interpreted as follows: the probability of violating the state constraint at the future time step $\ell \in \mathcal{T}_+$ is restricted to ε, given that the state of the system in (1) is measurable at each time step $k \in \mathcal{T}$. Even though $\mathcal{U}$ and $\mathcal{X}$ are compact convex sets, due to the chance constraint on the state trajectory, the feasible set of the optimization problem in (5) is a non-convex set, in general.

To handle the chance constraint (5c), we recall a scenario-based approximation [22]. w_k and δ_k at each sampling time $k \in \mathcal{T}$ are not necessarily independent and identically distributed (i.i.d.). In particular, they may have time-varying distributions and/or be correlated in time. We assume that a "sufficient number" of i.i.d. samples of the disturbance $w \in \mathcal{W}$ and $\delta \in \Delta$ can be obtained either empirically or by a random number generator. We denote the sets of given finite samples (scenarios) with $\mathcal{S}_w := \{w^1, \cdots, w^S\} \in \mathcal{W}^S$ and $\mathcal{S}_\delta := \{\delta^1, \cdots, \delta^S\} \in \Delta^S$, respectively. Following the approach in [23], we approximate the expected value of the objective function empirically by averaging the value of its argument for some number of different scenarios, which plays a tuning parameter role. Using $\bar{S}$ as the tuning parameter, consider $\bar{S}$ number of different scenarios of w and δ to build

$$\bar{\mathcal{S}}_{w,\delta} = \{(w^s, \delta^s) : w^s \in \mathcal{W}, \delta^s \in \Delta, s = 1, \cdots, \bar{S}\},$$

which has the cardinality $|\bar{\mathcal{S}}_{w,\delta}| = \bar{S}$. We then approximate the cost function empirically as follows:

$$J(x, u) = \mathbb{E}_{(w,\delta)}\left[\sum_{k=0}^{T-1} V(x_k(w_k, \delta_k), u_k)\right] \approx \frac{1}{\bar{S}} \sum_{(w^s, \delta^s) \in \mathcal{S}_{w,\delta}} \sum_{k=0}^{T-1} V(x_k(w_k^s, \delta_k^s), u_k),$$

where $V(x_k(w, \delta_k), u_k)$ represents a compact notation of the objective function $(x_k(w_k, \delta_k)^\top Q x_k(w_k, \delta_k) + u_k^\top R u_k) + x_T(w_k, \delta_k)^\top P x_T(w_k, \delta_k)$. Notice that $x_k(w_k, \delta_k)$ indicates the dependency of the state variables on the random variables.

We are now in a position to formulate an approximated version of the proposed stochastic optimization problem in (5) using the following finite horizon scenario program:

$$\min_{v \in \mathbb{R}^{Tm}} \quad J(x, u) \tag{6a}$$

$$\text{s.t.} \quad x_{k+1}^s = A(\delta_k^s)x_k^s + B(\delta_k^s)u_k^s + C(\delta_k^s)w_k^s, \tag{6b}$$

$$x_{k+\ell}^s \in \mathcal{X}, \ell \in \mathcal{T}_+, \forall w^s \in \mathcal{S}_w, \forall \delta^s \in \mathcal{S}_\delta, \tag{6c}$$

$$u_k^s = Kx_k^s + v_k \in \mathcal{U}, \quad \forall k \in \mathcal{T}, \tag{6d}$$

where superscript s indicates a particular sample realization. The solution of (6) is the optimal control input sequence $v^* = \{v_k^*, \cdots, v_{k+T-1}^*\}$. Based on the MPC paradigm, the current input is implemented as $u_k := Kx_k + v_k^*$ and we proceed in a receding horizon fashion. This means that the problem (6) is solved at each time step k by using the current measurement of the state x_k. Please note that new scenarios are needed to be generated at each sampling time $k \in \mathbb{N}_+$. It is important to note that the proposed large-scale scenario optimization problem (6) should be feasible and its feasibility domain must be a nonempty domain. This is a technical requirement (Assumption 1, [17]) to be able to use such a randomization technique. In case of infeasible solution, we need to generate new set of scenarios and resolve the problem.

Assumption 2. *The proposed large-scale scenario optimization problem (6) is feasible and its feasibility domain has a nonempty domain.*

Following Assumption 2 and based on the Weierstrass' theorem (Proposition A.8, [24]), the proposed large-scale scenario program (6) admits at least one optimal solution, and therefore the Slater's constraint qualification [25] holds for the proposed problem formulations. The key features of the optimization problem (6) are as follows: (1) there is no need to know the probability measures $\mathbb{P}_w$ and $\mathbb{P}_\delta$ explicitly, only the capability of obtaining random scenarios is enough, (2) formal results to quantify the level of approximations are available. In particular, the results follow the so-called scenario approach [17], which allows the binding a-priori of the violation probability of the obtained solution via (6).

In the following theorem, we restate the explicit theoretical bound of (Theorem 1, [17]), which measures the finite scenarios behavior of (6).

Theorem 1. *Let $\varepsilon, \beta \in (0,1)$ and $S \geq \mathsf{N}(\varepsilon, \beta, Tm)$, where*

$$\mathsf{N}(\varepsilon, \beta, Tm) := \min\left\{\mathbf{N} \in \mathbb{N} \,\middle|\, \sum_{i=0}^{Tm-1} \binom{\mathbf{N}}{i} \varepsilon^i (1-\varepsilon)^{\mathbf{N}-i} \leq \beta \right\}.$$

If the optimizer of problem (6) $v^* \in \mathbb{R}^{Tm}$ is applied to the discrete-time dynamical system (1) for a finite horizon of length T, then, with at least confidence $1 - \beta$, the original constraint (5c) is satisfied for all $k \in \mathcal{T}$ with probability more than $1 - \varepsilon$.

It was shown in [17] that the above bound is tight. The interpretation of Theorem 1 is as follows: when applying v^* in a finite horizon control problem, the probability of constraint violation of the state trajectory remains below ε with confidence $1 - \beta$:

$$\mathbb{P}^S\left[\mathcal{S}_w \in \mathcal{W}^S, \mathcal{S}_\delta \in \Delta^S \,:\, \text{Vio}(v^*) \leq \varepsilon \right] \geq 1 - \beta,$$

with $\text{Vio}(v^*) := \mathbb{P}[w \in \mathcal{W}, \delta \in \Delta : x_{k+\ell} = A_{cl}(\delta_k)x_k + B(\delta_k)v_k^* + C(\delta_k)w_k \notin \mathcal{X}, \ell \in \mathcal{T}_+ | x_k = x_0]$, where $A_{cl}(\delta_k) = A(\delta_k) + B(\delta_k)K$. It is worth mentioning that while the proposed constraint on the control input (6d) is also met in a probabilistic sense for all prediction time steps, except at the initial time step, due to the feedback parametrization and the nature of the scenario approach that appears in the proposed optimization problem (6). At the initial time step such a constraint (6d) is deterministic, and the superscript s can be dropped, as there is only one measured current state x_k. This holds for all the following proposed formulations.

3. Decomposed Problem Reformulation

Consider a partitioning of the system dynamics (1) through a decomposition into N subsystems and let $\mathcal{N} = \{1, 2, \cdots, N\}$ be the set of subsystem indices. The state variables x_k, control input signals u_k and the additive disturbance w_k can be considered to be $x_k = \text{col}_{i \in \mathcal{N}}(x_{i,k})$, $u_k = \text{col}_{i \in \mathcal{N}}(u_{i,k})$ and $w_k = \text{col}_{i \in \mathcal{N}}(w_{i,k})$, respectively, where $x_{i,k} \in \mathbb{R}^{n_i}$,

$u_{i,k} \in \mathbb{R}^{m_i}$, $w_{i,k} \in \mathbb{R}^{p_i}$, and $\sum_{i \in \mathcal{N}} n_i = n$, $\sum_{i \in \mathcal{N}} m_i = m$, $\sum_{i \in \mathcal{N}} p_i = p$. The following assumption is important to be able to partition the system parameters.

Assumption 3. *The control input and the additive disturbance of the subsystems are decoupled, e.g., $u_{i,k}$ and $w_{i,k}$ only affect subsystem $i \in \mathcal{N}$ for all $k \in \mathcal{T}$. Consider the state and control input constraint sets $\mathcal{X}$ and $\mathcal{U}$ of large-scale system dynamics (1) as defined in (2) to be $\mathcal{X} = \prod_{i \in \mathcal{N}} \mathcal{X}_i$, and $\mathcal{U} = \prod_{i \in \mathcal{N}} \mathcal{U}_i$ such that $\mathcal{X}_i$ and $\mathcal{U}_i$ for all subsystems $i \in \mathcal{N}$ can be in the following form:*

$$\mathcal{X}_i := \{x \in \mathbb{R}^{m_i} : G_i x \leq g_i\}, \mathcal{U}_i := \{u \in \mathbb{R}^{p_i} : H_i u \leq h_i\},$$

where $G = \text{diag}_{i \in \mathcal{N}}(G_i)$, $H = \text{diag}_{i \in \mathcal{N}}(H_i)$, $g = \text{col}_{i \in \mathcal{N}}(g_i)$ and $h = \text{col}_{i \in \mathcal{N}}(h_i)$. Also consider the objective function of each subsystem $i \in \mathcal{N}$ in the following form:

$$J_i(x_i, u_i) := \mathbb{E}\left[\sum_{k=0}^{T-1} \left(x_{i,k}^\top Q_i x_{i,k} + u_k^\top R_i u_{i,k} \right) + x_{i,T}^\top P_i x_{i,T} \right],$$

where $Q_i \in \mathbb{R}_{\succeq 0}^{n_i \times n_i}$, $R_i \in \mathbb{R}_{\succ 0}^{m_i \times m_i}$ such that $Q = \text{diag}_{i \in \mathcal{N}}(Q_i)$, and $R = \text{diag}_{i \in \mathcal{N}}(R_i)$. Please note that $x_i = \text{col}_{k \in \mathcal{T}}(x_{i,k})$ and $u_i = \text{col}_{k \in \mathcal{T}}(u_{i,k})$ such that $x = \text{col}_{i \in \mathcal{N}}(x_i)$ and $u = \text{col}_{i \in \mathcal{N}}(u_i)$.

It is important to note that under Assumption 3 there are no direct coupling constraints between each subsystem $i \in \mathcal{N}$ and its neighboring subsystems $j \in \mathcal{N}_i$. Instead, the subsystem $i \in \mathcal{N}$ is dynamically coupled with all its neighboring subsystems $j \in \mathcal{N}_i$ as it is presented in (8).

We refer to the additive disturbance $w_{i,k}$ as a local uncertainty source of each subsystem $i \in \mathcal{N}$, since it is assumed that it affects only the subsystem $i \in \mathcal{N}$. Motivated by an application to Smart Thermal Grids (STGs) in [26] where a network of agents is interconnected via an uncertain common resource pool between neighbors, we consider the uncertain variable δ_k as a common uncertainty source between all subsystems $i \in \mathcal{N}$. Such a STG application will also be presented in Section 7.2 as a second case study. Observe the fact that every common uncertain phenomenon can be considered to be a local uncertain variable, e.g., the outside weather condition for neighboring buildings. Therefore, we also consider having $\delta_k = \text{col}_{i \in \mathcal{N}}(\delta_{i,k})$ and refer to both random variables $w_{i,k}$ and $\delta_{i,k}$ as local uncertainty sources.

We are now able to decompose the large-scale system matrices $B(\delta_k) = \text{diag}_{i \in \mathcal{N}}(B_i(\delta_{i,k})) \in \mathbb{R}^{n \times m}$, $C(\delta_k) = \text{diag}_{i \in \mathcal{N}}(C_i(\delta_{i,k})) \in \mathbb{R}^{n \times p}$, and consider $A(\delta_k) \in \mathbb{R}^{n \times n}$ in the following form:

$$A(\delta_k) = \begin{bmatrix} A_{11}(\delta_{1,k}) & \cdots & A_{1N}(\delta_{1,k}) \\ \vdots & \ddots & \vdots \\ A_{N1}(\delta_{N,k}) & \cdots & A_{NN}(\delta_{N,k}) \end{bmatrix},$$

where $A_{ij}(\delta_{i,k}) \in \mathbb{R}^{n_i \times n_j}$, $B_i(\delta_{i,k}) \in \mathbb{R}^{n_i \times m_i}$, and $C_i(\delta_{i,k}) \in \mathbb{R}^{n_i \times p_i}$. Define the set of neighboring subsystems of subsystem i as follows:

$$\mathcal{N}_i = \left\{ j \in \mathcal{N} \backslash \{i\} \mid A_{ij}(\delta_{i,k}) \neq 0 \right\}, \tag{7}$$

where 0 denotes a matrix of all zeros with proper dimension. Please note that if subsystems are decoupled, they remain decoupled regardless of the uncertainties $\delta_{i,k}$ for all $i \in \mathcal{N}$. Consider now a large-scale network that consists of N interconnected subsystems, and each subsystem can be described by an uncertain discrete-time linear time-invariant system with additive disturbance of the form

$$\begin{cases} x_{i,k+1} &= A_{ii}(\delta_{i,k}) x_{i,k} + B_i(\delta_{i,k}) u_{i,k} + q_{i,k} \\ q_{i,k} &= \sum_{j \in \mathcal{N}_i} A_{ij}(\delta_{i,k}) x_{j,k} + C_i(\delta_{i,k}) w_{i,k} \end{cases}. \tag{8}$$

Following Assumption 3, one can consider a linear feedback gain matrix K_i for each subsystem $i \in \mathcal{N}$ such that $K = \text{diag}_{i \in \mathcal{N}}(K_i)$. Using K_i in each subsystem, we assume that there exists P_i for each subsystem $i \in \mathcal{N}$ such that $P = \text{diag}_{i \in \mathcal{N}}(P_i)$ preserves the condition in (4).

Define $\sigma_i := \text{col}_{s \in \{1, \cdots, S_i\}}(\sigma_i^s)$ such that $\sigma_i^s := \text{col}_{j \in \mathcal{N}_i}(z_{ij}^s)$, where $z_{ij}^s = \text{col}_{k \in \mathcal{T}}(z_{ij,k}^s) \in \mathbb{R}^{Tn_j}$, to be auxiliary decision variables, and using $v_i = \text{col}_{k \in \mathcal{T}}(v_{i,k}) \in \mathbb{R}^{Tm_i}$ such that $v = \text{col}_{i \in \mathcal{N}}(v_i)$, we are now able to reformulate the optimization problem in (6) into the following decomposable finite horizon scenario program:

$$\min_{\{\sigma_i, v_i\}_{i \in \mathcal{N}}} \quad \sum_{i \in \mathcal{N}} J_i(x_i, u_i) \tag{9a}$$

$$\text{s.t.} \quad x_{i,k+1}^s = A_{ii}(\delta_{i,k}^s) x_{i,k}^s + B_i(\delta_{i,k}^s) u_{i,k}^s + q_{i,k}^s, \tag{9b}$$

$$q_{i,k}^s = \sum_{j \in \mathcal{N}_i} A_{ij}(\delta_{i,k}^s) z_{ij,k}^s + C_i(\delta_{i,k}^s) w_{i,k}^s, \tag{9c}$$

$$z_{ij,k}^s = x_{j,k}^s, \forall x_j^s \in \mathcal{S}_{x_j}, \forall j \in \mathcal{N}_i, \tag{9d}$$

$$x_{i,k+\ell}^s \in \mathcal{X}_i, \ell \in \mathcal{T}_+, \forall w_i^s \in \mathcal{S}_{w_i}, \forall \delta_i^s \in \mathcal{S}_{\delta_i}, \tag{9e}$$

$$u_{i,k}^s = K_i x_{i,k}^s + v_{i,k} \in \mathcal{U}_i, \forall k \in \mathcal{T}, \forall i \in \mathcal{N} \tag{9f}$$

where $w_i = \text{col}_{k \in \mathcal{T}}(w_{i,k}) \in \mathcal{W}_i$ and $\delta_i = \text{col}_{k \in \mathcal{T}}(\delta_{i,k}) \in \Delta_i$ such that $\mathcal{W} = \prod_{i \in \mathcal{N}} \mathcal{W}_i$ and $\Delta = \prod_{i \in \mathcal{N}} \Delta_i$. The sets $\mathcal{S}_{w_i} := \{w_i^1, \cdots, w_i^{S_i}\} \in \mathcal{W}_i^{S_i}$ and $\mathcal{S}_{\delta_i} := \{\delta_i^1, \cdots, \delta_i^{S_i}\} \in \Delta_i^{S_i}$ denote sets of given finite samples (scenarios) of disturbance and uncertainties in each subsystem $i \in \mathcal{N}$, such that $\mathcal{S}_w = \prod_{i \in \mathcal{N}} \mathcal{S}_{w_i}$ and $\mathcal{S}_\delta = \prod_{i \in \mathcal{N}} \mathcal{S}_{\delta_i}$. It is important to highlight that the proposed optimization problem in (9) is decomposable into N subproblems with coupling only through the equality constraints (9d) between neighboring agents. The auxiliary variables $z_{ij,k}^s$ are defined between agent $i \in \mathcal{N}$ and its neighboring agent $j \in \mathcal{N}_j$ to represent the local estimation of each scenario information of the neighbor's state variable $x_j^s = \text{col}_{k \in \mathcal{T}}(x_{j,k}^s) \in \mathcal{X}_j^T := \mathbb{X}_j$ at each sampling time $k \in \mathcal{T}$. By taking into consideration that the interaction dynamics model $A_{ij}(\delta_{i,k}^s)$ by each neighboring agent $j \in \mathcal{N}_i$ is available for agent $i \in \mathcal{N}$, the only information that subsystem $i \in \mathcal{N}$ needs is a set of scenarios of the state variable $\mathcal{S}_{x_j} := \{x_j^1, \cdots, x_j^{S_i}\} \in \mathbb{X}_j^{S_i}$ at each sampling time $k \in \mathcal{T}$ from all its neighboring agents $j \in \mathcal{N}_i$. Please note that the cardinality of $\mathcal{S}_{x_j}$ is S_i, which is assigned by the agent $i \in \mathcal{N}$ for all neighboring agents $j \in \mathcal{N}_i$, and should be sent by the neighboring agents at each sampling time $k \in \mathcal{T}$, and this is known as the *hard communication scheme* between the neighboring agents.

In the following proposition, we provide a connection between the proposed optimization problem in (9) and the optimization problem in (6).

Proposition 1. *Given Assumption 3 and the block-diagonal structure for the state-feedback controller $K = diag_{i \in \mathcal{N}}(K_i)$ for the large-scale system dynamics (1), the optimization problem in (9) is an exact decomposition of the optimization problem in (6), such that the optimal objective value of the decomposed problem (9) is equal to the optimal objective of the problem (6).*

The proof is provided in the Appendix A.

An important key feature of the proposed decomposable scenario program in (9) compared to the centralized scenario problem in (6) is as follows. Such a decomposition yields a reduction of computation time complexity of the centralized scenario program compared to the decomposable scenario program (9). This however requires more communication between each subsystem, since at each time $k \in \mathcal{T}$ each agent $i \in \mathcal{N}$ should send to and receive from all the neighboring agents $j \in \mathcal{N}_i$ a set of their state variable scenarios $\mathcal{S}_{x_i}$ and $\mathcal{S}_{x_j}$, respectively.

Remark 1. *The proposed constraint* (9e) *represents an approximation of the following chance constraint on the state of each subsystem* $i \in \mathcal{N}$:

$$\mathbb{P}[\, x_{i,k+\ell} \in \mathcal{X}_i \,,\, \ell \in \mathcal{T}_+ \,] \geq 1 - \varepsilon_i \,, \tag{10}$$

where $\varepsilon_i \in (0,1)$ *is the admissible state constraint violation parameter of each subsystem* (8). *One can also consider* $\alpha_i = 1 - \varepsilon_i$ *as the desired level of state feasibility parameter of each subsystem* (8).

The following theorem can be considered to be the main result of this section to quantify the robustness of the solutions obtained by (9).

Theorem 2. *Let* $\varepsilon_i, \beta_i \in (0,1)$ *be chosen such that* $\varepsilon = \sum_{i\in\mathcal{N}} \varepsilon_i \in (0,1)$, $\beta = \sum_{i\in\mathcal{N}} \beta_i \in (0,1)$, *and* $N_{s_i} \geq \mathsf{N}(\varepsilon_i, \beta_i, Tm_i)$ *for all subsystem* $i \in \mathcal{N}$. *If* $v^* = \mathrm{col}_{i\in\mathcal{N}}(v_i^*)$, *the collection of the optimizers of problem* (9) *for all subsystem* $i \in \mathcal{N}$, *is applied to the discrete-time dynamical system* (1) *for a finite horizon of length T, then, with at least confidence* $1 - \beta$, *the original constraint* (5c) *is satisfied for all* $k \in \mathcal{T}$ *with probability more than* $1 - \varepsilon$.

The proof is provided in the Appendix B.

The interpretation of Theorem 2 is as follows. In the proposed decomposable scenario program (9), each subsystem $i \in \mathcal{N}$ can have a desired level of constraint violation ε_i and a desired level of confidence level $1 - \beta_i$. To keep the robustness level of the collection of solutions in a probabilistic sense (5c) for the discrete-time dynamical system (1), these choices have to follow a certain design rule, e.g., $\varepsilon = \sum_{i\in\mathcal{N}} \varepsilon_i \in (0,1)$ and $\beta = \sum_{i\in\mathcal{N}} \beta_i \in (0,1)$. This yields a fixed ε, β for the large-scale system (1) and the individual ε_i, β_i for each subsystem $i \in \mathcal{M}$. It is important to mention that to maintain the violation level for the large-scale system with many partitions, i.e., $|\mathcal{N}| = N \uparrow$, the violation level of individual agents needs to decrease, i.e., $\varepsilon_i \downarrow$, which may lead to conservative results for each subsystem, since the number of samples needs to increase, i.e., $S_i \uparrow$. Addressing such a limitation is subject of ongoing research work. The following corollary is a direct result of Theorem 2.

Corollary 1. *If the optimal solution* v_i^* *for each agent* $i \in \mathcal{N}$ *obtained via the proposed decomposable problem* (9) *is applied to the agent dynamical system* (8) *for a finite horizon of length T, then, with at least confidence* $1 - \beta_i$, *the original constraint* (10) *is satisfied for all* $k \in \mathcal{T}$ *with probability more than* $1 - \varepsilon_i$.

4. Distributed Scenario Optimization

In this section, we continue by developing a distributed framework using ADMM to solve the proposed decomposable formulation in (9). It has been proven that ADMM for this type of problem converges linearly [27]. We follow a similar approach as in [28,29] to solve the decomposable problem (9) in a distributed manner by extending to the scenario program. Define the local feasibility set denoted with $\mathcal{L}_i$ for all agents $i \in \mathcal{N}$:

$$\mathcal{L}_i := \left\{ \{\sigma_i, v_i\} \,\middle|\, u_{i,k}^s = K_i x_{i,k}^s + v_{i,k} \in \mathcal{U}_i, \forall k \in \mathcal{T}, \right.$$
$$x_{i,k+1}^s = A_{ii}(\delta_{i,k}^s) x_{i,k}^s + B_i(\delta_{i,k}^s) u_{i,k}^s + q_{i,k}^s,$$
$$q_{i,k}^s = \sum_{j\in\mathcal{N}_i} A_{ij}(\delta_{i,k}^s) z_{ij,k}^s + C_i(\delta_{i,k}^s) w_{i,k}^s,$$
$$\left. x_{i,k+\ell}^s \in \mathcal{X}_i, \ell \in \mathcal{T}_+, \forall w_i^s \in \mathcal{S}_{w_i}, \forall \delta_i^s \in \mathcal{S}_{\delta_i} \right\},$$

where all the system parameters together with the local scenario sets, $\mathcal{S}_{w_i}, \mathcal{S}_{\delta_i}$, are fixed and available locally for each agent $i \in \mathcal{N}$ at each sampling time $k \in \mathcal{T}$. We are now able to define the augmented Lagrangian function as follows:

$$L(\{\sigma_i, v_i\}_{i \in \mathcal{N}}) := \sum_{i \in \mathcal{N}} \left(J_i(x_i, u_i) + I_{\mathcal{L}_i}(\{\sigma_i, v_i\}) + \sum_{j \in \mathcal{N}_i} \sum_{s=1}^{S_i} \left(\frac{\mu}{2} \left\| z_{ij}^s - x_j^s + \frac{\Lambda_{ij}^s}{\mu} \right\|_2^2 + \frac{1}{2\mu} \left\| \Lambda_{ij}^s \right\|_2^2 \right) \right),$$

where $I_{\mathcal{L}_i}(\{\sigma_i, v_i\})$ is a convex indicator function for the local constraint set $\mathcal{L}_i$ that maps to infinity if one of the constraints is violated, and to zero otherwise. Step size μ is a fixed constant, and multipliers $\Lambda_{ij}^s \in \mathbb{R}^{n_j}$ are introduced for every coupling constraint of each scenario. The indicator function makes sure all local constraints are satisfied, and the squared norm penalty term forces constraint (9d) to be satisfied.

We now describe the steps of the ADMM algorithm as follows:

(1) Update primal variables

The multipliers Λ_{ij}^s and the local estimation of neighboring state scenario z_{ij}^s for all scenarios $s = 1, \cdots, S_i$ and for all the neighbors $j \in \mathcal{N}_i$ are fixed at their value of the previous iteration. Consider the following minimization problem for all $i \in \mathcal{N}$:

$$v_i^{(l+1)} = \arg \min_{v_i} \left\{ J_i(x_i, u_i) + I_{\mathcal{L}_i}(\{\sigma_i^{(l)}, v_i\}) + \sum_{j \in \mathcal{N}_i} \sum_{s=1}^{S_j} \left(\frac{\mu}{2} \left\| z_{ji}^{s,(l)} - x_i^s + \frac{\Lambda_{ji}^{s,(l)}}{\mu} \right\|_2^2 \right) \right\}. \quad (11)$$

It is important to note that the augmented part of the aforementioned step in each agent $i \in \mathcal{N}$ refers to the coupling constraints from the neighboring agent $j \in \mathcal{N}_i$ point of view and $z_{ji}^{s,(l)}$ is the estimation of neighboring agent $j \in \mathcal{N}_i$ from the state variable scenario of agent $i \in \mathcal{N}$ for all $s \in \{1, \cdots, S_j\}$. Since the minimization is only in v_i, all terms of Λ_{ij}^s drop out. This results in $|\mathcal{N}| = N$ separate convex programs.

(2) Update and exchange the set of state scenarios

The resulting primarily decision variables $v_i^{(l+1)}$ for all $i \in \mathcal{N}$ are used to first update and then exchange the set of state scenarios. Please note that each agent only needs to communicate the updated local state scenarios set to all its neighboring agents and receive the updated set of all its neighboring state scenarios. Consider now the following steps which we refer to as the Scenario Updating Steps (SUS) for a generic agent $i \in \mathcal{N}$ as follows:

$$\begin{cases} x_{i,k+1}^{s,(l+1)} = A_{ii}(\delta_{i,k}^s) x_{i,k}^{s,(l+1)} + B_i(\delta_{i,k}^s) u_{i,k}^{s,(l+1)} + q_{i,k}^{s,(l)} \\ q_{i,k}^{s,(l)} = \sum_{j \in \mathcal{N}_i} A_{ij}(\delta_{i,k}^s) z_{ij,k}^{s,(l)} + C_i(\delta_{i,k}^s) w_{i,k}^s \\ u_{i,k}^{s,(l+1)} = K_i x_{i,k}^{s,(l+1)} + v_{i,k}^{(l+1)}, \forall k \in \mathcal{T} \end{cases} \quad (12)$$

By repeating SUS for all local scenarios, $\forall w_i^s \in \mathcal{S}_{w_i}$ and $\forall \delta_i^s \in \mathcal{S}_{\delta_i}$, the agent $i \in \mathcal{N}$ should first update the set of its local state scenarios $x_i^{s,(l+1)} \in \mathcal{S}_{x_i}^{(l+1)}$ and send it to all its neighboring agents. Then, the agent $i \in \mathcal{N}$ should receive the updated set of all its neighboring state scenarios $\mathcal{S}_{x_j}^{(l+1)}$. It is important to note that the set of local scenarios, $\mathcal{S}_{w_i}$ and $\mathcal{S}_{\delta_i}$, is fixed for all the SUS steps and iteration steps, i.e., $l \in \mathbb{N}_+$.

(3) Update and exchange the state estimation variables

Using the updated set of all neighboring state scenarios $\mathcal{S}_{x_j}^{(l+1)}$ and the updated primal variables, consider the following projection problem for all $i \in \mathcal{N}$:

$$\sigma_i^{(l+1)} = \arg \min_{\sigma_i} \left\{ I_{\mathcal{L}_i}(\{\sigma_i, v_i^{(l+1)}\}) + \sum_{j \in \mathcal{N}_i} \sum_{s=1}^{S_i} \left(\frac{\mu}{2} \left\| z_{ij}^s - x_j^{s,(l+1)} + \frac{\Lambda_{ij}^{s,(l)}}{\mu} \right\|_2^2 \right) \right\}, \quad (13)$$

Since the projection is only in σ_i, all terms of Λ_{ij}^s drop out and this results in $|\mathcal{N}| = N$ separate convex programs. Each agent $i \in \mathcal{N}$ should now send the updated local estimation of the neighboring state variable $z_{ij}^{s,(l+1)}$ to its neighbor $j \in \mathcal{N}_i$.

(4) Update and exchange the multiplier variables

The multipliers are updated as follows $\forall i \in \mathcal{N}, \forall j \in \mathcal{N}_i$, and $s = 1, \cdots, S_i$:

$$\Lambda_{ij}^{s,(l+1)} = \Lambda_{ij}^{s,(l)} + \mu\left(z_{ij}^{s,(l+1)} - x_j^{s,(l+1)}\right). \tag{14}$$

Notice that no information exchange is needed for the update of the multiplier, since agents receive the updated set of all their neighboring state scenarios $x_j^{s,(l+1)} \in \mathcal{S}_{x_j}^{(l+1)}$. As the final step, each agent $i \in \mathcal{N}$ should receive the updated multiplier variable $\Lambda_{ji}^{s,(l+1)}$ from all its neighboring agents $j \in \mathcal{N}_i$ for all the scenarios $s = 1, \cdots, S_j$.

In Algorithm 1, we summarize the proposed distributed scenario exchange framework using ADMM algorithm to illustrate the calculation and communication steps for each agent $i \in \mathcal{N}$. Each agent needs to solve two small-scale scenario programs in Lines 4 and 8 at each iteration that can be considered to be the highest computational cost in the proposed algorithm. In Line 5 each agent first updates $\mathcal{S}_{x_i}^{(l+1)}$ using SUS, which works with only simple operations, e.g., addition, subtraction and scaling, and then, it broadcasts and receives the updated local $\mathcal{S}_{x_i}^{(l+1)}$ and $\mathcal{S}_{x_j}^{(l+1)}$ to and from the neighbors, respectively. Finally, each agent in Lines 9 and 11 broadcasts the updated local estimation from the neighboring agents state variable scenarios and the related multipliers.

Algorithm 1 Distributed Scenario Exchange Algorithm

1: **for all** $i \in \mathcal{N}$ **do**

2:     **initialize:** $x_{i,0} \in \mathcal{X}_i$, $l = 0$, $\Lambda_{ij}^{s,(l)}$, $z_{ij}^{s,(l)}$, $z_{ji}^{s,(l)}$, $\forall s \in \{1, \cdots, S_i\}$, and $\forall j \in \mathcal{N}_i$

3:     **while** $\eta_i^{(l)} > \epsilon_i^{\mathrm{des}}$ using (15) **do**

4:         **Update** $v_i^{(l+1)}$ using (11)

5:         **Update** $\mathcal{S}_{x_i}^{(l+1)}$ using SUS procedure (12)

6:         **Broadcast** $\mathcal{S}_{x_i}^{(l+1)}$ to all $j \in \mathcal{N}_i$

7:         **Receive** $\mathcal{S}_{x_j}^{(l+1)}$ from all $j \in \mathcal{N}_i$

8:         **Update** $z_{ij}^{s,(l+1)}$ using (13) for all $j \in \mathcal{N}_i$ and for all $s \in \{1, \cdots, S_i\}$,

9:         **Broadcast** $z_{ij}^{s,(l+1)}$ to all $j \in \mathcal{N}_i$

10:        **Update** $\Lambda_{ij}^{s,(l+1)}$ using (14) for all $j \in \mathcal{N}_i$ and for all $s \in \{1, \cdots, S_i\}$

11:        **Broadcast** $\Lambda_{ij}^{s,(l+1)}$ to all $j \in \mathcal{N}_i$ and for all $s \in \{1, \cdots, S_i\}$

12:        **set** $l \leftarrow l + 1$

13:     **output:** $v_i^* = v_i^{l+1}$

We define the agreement on the local estimation from the state variable scenario of the neighbors z_{ij}^s as the convergence criteria of the proposed ADMM algorithm for each agent $i \in \mathcal{N}$ as follows:

$$\eta_i^{(l)} = \sum_{j \in \mathcal{N}_i} \sum_{s=1}^{S_i} \left(\frac{\mu}{2} \left\| z_{ij}^{s,(l)} - x_j^{s,(l)} \right\|_2^2 \right), \tag{15}$$

and the overall convergence criteria is $\eta^{(l)} := \sum_{i \in \mathcal{N}} \eta_i^{(l)}$. If the residual sequence $\left\{ \eta^{(l)} \right\}_{l=1}^{+\infty}$ is sufficiently small, less than or equal to $\sum_{i \in \mathcal{N}} \epsilon_i^{\mathrm{des}}$ where $\epsilon_i^{\mathrm{des}}$ is the desired level of convergence error for each agent $i \in \mathcal{N}$, then all neighboring agents have reached a consistent set of state variable scenarios. In light of Assumption 2, we can now provide convergence of Algorithm 1 based on the results in [30].

Theorem 3. *Assume that Slater's condition [25] holds for the decomposable scenario optimization problem* (9)*, and consider the iterative steps given in Algorithm 1. Then the following statements hold:*

- *The residual sequence $\{\eta^{(k)}\}_{k=0}^{+\infty}$ tends to 0 in a non-increasing way as l goes to $+\infty$, and consequently, for all $j \in \mathcal{N}_j$, and each $i \in \mathcal{N}$:*

$$z_{ij}^{s,(+\infty)} = x_j^{s,(+\infty)}.$$

- *The sequence $\{v_i^{(l)}\}_{\forall i \in \mathcal{N}}$ generated by Algorithm 1 converges to an optimal solution $\{v_i^*\}_{\forall i \in \mathcal{N}}$ of the decomposable scenario program* (9) *as l tends to $+\infty$.*

Proof. The theorem follows from [30] that studies the convergence of a standard ADMM problem. The details are omitted for brevity. □

Remark 2. *The proposed distributed scenario exchange algorithm uses a Gauss–Siedel update on the primal variables and the set of state variable scenarios, after which the multiplier variables are updated. Since either the primal or the set of state variable scenarios are fixed in the Gauss–Siedel steps, the problem can be distributed for all the agents. The main advantage to distribute a large-scale scenario optimization problem* (6) *is the ability of finding local solutions for each agent based on the information received in the previous iteration. Such calculations can therefore be carried out in parallel. Although an actual parallel implementation is outside the scope of this work, it is important to mention that the proposed algorithm is amenable to such an implementation. ADMM algorithms typically need a large number of iterations to converge to high accuracy, so the local agent problems need to be solved many times before finding a good enough solution. Thus, the ADMM approach without parallelization might not be the quickest method to solve the large-scale scenario program. Such a distributed framework is advantageous especially when the global scenario optimization problem is too large and cannot be solved within polynomial time due to the curse of dimensionality or memory limitations and computational constraints.*

We next present the overall steps needed to execute the proposed distributed scenario optimization. Algorithm 2 summarizes all the steps such that after decomposing the large-scale dynamical system (1) and determining the index sets of the neighboring agents, each agent first starts to generate some scenarios of its local uncertainty sources to approximate its cost function and achieve the feasibility of its chance constraint with high confidence level following Theorem 2. Next, all the agents solve their own local problem using the proposed distributed scenario exchange scheme (Algorithm 1) to obtain their local optimal solution. To execute Algorithm 1, it is assumed that the feedback control gain matrices K_i for all agents $i \in \mathcal{N}$ are given (4), and the coupling terms $A_{ij}(\delta_{i,k})$ are known between each agent $i \in \mathcal{N}$ and its neighboring agents $j \in \mathcal{N}_i$. Using the obtained solution via Algorithm 1, each agent generates a new set of local state variable scenarios to send and receive to and from all its neighboring agents, respectively. Finally, the first optimal control

input is applied to the real system and the new state variables are measured, and proceed in the classical receding horizon fashion.

Remark 3. *The proposed decomposable scenario program in* (9) *is a general deterministic optimization problem for each agent that are coupled via* (9d). *Therefore, the proposed technique to solve* (9) *in Algorithm* 1, *namely the distributed scenario exchange scheme, can be easily considered to be a general solution method to solve the agents' problem with the nonempty intersection between their local feasible sets. Such a case where agents are only coupled via coupling constraints has been presented in Section* 7.2 *as a second case study.*

Algorithm 2 Distributed Scenario Optimization (DSO)

1: **Decompose** the large-scale dynamical system (1) into N agents as the proposed form in (8)

2: **Determine** the index set of neighboring agents $\mathcal{N}_i$ for each agent $i \in \mathcal{N}$

3: **for all** agent $i \in \mathcal{N}$ **do**

4:    **fix** initial state $x_{i,0} \in \mathcal{X}_i$, $\varepsilon_i \in (0,1)$, and $\beta_i \in (0,1)$ such that $\varepsilon = \sum_{i \in \mathcal{N}} \varepsilon_i \in (0,1)$, $\beta = \sum_{i \in \mathcal{N}} \beta_i \in (0,1)$

5:    **determine** $\bar{S}_i \in (0,+\infty)$ to approximate the objective function, and S_i following Theorem 2 to approximate

        the chance constraint (10) in an equivalent sense

6:    **generate** $\bar{S}_i$, S_i scenarios of w_i, δ_i to determine the sets of $\bar{S}_{w_i,\delta_i}$ and $\mathcal{S}_{w_i}$, $\mathcal{S}_{\delta_i}$

7:    **solve** the proposed optimization problem in (9) using the proposed distributed scenario exchange

        algorithm (Algorithm 1) and determine v_i^*

8:    **generate** S_i scenarios of x_i using SUS procedure (12) and taking into consideration v_i^* and $\mathcal{S}_{w_i}$, $\mathcal{S}_{\delta_i}$

9:    **send** the set $\mathcal{S}_{x_i}$ to all neighboring agents $j \in \mathcal{N}_i$

10:    **receive** the sets $\mathcal{S}_{x_j}$ from all neighboring agents $j \in \mathcal{N}_i$

11:    **apply** the first input of solution $u_{i,k}^* = K_i x_{i,k} + v_{i,k}^*$ into the uncertain subsystem (8)

12:    **measure** the state and substitute it as the initial state of the next step $x_{i,0}$

13:    **set** $k \leftarrow k+1$

14:    **go to** Step (4)

---

5. Information Exchange Scheme

When the proposed distributed scenario optimization framework (Algorithm 2) is applied to the large-scale scenario program (6), all neighboring agent $j \in \mathcal{N}_i$ of the agent $i \in \mathcal{N}$ should send a set of scenarios of the state variable $\mathcal{S}_{x_j} := \{x_j^1, \cdots, x_j^{S_i}\} \in \mathbb{X}_j^{S_i}$ to agent i at each sampling time $k \in \mathcal{T}$ following the proposed distributed scenario exchange scheme in Algorithm 1. Based on ADMM in Algorithm 1, it may require a large number of iterations to achieve an agreement between neighboring agents on the consistency of exchange scenarios, which may also turn out to be too costly in terms of required communication bandwidth. To address this shortcoming, we propose a set-based information exchange scheme which is referred to as a *soft communication protocol*.

We now describe the soft communication protocol in more detail. The neighboring agent $j \in \mathcal{N}_i$ must first generate $\tilde{S}_j$ samples of x_j in order to build the set $\mathcal{S}_{x_j}$. It is important to notice that in the soft communication protocol the number $\tilde{S}_j$ of samples generated by agent j may be different than the one needed by agent i, which is S_i, as will be remarked later. Let us then introduce $\mathcal{B}_j \subset \mathbb{R}^{n_j}$ as a bounded set containing all the elements of $\mathcal{S}_{x_j}$. We assume for simplicity that $\mathcal{B}_j$ is an axis-aligned hyper-rectangular set [14]. This is not a restrictive assumption and any convex set, e.g., ellipsoids and polytopes, could have been chosen instead as described in [26]. We can define $\mathcal{B}_j := [-b_j, b_j]$ as an interval, where the vector $b_j \in \mathbb{R}^{n_j}$ defines the hyper-rectangle bounds.

Consider now the following optimization problem that aims to determine the set $\mathcal{B}_j$ with minimal volume:

$$
\begin{cases}
\min_{b_j \in \mathbb{R}^{n_j}} & \|b_j\|_1 \\
\text{s.t.} & x_j^l \in [-b_j, b_j] , \ \forall x_j^l \in \mathcal{S}_{x_j} , \\
& l = 1, \cdots, \tilde{S}_j
\end{cases}
\tag{16}
$$

where $\tilde{S}_j$ is the number of samples $x_j \in \mathcal{S}_{x_j}$ that neighboring agent j must take into account in order to determine $\mathcal{B}_j$. If we denote by $\tilde{\mathcal{B}}_j = [-\tilde{b}_j, \tilde{b}_j]$ the optimal solution of (16) computed by the neighbor agent j, then for implementing the soft communication protocol, agent j needs to communicate only the vector $\tilde{b}_j$ along with the level of reliability $\tilde{\alpha}_j$ to the agent i.

Definition 1. *A set $\tilde{\mathcal{B}}_j$ is called $\tilde{\alpha}_j-$reliable if*

$$
\mathbb{P}\big[x_j \in \mathbb{X}_j \ : \ x_j \notin [-\tilde{b}_j, \tilde{b}_j]\big] \leq 1 - \tilde{\alpha}_j ,
\tag{17}
$$

and we refer to $\tilde{\alpha}_j$ as the level of reliability of the set $\tilde{\mathcal{B}}_j$.

We now provide the following theorem to determine $\tilde{\alpha}_j$ as the level of reliability of the set $\tilde{\mathcal{B}}_j$.

Theorem 4. *Fix $\tilde{\beta}_j \in (0,1)$ and let*

$$
\tilde{\alpha}_j = \sqrt[\kappa]{\frac{\tilde{\beta}_j}{\binom{\tilde{S}_j}{n_j}}} .
\tag{18}
$$

where $\kappa = \tilde{S}_j - n_j$ is the degree of the root. We then have

$$
\mathbb{P}^{\tilde{S}_j}\Big\{ \{x_j^1, \cdots, x_j^{\tilde{S}_j}\} \in \mathbb{X}_j^{\tilde{S}_j} :
\tag{19}
$$

$$
\mathbb{P}\big[x_j \in \mathbb{X}_j \ : \ x_j \notin [-\tilde{b}_j, \tilde{b}_j]\big] \leq 1 - \tilde{\alpha}_j \Big\} \geq 1 - \tilde{\beta}_j .
$$

The proof is provided in the Appendix C.

Theorem 4 implies that given an hypothetical new sample $x_j \in \mathbb{X}_j$, agent $j \in \mathcal{N}_i$ has a confidence of at least $1 - \tilde{\beta}_j$ that the probability of $x_j \in \tilde{\mathcal{B}}_j = [-\tilde{b}_j, \tilde{b}_j]$ is at least $\tilde{\alpha}_j$. Therefore, one can *rely on $\tilde{\mathcal{B}}_j$ up to $\tilde{\alpha}_j$ probability*. The number of samples $\tilde{S}_j$ in the proposed formulation (16) is a design parameter chosen by the neighboring agent $j \in \mathcal{N}_i$. We however remark that one can also set a given $\tilde{\alpha}_j$ as the desired level of reliability and obtain from (18) the required number of samples $\tilde{S}_i$.

When an agent $i \in \mathcal{N}$ and its neighbor $j \in \mathcal{N}_i$ adopt the soft communication scheme, there is an important effect on the probabilistic feasibility of agent i, following Remark 1. Such a scheme introduces some level of stochasticity on the probabilistic feasibility of

agent i, due to the fact that the neighboring information is only *probabilistically reliable*. This will affect the local probabilistic robustness guarantee of feasibility as discussed in Theorem 2 and consequently in Theorem 1. To accommodate the level of reliability of neighboring information, we need to marginalize a joint cumulative distribution function (cdf) probability of x_i and the generic sample $x_j \in \mathbb{X}_j$ appearing in Theorem 4. We thus have the following theorem, which can be regarded as the main theoretical result of this section.

Theorem 5. *Given $\tilde{\alpha}_j \in (0,1)$ for all $j \in \mathcal{N}_i$ and a fixed $\alpha_i = 1 - \varepsilon_i \in (0,1)$, the state trajectory of a generic agent $i \in \mathcal{N}$ is probabilistically $\bar{\alpha}_i$–feasible for all $w_i \in \mathcal{W}_i$, $\delta_i \in \Delta_i$, i.e.,*

$$\mathbb{P}[x_{i,k+\ell} \in \mathcal{X}_i, \, \ell \in \mathcal{T}_+] \geq \bar{\alpha}_i, \tag{20}$$

where $\bar{\alpha}_i = 1 - \frac{1-\alpha_i}{\tilde{\alpha}_i}$ such that $\tilde{\alpha}_i = \prod_{j \in \mathcal{N}_i}(\tilde{\alpha}_j)$.

The proof is provided in the Appendix D.

Following the statement of Theorem 5, it is straightforward to observe that if for all neighboring agents $j \in \mathcal{N}_i$, $\tilde{\alpha}_j \to 1$ then $\bar{\alpha}_i \to \alpha_i$. This means that if *the level of reliability* of the neighboring information is one, i.e., $\mathbb{P}[x_j \in \tilde{\mathcal{B}}_j : \forall j \in \mathcal{N}_i] = 1$, then, the state feasibility of agent i will have the same probabilistic level of robustness as the hard communication scheme, i.e., $\mathbb{P}[x_i \in \mathbb{X}_i] \geq \alpha_i = 1 - \varepsilon_i$. Combining this result with the statement of Theorem 2, the proposed soft communication scheme introduces some level of stochasticity on the feasibility of the large-scale system as in (5c). In particular, $\varepsilon_i \in (0,1)$ the level of constraint violation in each agent $i \in \mathcal{N}$ will increase, since it is proportional with the inverse of $\prod_{j \in \mathcal{N}_i}(\tilde{\alpha}_j) \in (0,1)$, and therefore, $\varepsilon = \sum_{i \in \mathcal{N}} \varepsilon_i \in (0,1)$ will also increase. After receiving the parametrization of $\tilde{\mathcal{B}}_j$ and the level of reliability $\tilde{\alpha}_j$, agent $i \in \mathcal{N}$ should immunize itself against all possible variation of $x_j \in \tilde{\mathcal{B}}_j$ by taking the worst-case of $\tilde{\mathcal{B}}_j$, similar to the worst-case reformulation proposed in (Proposition 1, [16]). It is important to notice that in this way, we decoupled the sample generation of agent $j \in \mathcal{N}_i$ from agent $i \in \mathcal{N}$.

6. Plug-and-Play Operational Framework

Using the soft communication scheme, each agent $i \in \mathcal{N}$ should first receive the parametrization of $\tilde{\mathcal{B}}_j$ from all its neighboring agents $j \in \mathcal{N}_i$ together with the level of reliability $\tilde{\alpha}_j$. Then, all agents should immunize themselves against all possible variation of $x_j \in \tilde{\mathcal{B}}_j$. To this end, we formulate the following *robust-communication* scenario program for each agent $i \in \mathcal{N}$:

$$\min_{v_i} \quad J_i(x_i, u_i) \tag{21a}$$

$$\text{s.t.} \quad x_{i,k+1}^s = A_{ii}(\delta_{i,k}^s)x_{i,k}^s + B_i(\delta_{i,k}^s)u_{i,k}^s + q_{i,k}^s, \tag{21b}$$

$$x_{i,k+\ell}^s \in \mathcal{X}_i, \ell \in \mathcal{T}_+, \forall w_i^s \in \mathcal{S}_{w_i}, \forall \delta_i^s \in \mathcal{S}_{\delta_i}, \tag{21c}$$

$$u_{i,k}^s = K_i x_{i,k}^s + v_{i,k} \in \mathcal{U}_i, \forall k \in \mathcal{T}, \tag{21d}$$

$$\begin{cases} q_{i,k}^s = \sum_{j \in \mathcal{N}_i} A_{ij}(\delta_{i,k}^s)x_{j,k}^s + C_i(\delta_{i,k}^s)w_{i,k}^s \\ \forall x_j^s \in \tilde{\mathcal{B}}_j, \forall j \in \mathcal{N}_i \end{cases}. \tag{21e}$$

It is important to highlight that we refer to the aforementioned formulation as the *robust-communication* scenario program, since the communicated variable between neighboring agents should be taken into consideration by the worst-case of $\tilde{\mathcal{B}}_j$. In this way, there is no need for the many iterations of Algorithm 2, and instead, a robustification against all possible variation of $x_j \in \tilde{\mathcal{B}}_j$ is used. It is also important to mention that another feature of using the soft communication scheme is the relaxation of the condition on the required number of scenarios, e.g., each agent $i \in \mathcal{N}$ requests S_i number of scenarios from all its neighboring agents $j \in \mathcal{N}_i$, which may give rise to privacy concerns for the

neighboring agents. The proposed problem (21), specifically the robust constraint (21e), can be solved using a robust (worst-case) reformulation similar to (Proposition 1, [16]) and [7]. It is important to highlight that based on Assumption 2, the proposed optimization problem (21) should be feasible and its feasibility domain has to be a nonempty domain. In case of infeasible solution, each agent $i \in \mathcal{N}$ needs to generate new set of local scenarios, $\mathcal{S}_{w_i}, \mathcal{S}_{\delta_i}$, and also needs to receive a new set $x_j \in \tilde{\mathcal{B}}_j$ from all its neighbors $j \in \mathcal{N}_i$.

We summarize the proposed distributed scenario optimization using soft communication scheme in Algorithm 3. Similar to Algorithm 2, in Algorithm 3 it is also assumed that the feedback control gain matrices K_i for all agents $i \in \mathcal{N}$ are given (4), and the coupling terms $A_{ij}(\delta_{i,k})$ are known between each agent $i \in \mathcal{N}$ and its neighboring agents $j \in \mathcal{N}_i$. It is important to note that Step 5 of Algorithm 3, initializes $\tilde{\mathcal{B}}_j$ of all neighboring agents $j \in \mathcal{N}_i$, which is only used for the initial iteration in Step 8, and then, at each iteration all agents $i \in \mathcal{N}$ will send and receive $\tilde{\mathcal{B}}_j$ from all its neighboring agents $j \in \mathcal{N}_i$ as in Steps 11 and 12, respectively.

We also summarize the steps that are needed for plug-in and plug-out operations of each agent $i \in \mathcal{N}$ in Algorithm 4. Please note that in a plugged-in or plugged-out operation all agents $i \in \mathcal{N}$ have to update their ε_i with β_i to respect the condition in Theorem 2 as in (4) to achieve the desired level of constraint feasibility for the large-scale system (1). One can also redesign K_i to potentially improve the local control performance of each agent $i \in \mathcal{N}$.

Algorithm 3 DSO using Soft Communication Scheme

1: **Decompose** the large-scale dynamical system (1) into N agents as the proposed form in (8)

2: **Determine** the index set of neighboring agents $\mathcal{N}_i$ for each agent $i \in \mathcal{N}$

3: **for all** $i \in \mathcal{N}$ **do**

4:     **fix** initial state $x_{i,0} \in \mathcal{X}_i$, $\varepsilon_i \in (0,1)$, and $\beta_i \in (0,1)$ such that $\varepsilon = \sum_{i \in \mathcal{N}} \varepsilon_i \in (0,1)$, $\beta = \sum_{i \in \mathcal{N}} \beta_i \in (0,1)$

5:     **initialize** $\tilde{\mathcal{B}}_j$ for all neighboring agents $j \in \mathcal{N}_i$

6:     **determine** $\bar{S}_i \in (0,+\infty)$ to approximate the objective function, and S_i following Theorem 2 to approximate

                the chance constraint (10) in an equivalent sense

7:     **generate** $\bar{S}_i$, S_i scenarios of w_i, δ_i to determine the sets of $\bar{\mathcal{S}}_{w_i,\delta_i}$ and $\mathcal{S}_{w_i}$, $\mathcal{S}_{\delta_i}$

8:     **solve** the optimization problem in (21) by taking into account the worst-case of $\tilde{\mathcal{B}}_j$ and determine v_i^*

9:     **generate** $\tilde{S}_i$ scenarios of x_i using the dynamics of agent i in form of (8) and v_i^* together with $\mathcal{S}_{w_i}$, $\mathcal{S}_{\delta_i}$

10:     **determine** set $\tilde{\mathcal{B}}_i$ by solving the optimization problem (16)

11:     **send** the set $\tilde{\mathcal{B}}_i$ to all neighboring agents $j \in \mathcal{N}_i$

12:     **receive** the sets $\tilde{\mathcal{B}}_j$ from all neighboring agents $j \in \mathcal{N}_i$

13:     **apply** the first input of solution $u_{i,k}^* = K_i x_{i,k} + v_{i,k}^*$ into the uncertain subsystem (8)

14:     **measure** the state and substitute it as the initial state of the next step $x_{i,0}$

15:     **set** $k \leftarrow k+1$

16:     **go to** Step (7)

Algorithm 4 Plug-and-Play Operation

    Plug-in Operation

1: **Add** the number of new subsystems into the previous number of subsystems,

      e.g., one additional agent $N \leftarrow N + 1$ such that $|\mathcal{N}| = N + 1$

2: **Update** the index set of neighboring agents $\mathcal{N}_i$ for each agent $i \in \mathcal{N}$

3: **Go to** Step (3) of Algorithm 3

    Plug-out Operation

4: **Remove** the number of departing subsystems from the previous number of subsystems,

      e.g., one agent leaving leads to $N \leftarrow N - 1$ such that $|\mathcal{N}| = N - 1$

5: **Update** the index set of neighboring agents $\mathcal{N}_i$ for each agent $i \in \mathcal{N}$

6: **Go to** Step (3) of Algorithm 3

Comparison: DSO without vs. with Soft Communication

    We now provide a comparison between the proposed distributed scenario optimization without vs. with soft communication in Algorithm 2 and Algorithm 3, respectively, in terms of their computational complexities and the conservatism level of obtained solutions as follows:

    (1) Computational complexity: It is easy to realize that the proposed distributed scenario optimization (Algorithm 3) using soft communication scheme is computationally advantageous compared to the distributed scenario optimization without soft communication (Algorithm 2). This can be clearly seen by just comparing Step 7 and Step 8 in Algorithms 2 and 3, respectively. In Step 7 of Algorithm 2 each agent $i \in \mathcal{N}$ needs to execute the proposed distributed scenario exchange scheme in Algorithm 1 until all its neighboring agents agree on consistency of the set of their state variable scenarios $\mathcal{S}_{x_j}$. However, in Step 8 of Algorithm 3 each agent $i \in \mathcal{N}$ needs to solve only once the robust-communication scenario program in (21) and also solve the proposed optimization problem (16) in Step 10 of Algorithm 3 in order to determine the set $\tilde{\mathcal{B}}_i$.

    (2) Conservatism level: As shown in Theorem 2, the proposed distributed scenario optimization using Algorithm 2 retrieves exactly the same property of the original problem (5) under certain conditions, i.e., $\varepsilon = \sum_{i \in \mathcal{N}} \varepsilon_i \in (0, 1)$, $\beta = \sum_{i \in \mathcal{N}} \beta_i \in (0, 1)$. This is however different in the distributed scenario optimization using the proposed soft communication scheme as presented in Theorem 5. In fact, such a scheme introduces some level of stochasticity on the probabilistic feasibility of the agent $i \in \mathcal{N}$, due to the probabilistic reliability of the neighboring information. It is important to mention that guaranteeing the optimality of the obtained solutions in terms of performance objective(s) using the proposed distributed scenario optimization without or with soft communication in Algorithm 2 and Algorithm 3, respectively, is not included in the scope of this paper and it is subject of our ongoing research work.

7. Numerical Study

    This section presents two different case studies to illustrate the functionality of our proposed framework to deal with dynamically coupled (Section 7.1) and coupling constraints (Section 7.2) neighboring agents with private and common uncertainty sources in networked control problems. The simulation environment for both cases was MATLAB with YALMIP as the interface [31] and Gurobi as the solver.

    We simulate four different problem formulations, namely a centralized SMPC (CSMPC) using (6), a distributed SMPC (DSMPC) via Algorithm 2, and DSMPC with the proposed

soft communication scheme with $0.85-$reliability (DSMPCS-0.85) as described in Definition 1 and DSMPCS-0.50, both following Algorithm 3 in a closed-loop control system framework. For comparison purposes, we also present the results obtained via decoupled SMPC (DeSMPC), where the impact of coupling neighboring dynamics in (8) are relaxed.

In Figures 1 and 2, we evaluate our proposed framework in terms of a-posteriori feasibility validation of the obtained results in both case studies. The "red" line represents the results obtained via DeSMPC, the "blue" line shows the results obtained via CSMPC, the "magenta" presents the results obtained by using DSMPC, the "dark green" and "light green" lines show the results obtained via DSMPCS-0.85 and DSMPCS-0.50, respectively. The "black" lines indicate the bounds of the three dynamically coupled systems.

7.1. Three-Room Case Study

We simulate a building climate comfort system with three rooms such that the temperature of rooms is dynamically coupled without any common constraints. The outside weather temperature and the related losses, e.g., through windows, is considered to be the private uncertainty.

Consider now a three-room building system dynamic:

$$x_{k+1} = A(\delta_k)x_k + B(\delta_k)u_k + C(\delta_k)w_k,$$

where $B = \text{diag}([0.01, 0.01, 0.01])$, $C = \text{diag}([0.02, 0.02, 0.02])$, and A as follows:

$$A = \begin{bmatrix} 0.2 & 0.3 & 0 \\ 0.2 & 0.1 & 0.1 \\ 0.2 & 0 & 0.4 \end{bmatrix}$$

such that $A(\delta_k) = A + \delta_k$ and $B(\delta_k) = B + \delta_k$ as well as $C(\delta_k) = C + \delta_k$. The system matrices are a simplified model of a three-room building such that the states $x_{i,k}$ for $i = 1, 2, 3$, denote the temperature of rooms. The uncertain variable $\delta_k \in \mathbb{R}$ represents the modeling errors, losses through windows, and $w_k \in \mathbb{R}$ can be realized as the outside weather temperatures.

To generate random scenarios from the additive disturbance, we built a discrete normal process such that one day hourly based forecasted (nominal) outside weather temperature is used which varies within 10% of its nominal scenario at each sampling time. As for the uncertainty δ_k, we generate a random variable from a normal distribution with a mean value 0, variance 1 and a maximal magnitude of 0.01 at each sampling time.

The initial state variables are $[21\ 19\ 23]^\top$ and the objective is to keep the temperature of rooms within our desired lower $[20.5\ 18.5\ 22.5]^\top$ and $[21.5\ 19.5\ 23.5]^\top$ upper bounds at the minimum control input u_k. The control input u_k is also constrained to be within -1.5 [kWh] and 1.5 [kWh] for all three rooms, due to actuator saturation. The initialization of the $\tilde{\mathcal{B}}_j$ for all neighboring agents $j \in \mathcal{N}_i$ as in Step 5 in Algorithm 3 can be done for instance by assuming the initial temperature of the neighboring rooms are known for all rooms.

In Figure 1, the dynamically coupled state trajectories for all three rooms are shown. One can clearly see in Figure 1 that the dynamically coupled state trajectories are feasible in a probabilistic sense, since the agent operations are within the lower and upper bounds compared to DeSMPC that violates the constraints completely; the obtained solution via DeSMPC is completely outside of the feasible areas after the first sampling time and thus, we just keep the other results for our discussions. This is a direct result of Theorem 2 such that the obtained solutions via our proposed formulations have to be probabilistically feasible, that can be clearly seen in Figure 1, since the trajectories are on the lower bounds. Italics, space, bold, superscript, subscript.
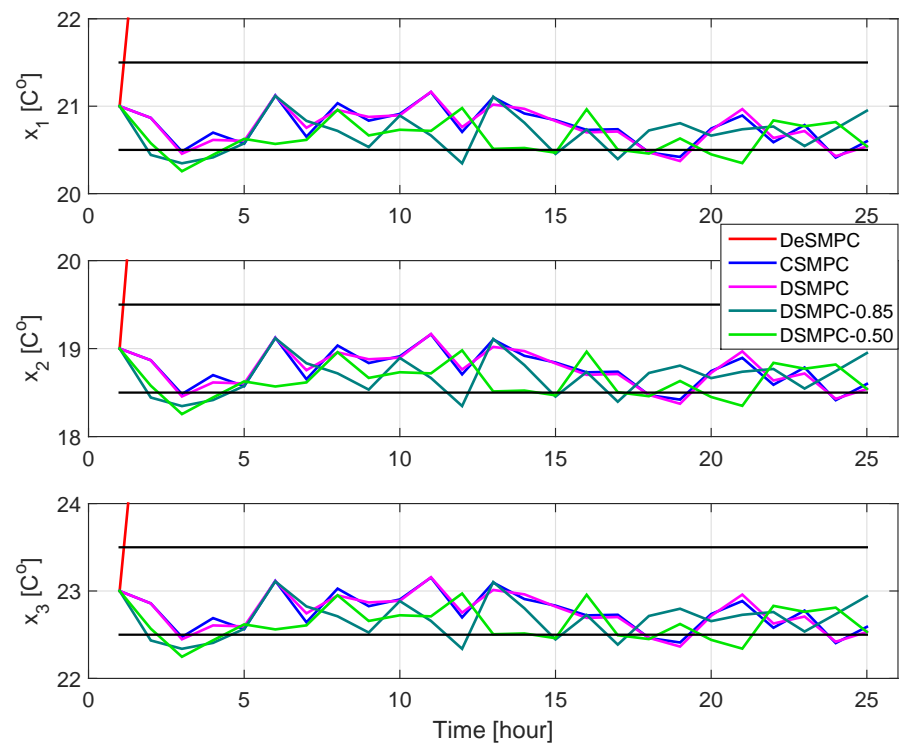
Figure 1. A-posteriori feasibility validation of the obtained results. The "red", "blue", "magenta", "dark green", and "light green" lines are related to the obtained results via DeSMPC, CSMPC, DSMPC, DSMPCS-0.85, and DSMPCS-0.50, respectively. The "black" lines are related to the upper bound values. The top, middle, and down plots are related to the temperatures of room $1, 2$, and 3, respectively.

7.2. Three-Building (ATES Systems) Case Study

We next simulate the thermal energy demands of three buildings modeled using realistic parameters and the actual registered weather data in the city center of Utrecht, The Netherlands, where these buildings are located and these had been equipped with aquifer thermal energy storage (ATES) systems. A simulation study is carried out for one year based on actual weather conditions from March 2011–March 2012 in a weekly based time steps with three months prediction horizon to meet the coupling constraints between ATES systems. We refer interested readers to [26] for the detailed explanations on this case study.

To generate random scenarios from the additive disturbance, we built a discrete normal process such that the nominal scenario is 10% of the amplitude of the energy content in a deterministic ATES system model and varies within 5% of its nominal scenario at each sampling time. As for the uncertainty δ_k, we generate a random variable from a Gaussian distribution with a mean value 0, variance 0.3 and a maximal magnitude of 0.03 at each sampling time.

In Figure 2 we show a-posteriori feasibility validation of the coupling constraints between each agent $i = 1, 2, 3$, and neighboring agents, such that (a) shows the results from mid-May to mid-August 2011, (b) presents the results of December 2011 to February 2012, and (c) depicts the results of mid-April to mid-July 2011. As a first desired achievement, one can clearly see in Figure 2 that the constraints are feasible in a probabilistic sense, since the agent operations are quite close to the upper bounds in the critical time periods compared to DeSMPC that violates the constraints. Strictly speaking, using our proposed framework one can achieve the maximum usage of the aquifer (subsurface) to store thermal energy without affecting the neighboring thermal storage. This is a direct result of Theorem 2 such that the obtained solutions via our proposed formulations have to be probabilistically feasible.
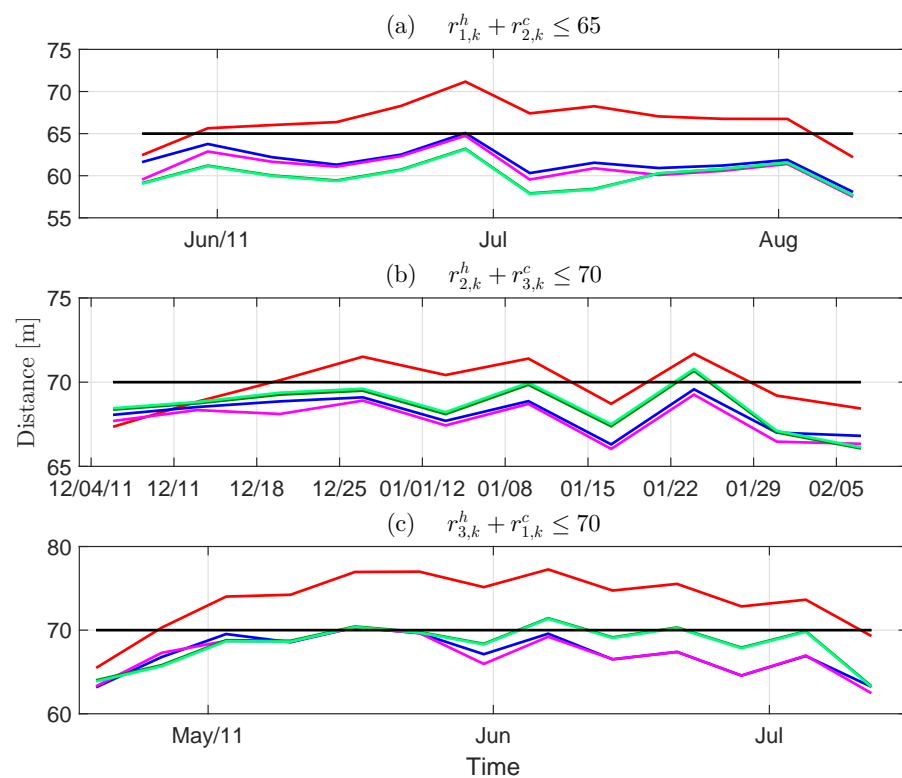
Figure 2. A-posteriori feasibility validation of the obtained results. The "red", "blue", "magenta", "solid green", and "dashed green" lines are related to the obtained results via DeSMPC, CSMPC, DSMPC, DSMPCS−0.85, and DSMPCS−0.50, respectively. The "black" lines are related to the upper bound values.

It is worth mentioning that both Figures 1 and 2 illustrate our other two main contributions more precisely: (1) the obtained results via CSMPC (blue line) and DSMPC (magenta line) are practically equivalent throughout the simulation studies; this is due to Proposition 1 and Theorem 2. Actually, the solutions via DSMPC are slightly more conservative compared to the results via CSMPC, and this is a direct consequence of Theorem 2. In fact, the level of violation in CSMPC is considered to be $\varepsilon = 0.05$ and leading to $\varepsilon_i = 0.0167$ for all agents, which is more restrictive. (2) the proposed soft communication scheme yields fewer conservative solutions as explicitly derived in Theorem 5, and can be clearly seen in Figures 1 and 2 with the obtained results via DSMPCS−0.85 (dark green) and DSMPCS−0.50 (light green). Following Theorem 5 the new violation level using DSMPCS−0.85 is $\bar{\varepsilon}_i = 0.0231$, and using DSMPCS−0.50 is $\bar{\varepsilon}_i = 0.0668$. It is important to notice that the violation level of global chance constraint will increase to $\bar{\varepsilon} = 0.0702$ and $\bar{\varepsilon} = 0.2004$ using DSMPCS−0.85 and DSMPCS−0.50, respectively.

8. Concluding Remarks and Future Work

In this paper, we presented a rigorous approach to distributed stochastic optimization using the scenario-based approximation for large-scale linear systems with local and common uncertainty sources. To highlight the main outcomes of this paper, the following significant findings are listed:

- Extension of the existing results to quantify the robustness of the resulting solutions for both cases of local and common uncertainties in a distributed stochastic optimization framework using the so-called distributed scenario exchange scheme based on ADMM.
- A novel inter-agent soft communication scheme is developed to minimize the amount of information exchange between each subsystem.

- Using a set-based parametrization technique, a novel reliability notion is introduced and quantified the level of feasibility of the obtained solutions via the PnP distributed scenario optimization integrated with the so-called soft communication scheme in a probabilistic sense.
- The theoretical guarantees of the proposed distributed framework coincide with its centralized counterpart.

Potential future research directions are as follows:

- Following the interpretation of Theorem 2, when the number of agents are increasing the violation level of individual agents needs to decrease, which may lead to conservative results for each agent, since the number of samples needs to increase. Addressing such a scalability limitation is subject of our current research work.
- Enhancing the proposed distributed framework to formally address the recursive feasibility and stability of the closed-loop control problems.
- Analyzing the performance objective(s) by means of optimality of the scenario program solutions in a distributed setting.
- Investigating the possibility of extending our proposed frameworks to the fault detection and isolation problems following the results in [32].

Author Contributions: Conceptualization, V.R.; Data curation, V.R.; Formal analysis, V.R.; Funding acquisition, T.K.; Investigation, V.R.; Methodology, V.R.; Project administration, T.K.; Resources, T.K.; Software, V.R.; Supervision, T.K.; Validation, V.R.; Visualization, V.R.; Writing—original draft, V.R.; Writing—review & editing, V.R. & T.K. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proof of Proposition 1

Given Assumption 3 and following the proposed decomposition technique in Section 3, any optimizer of each subsystem v_i^* yields a feasible pair of the state and control input variables of its subsystem $\{x_i^*, u_i^*\} \in \mathbb{X}_i \times \mathbb{U}_i$ such that $\mathbb{X}_i = \mathcal{X}_i^T$, and $\mathbb{X}_i = \mathcal{U}_i^T$. Therefore, the collection of the optimizers $v^* = \mathrm{col}_{i \in \mathcal{N}}(v_i^*)$ will yield the collection of feasible pairs of the state and control input variables of their subsystem:

$$\{x^* = \mathrm{col}_{i \in \mathcal{N}}(x_i^*), u^* = \mathrm{col}_{i \in \mathcal{N}}(u_i^*)\} \in \mathbb{X} \times \mathbb{U},$$

where $\mathbb{X} := \mathcal{X}^T = \prod_{i \in \mathcal{N}} \mathbb{X}_i$ and $\mathbb{U} := \mathcal{U}^T = \prod_{i \in \mathcal{N}} \mathbb{U}_i$, which eventually yields a feasible point for the optimization problem in (6). It is straightforward to use the above relation and to show that any optimizer of the optimization problem in (6) v^* also yields a feasible solution for the proposed optimization problem in (9). We then have to show that both optimization problems will have the same performance index in terms of their objective function values. Due to the proposed decomposition technique and under Assumption 3, it is easy to see that the objective function in (6) can be formulated as additive components such that each component represents the objective function of each subsystem $i \in \mathcal{N}$, and thus: $J(x^*, u^*) = \sum_{i \in \mathcal{N}} J_i(x_i^*, u_i^*)$. The proof is completed.

Appendix B. Proof of Theorem 2

Define $\xi_{i,k} := (w_{i,k}, \delta_{i,k}, \{x_{j,k}\}_{j \in \mathcal{N}_i})$ to be a concatenated uncertain variable for each agent $i \in \mathcal{N}$ such that $\xi_i := \{\xi_{i,k}\}_{k \in \mathcal{T}}$ is defined on probability space $(\Xi_i, \mathfrak{B}(\Xi_i), \mathbb{P}_{\xi_i})$, where $\mathbb{P}_{\xi_i}$ is a probability measure defined over $\Xi_i := \mathcal{W}_i \times \Delta_i \times \prod_{j \in \mathcal{N}_i} \mathbb{X}_j$ and $\mathfrak{B}(\cdot)$ denotes a

Borel σ-algebra. Following this definition, it is straightforward to consider $\xi = \text{col}_{i \in \mathcal{N}}(\xi_i)$ and $\Xi = \prod_{i \in \mathcal{N}} \Xi_i$. Consider also the sample set $\mathcal{S}_{\xi_i} = \mathcal{S}_{w_i} \times \mathcal{S}_{\delta_i} \times \prod_{j \in \mathcal{N}_i} \mathcal{S}_{x_j}$ for each agent $i \in \mathcal{N}$ such that $\mathcal{S}_{\xi} = \prod_{i \in \mathcal{N}} \mathcal{S}_{\xi_i}$.

Consider now v^* to be the optimizer of the centralized scenario MPC problem (6) and define $\text{Vio}(v^*)$ to be the violation probability of the chance constraint (5c) as follows:

$$\text{Vio}(v^*) := \mathbb{P}_{\xi}[\, \xi \in \Xi \, : \, g(v^*, \xi) \notin \mathbb{X} \,], \tag{A1a}$$

where $g(v^*, \xi)$ represents the predicted state trajectory of large-scale system dynamics (1) in a more compact form. In particular, the violation probability can be precisely written as $\text{Vio}(v^*) := \mathbb{P}[\, w \in \mathcal{W}, \delta \in \Delta \, : \, x_{k+\ell} = A_{cl}(\delta_k) x_k + B(\delta_k) v_k^* + C(\delta_k) w_k \notin \mathcal{X} \, , \, \ell \in \mathcal{T}_+ \,|\, x_k = x_0 \,]$, where $A_{cl}(\delta_k) = A(\delta_k) + B(\delta_k) K$. Following Theorem 1, we have

$$\mathbb{P}_{\xi}^S \Big[\mathcal{S}_{\xi} \in \Xi^S \, : \, \text{Vio}(v^*) \leq \varepsilon \Big] \geq 1 - \beta. \tag{A1b}$$

Given Theorem 1, the problem (9) is an exact decomposition of the problem (6). This yields the following equivalence relations:

$$v^* := \text{col}_{i \in \mathcal{N}}(v_i^*) \, , \, \mathbb{X} := \prod_{i \in \mathcal{N}} \mathbb{X}_i, \tag{A2}$$

where v^* is the optimizer of the problem (6) with v_i^* as the optimizer of each agent $i \in \mathcal{N}$ obtained via the problem (9). To this end, it is necessary to prove that the above statements (A1) still hold under the aforementioned relations (A2). We now break down the proof into two steps. We first show the results for each agent $i \in \mathcal{N}$, and then extend into the large-scale scenario optimization problem (6).

(1) Define $\text{Vio}(v_i^*)$ to be the violation probability of each agent $i \in \mathcal{N}$ for the chance constraint (10) as follows:

$$\text{Vio}(v_i^*) := \mathbb{P}_{\xi_i}[\, \xi_i \in \Xi_i \, : \, g_i(v_i^*, \xi_i) \notin \mathbb{X}_i \,], \tag{A3}$$

where $g_i(v_i^*, \xi_i)$ corresponds to the predicted state trajectory of subsystem dynamics (8) for each agent $i \in \mathcal{N}$. Applying the existing results in Theorem 1 for each agent $i \in \mathcal{N}$, we have

$$\mathbb{P}_{\xi_i}^{S_i} \Big[\mathcal{S}_{\xi_i} \in \Xi_i^{S_i} \, : \, \text{Vio}(v_i^*) \leq \varepsilon_i \Big] \geq 1 - \beta_i. \tag{A4}$$

(2) Following the relations (A2), it is easy to rewrite $\text{Vio}(v^*)$ in the following form:

$$\text{Vio}(v^*) = \mathbb{P}_{\xi} \Bigg[\xi \in \Xi \, : \, g(v^*, \xi) \notin \prod_{i \in \mathcal{N}} \mathbb{X}_i \Bigg].$$

It is then sufficient to show that for $S = \max_{i \in \mathcal{M}} S_i$:

$$\mathbb{P}_{\xi}^S \Big[\mathcal{S}_{\xi} \in \Xi^S \, : \, \text{Vio}(v^*) \geq \varepsilon \Big] \leq \beta. \tag{A5}$$

where $\varepsilon = \sum_{i \in \mathcal{N}} \varepsilon_i \in (0, 1)$ and $\beta = \sum_{i \in \mathcal{N}} \beta_i \in (0, 1)$. Hence

$$\text{Vio}(v^*) = \mathbb{P}_{\xi} \Bigg[\xi \in \Xi \, : \, g(v^*, \xi) \notin \prod_{i \in \mathcal{N}} \mathbb{X}_i \Bigg]$$

$$= \mathbb{P}_{\xi}[\, \xi \in \Xi \, : \, \exists\, i \in \mathcal{N} \, , \, g(v_i^*, \xi_i) \notin \mathbb{X}_i \,]$$

$$= \mathbb{P}_{\xi} \Bigg[\bigcup_{i \in \mathcal{N}} \{\, \xi_i \in \Xi_i \, : \, g(v_i^*, \xi_i) \notin \mathbb{X}_i \,\} \Bigg]$$

$$\leq \sum_{i \in \mathcal{N}} \mathbb{P}_{\xi_i}[\, \xi_i \in \Xi_i : g(v_i^*, \xi_i) \notin \mathbb{X}_i \,] = \sum_{i \in \mathcal{N}} \text{Vio}(v_i^*).$$

The last statement implies that $\mathrm{Vio}(v^*) \leq \sum_{i \in \mathcal{N}} \mathrm{Vio}(v_i^*)$, and thus, we have

$$
\begin{aligned}
\mathbb{P}_{\tilde{\zeta}}^S & \left[\mathcal{S}_{\tilde{\zeta}} \in \Xi^S \ : \ \mathrm{Vio}(v^*) \geq \varepsilon \right] \\
& \leq \mathbb{P}_{\tilde{\zeta}}^S \left[\mathcal{S}_{\tilde{\zeta}} \in \Xi^S \ : \ \sum_{i \in \mathcal{N}} \mathrm{Vio}(v_i^*) \geq \sum_{i \in \mathcal{N}} \varepsilon_i \right] \\
& = \mathbb{P}_{\tilde{\zeta}}^S \left[\bigcup_{i \in \mathcal{N}} \left\{ \mathcal{S}_{\tilde{\zeta}_i} \in \Xi_i^{S_i} \ : \ \mathrm{Vio}(v_i^*) \geq \varepsilon_i \right\} \right] \\
& \leq \sum_{i \in \mathcal{N}} \mathbb{P}_{\tilde{\zeta}_i}^{S_i} \left[\mathcal{S}_{\tilde{\zeta}_i} \in \Xi_i^{S_i} \ : \ \mathrm{Vio}(v_i^*) \geq \varepsilon_i \right] \leq \sum_{i \in \mathcal{N}} \beta_i = \beta \ .
\end{aligned}
$$

The obtained bounds in the above procedure are the desired assertions as stated in the theorem. The proof is completed by noting that the feasible set $\mathbb{X} = \prod_{i \in \mathcal{N}} \mathbb{X}_i$ of the problem (6) and (9) has a nonempty interior, and it thus admits at least one feasible solution $v^* = \mathrm{col}_{i \in \mathcal{N}}(v_i^*)$.

Appendix C. Proof of Theorem 4

Equation (19) is a direct result of the scenario approach theory in [33], if $\tilde{\beta}_j$ is chosen such that

$$
\binom{\tilde{S}_j}{n_j} \tilde{\alpha}_j^{\tilde{S}_j - n_j} \leq \tilde{\beta}_j \ . \tag{A6}
$$

Considering the worst-case equality in the above relation and some algebraic manipulations, one can obtain the above assertion. The proof is completed.

Appendix D. Proof of Theorem 5

The proof consists of two main steps. We first provide an analytical expression for the robustness of the solution in agent i by taking into account the effect of just one neighboring agent $j \in \mathcal{N}_j$, and then extend the obtained results for the case when the agent i interacts with more neighboring agents, e.g., for all $j \in \mathcal{N}_j$.

Following Remark 1, we have the following updated situation:

$$
\alpha_i \leq \mathbb{P} \left[x_i \in \mathbb{X}_i \ , \ x_j \in \tilde{\mathcal{B}}_j \right] ,
$$

which is a joint probability of $x_i \in \mathbb{X}_i$ and $x_j \in \tilde{\mathcal{B}}_j$. Such a joint probability can be equivalently written as a joint cumulative distribution function (CDF):

$$
\begin{aligned}
\alpha_i & \leq \mathbb{P} \left[x_i \in \mathbb{X}_i \ , \ x_j \in \tilde{\mathcal{B}}_j \right] \\
& = \int_{\mathbb{X}_i} \int_{\tilde{\mathcal{B}}_j} p(x_i \, , \, x_j) \, \mathrm{d}x_i \, \mathrm{d}x_j = F_{x_i, x_j} \left(\mathbb{X}_i \, , \, \tilde{\mathcal{B}}_j \right) ,
\end{aligned} \tag{A7}
$$

where $p(x_i \, , \, x_j)$ is the joint probability density function (PDF) of x_i and x_j. Our goal is to calculate:

$$
\mathbb{P} \left[x_i \in \mathbb{X}_i \right] = \int_{\mathbb{X}_i} p(x_i) \, \mathrm{d}x_i = F_{x_i} \left(\mathbb{X}_i \right) ,
$$

where $p(x_i)$ is the PDF of x_i. To transform the joint CDF into the marginal CDF of x_i, one can take the limit of the joint CDF as $\tilde{\mathcal{B}}_j$ approaches $\mathbb{R}^{n_j}$:

$$
\begin{aligned}
\mathbb{P}\left[\,x_i \in \mathbb{X}_i\,\right] = F_{x_i}\left(\mathbb{X}_i\right) &= \lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_i, x_j}\left(\mathbb{X}_i\,,\,\tilde{\mathcal{B}}_j\right) \\
&= \lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_i \mid x_j}\left(\mathbb{X}_i \mid \tilde{\mathcal{B}}_j\right) F_{x_j}\left(\tilde{\mathcal{B}}_j\right) \\
&= F_{x_i}\left(\mathbb{X}_i\right) \lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_j}\left(\tilde{\mathcal{B}}_j\right),
\end{aligned}
\tag{A8}
$$

where the last equality is due to the fact that x_i and x_j are conditionally independent.

To determine $\lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_j}\left(\tilde{\mathcal{B}}_j\right)$, one can calculate:

$$
\begin{aligned}
\lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_j}\left(\tilde{\mathcal{B}}_j\right) &= \int_{\mathbb{R}^{n_j}} p(x_j)\mathrm{d}x_j \\
&= \int_{\mathbb{R}^{n_j} \setminus \tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j + \int_{\tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j \\
&= \mathbb{P}\left[\,x_j \notin \tilde{\mathcal{B}}_j\,\right] + \mathbb{P}\left[\,x_j \in \tilde{\mathcal{B}}_j\,\right] \\
&= (1 - \tilde{\alpha}_j) + \tilde{\alpha}_j = 1,
\end{aligned}
\tag{A9}
$$

where $p(x_j)$ is the PDF of x_j, and the last equality is a direct result of Theorem 4.

We now put all the steps together as follows:

$$
\begin{aligned}
\alpha_i \leq \mathbb{P}\left[\,x_i \in \mathbb{X}_i\,,\,x_j \in \tilde{\mathcal{B}}_j\,\right] &= F_{x_i, x_j}\left(\mathbb{X}_i\,,\,\tilde{\mathcal{B}}_j\right) \\
&\leq F_{x_i}\left(\mathbb{X}_i\right) \lim_{\tilde{\mathcal{B}}_j \to \mathbb{R}^{n_j}} F_{x_j}\left(\tilde{\mathcal{B}}_j\right) \\
&= \mathbb{P}\left[\,x_i \in \mathbb{X}_i\,\right] \left(\int_{\mathbb{R}^{n_j} \setminus \tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j + \int_{\tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j \right) \\
&\leq \int_{\mathbb{R}^{n_j} \setminus \tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j + \mathbb{P}\left[\,x_i \in \mathbb{X}_i\,\right] \int_{\tilde{\mathcal{B}}_j} p(x_j)\mathrm{d}x_j \\
&= (1 - \tilde{\alpha}_j) + \tilde{\alpha}_j\,\mathbb{P}\left[\,x_i \in \mathbb{X}_i\,\right],
\end{aligned}
$$

where the first inequality and equality is due to (A7), the second inequality is due to (A8), the second and last equality is due to (A9), and the last inequality is considering the worst-case situation, e.g., $\mathbb{P}\left[\,x_i \in \mathbb{X}_i \mid x_j \notin \tilde{\mathcal{B}}_j\,\right] = 1$.

By rearranging the last equation in above result:

$$
\frac{\alpha_i - (1 - \tilde{\alpha}_j)}{\tilde{\alpha}_j} = 1 - \frac{1 - \alpha_i}{\tilde{\alpha}_j} = \bar{\alpha}_i \leq \mathbb{P}\left[\,x_i \in \mathbb{X}_i\,\right].
\tag{A10}
$$

This completes the proof of first part. We now need to show the effect of having more than one neighboring agent. To this end, the most straightforward step, in order to extend the current results, is to use the fact that all neighboring agents are independent from each other. We therefore can apply the previous results for a new situation where the agent i

with the probabilistic level of feasibility $\bar{\alpha}_i$ have another neighboring agent $\nu \in \mathcal{N}_i$ with $\tilde{\alpha}_\nu$ the level of reliability of $\tilde{\mathcal{B}}_\nu$. By using (A10), we have the following relations for $j, \nu \in \mathcal{N}_i$

$$
1 - \frac{1 - \bar{\alpha}_i}{\tilde{\alpha}_\nu} = 1 - \frac{1 - \left(1 - \frac{1 - \alpha_i}{\tilde{\alpha}_j}\right)}{\tilde{\alpha}_\nu}
$$

$$
= 1 - \frac{1 - \alpha_i}{\tilde{\alpha}_j \, \tilde{\alpha}_\nu} \leq \mathbb{P}[\, x_i \in \mathbb{X}_i \,] \, .
$$

By continuing the similar arguments for all neighboring agents, one can obtain $\bar{\alpha}_i = 1 - \frac{1 - \alpha_i}{\tilde{\alpha}_i} \leq \mathbb{P}[\, x_i \in \mathbb{X}_i \,]$ such that $\tilde{\alpha}_i = \tilde{\alpha}_1 \cdots \tilde{\alpha}_j \, \tilde{\alpha}_\nu \cdots \tilde{\alpha}_{|\mathcal{N}_i|} = \prod_{j \in \mathcal{N}_i}(\tilde{\alpha}_j)$. The proof is completed.

References

1. Kouvaritakis, B.; Cannon, M.; Tsachouridis, V. Recent developments in stochastic MPC and sustainable development. *Annu. Rev. Control* **2004**, *28*, 23–35. [CrossRef]
2. Primbs, J.A. A soft constraint approach to stochastic receding horizon control. In Proceedings of the Conference on Decision and Control (CDC), New Orleans, LA, USA, 12–14 December 2007; pp. 4797–4802.
3. Bemporad, A.; Morari, M. Robust model predictive control: A survey. In *Robustness in Identification and Control*; Springer: Berlin/Heidelberg, Germany, 1999; pp. 207–226.
4. Lavaei, J.; Aghdam, A.G. Control of continuous-time LTI systems by means of structurally constrained controllers. *Automatica* **2008**, *44*, 141–148. [CrossRef]
5. Richards, A.; How, J.P. Robust distributed model predictive control. *Int. J. Control* **2007**, *80*, 1517–1531. [CrossRef]
6. Cannon, M.; Kouvaritakis, B.; Rakovic, S.V.; Cheng, Q. Stochastic tubes in model predictive control with probabilistic constraints. *IEEE Trans. Autom. Control* **2011**, *56*, 194–200. [CrossRef]
7. Riverso, S.; Farina, M.; Ferrari-Trecate, G. Plug-and-play decentralized model predictive control for linear systems. *IEEE Trans. Autom. Control* **2013**, *58*, 2608–2614. [CrossRef]
8. Dai, L.; Xia, Y.; Gao, Y.; Cannon, M. Distributed Stochastic MPC of Linear Systems with Additive Uncertainty and Coupled Probabilistic Constraints. *IEEE Trans. Autom. Control* **2016**, *62*, 3474–3481. [CrossRef]
9. Hokayem, P.; Cinquemani, E.; Chatterjee, D.; Ramponi, F.; Lygeros, J. Stochastic receding horizon control with output feedback and bounded controls. *Automatica* **2012**, *48*, 77–88. [CrossRef]
10. Prandini, M.; Garatti, S.; Lygeros, J. A randomized approach to stochastic model predictive control. In Proceedings of the Conference on Decision and Control (CDC), Maui, HI, USA, 10–13 December 2012; pp. 7315–7320.
11. Calafiore, G.C.; Fagiano, L. Stochastic model predictive control of LPV systems via scenario optimization. *Automatica* **2013**, *49*, 1861–1866. [CrossRef]
12. Schildbach, G.; Morari, M. Scenario MPC for linear time-varying systems with individual chance constraints. In Proceedings of the American Control Conference (ACC), Chicago, IL, USA, 1–3 July 2015; pp. 415–421.
13. Lorenzen, M.; Allgöwer, F.; Dabbene, F.; Tempo, R. Scenario-based Stochastic MPC with guaranteed recursive feasibility. In Proceedings of the Conference on Decision and Control (CDC), Osaka, Japan, 15–18 December 2015; pp. 4958–4963.
14. Margellos, K.; Goulart, P.; Lygeros, J. On the road between robust optimization and the scenario approach for chance constrained optimization problems. *IEEE Trans. Autom. Control* **2014**, *59*, 2258–2263. [CrossRef]
15. Zhang, X.; Margellos, K.; Goulart, P.; Lygeros, J. Stochastic model predictive control using a combination of randomized and robust optimization. In Proceedings of the Conference on Decision and Control (CDC), Florence, Italy, 10–13 December 2013; pp. 7740–7745.
16. Rostampour, V.; Keviczky, T. Robust randomized model predictive control for energy balance in smart thermal grids. In Proceedings of the European Control Conference (ECC), Aalborg, Denmark, 29 June–1 July 2016; pp. 1201–1208.
17. Campi, M.C.; Garatti, S. The exact feasibility of randomized solutions of uncertain convex programs. *SIAM J. Optim.* **2008**, *19*, 1211–1230. [CrossRef]
18. Long, Y.; Liu, S.; Xie, L.; Johansson, K.H. A scenario-based distributed stochastic MPC for building temperature regulation. In Proceedings of the Conference on Automation Science and Engineering (CASE), Taipei, Taiwan, 18–22 August 2014; pp. 1091–1096.
19. Liu, S.; Sadowska, A.; Hellendoorn, H.; De Schutter, B. Scenario-based Distributed Model Predictive Control for freeway networks. In Proceedings of the International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 1779–1784.
20. Calafiore, G.C.; Lyons, D. Random convex programs for distributed multi-agent consensus. In Proceedings of the European Control Conference (ECC), Zurich, Switzerland, 17–19 July 2013; pp. 250–255.
21. Papaefthymiou, G.; Klockl, B. MCMC for wind power simulation. *IEEE Trans. Energy Convers.* **2008**, *23*, 234–240. [CrossRef]
22. Schildbach, G.; Fagiano, L.; Frei, C.; Morari, M. The scenario approach for stochastic model predictive control with bounds on closed-loop constraint violations. *Automatica* **2014**, *50*, 3009–3018. [CrossRef]

23. Rostampour, V.; Esfahani, P.M.; Keviczky, T. Stochastic nonlinear model predictive control of an uncertain batch polymerization reactor. In Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC), Seville, Spain, 17–20 September 2015; Volume 48, pp. 540–545.

24. Bertsekas, D.P.; Tsitsiklis, J.N. *Parallel and Distributed Computation: Numerical Methods*; Prentice Hall: Englewood Cliffs, NJ, USA, 1989; Volume 23.

25. Slater, M. *Lagrange Multipliers Revisited*; Technical Report; No. 80; Cowles Foundation for Research in Economics, Yale University: New Haven, CT, USA, 1959.

26. Rostampour, V.; Keviczky, T. Probabilistic Energy Management for Building Climate Comfort in Smart Thermal Grids with Seasonal Storage Systems. *IEEE Trans. Smart Grid* **2018**. [CrossRef]

27. Shi, W.; Ling, Q.; Yuan, K.; Wu, G.; Yin, W. On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. *IEEE Trans. Signal Process.* **2014**, *62*, 1750–1761. [CrossRef]

28. Rostampour, V.; ter Haar, O.; Keviczky, T. Distributed Stochastic Reserve Scheduling in AC Power Systems With Uncertain Generation. *IEEE Trans. Power Syst.* **2017**. [CrossRef]

29. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [CrossRef]

30. He, B.; Yuan, X. On non-ergodic convergence rate of Douglas–Rachford alternating direction method of multipliers. *Numerische Mathematik* **2015**, *130*, 567–577. [CrossRef]

31. Löfberg, J. YALMIP: A toolbox for modeling and optimization in MATLAB. In Proceedings of the International Symposium on Computer Aided Control Systems Design, New Orleans, LA, USA, 2–4 September 2004; pp. 284–289.

32. Rostampour, V.; Ferrari, R.; Keviczky, T. A set based probabilistic approach to threshold design for optimal fault detection. In Proceedings of the American Control Conference (ACC), Seattle, WA, USA, 24–26 May 2017; pp. 5422–5429.

33. Calafiore, G.C.; Campi, M.C. The scenario approach to robust control design. *IEEE Trans. Autom. Control* **2006**, *51*, 742–753. [CrossRef]