Vistas in Control | ETH Zurich | September 10 - 11 2018

Multi-agent distributed optimization over networks and its application to energy systems

Maria Prandini

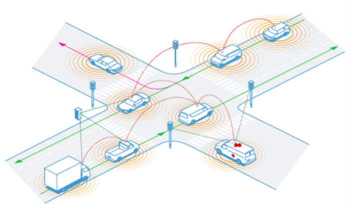
POLITECNICO
MILANO 1863

Introduction

Social networks

Robotic networks

Transportation systems



Energy systems

Introduction

Goal

Optimize the performance of the network

Introduction

Goal

Optimize the performance of the network

Characteristics of the network

- Large scale – System with multiple interacting components
- Multi-agent – Components can perform computations, communicate with each other, and cooperate to reach a common goal
- Heterogeneous – Different physical or technological constraints per agent; different objectives per agent
- Uncertain – Endogenous and/or exogenous uncertainty affects the system globally and/or locally
- Combinatorial – Discrete and continuous decision variables

Introduction

Challenges

- Computation: Problem size too big, even combinatorial!
- Communication: Not all communication links at place; link failures
- Information privacy: Agents may not want to share information with everyone
- Uncertainty: Neglecting uncertainty may lead to an infeasible solution; uncertainty often known through data

Challenges

- Computation: Problem size too big, even combinatorial!
- Communication: Not all communication links at place; link failures
- Information privacy: Agents may not want to share information with everyone
- Uncertainty: Neglecting uncertainty may lead to an infeasible solution; uncertainty often known through data

Distributed data-based optimization

Find an optimal solution by solving in parallel smaller optimization problems local to each agent while accounting for uncertainty known locally to each agent through data

Why go distributed?

1. Scalable methodology
    - Communication: Only between neighbors, limited amount of info exchanged
    - Computation: Only local; in parallel for all agents on a smaller problem

Introduction

Why go distributed?

1. Scalable methodology
   - Communication: Only between neighbors, limited amount of info exchanged
   - Computation: Only local; in parallel for all agents on a smaller problem
2. Resilience to communication failures

Introduction

Why go distributed?

1. Scalable methodology
   - Communication: Only between neighbors, limited amount of info exchanged
   - Computation: Only local; in parallel for all agents on a smaller problem
2. Resilience to communication failures
3. Information privacy
   - Agents do not reveal information about their preferences (encoded by objective and constraint functions) to each other

Outline

1. The deterministic case
    - Problem set-up
    - Distributed proximal algorithm
    - Analysis (assumptions + convergence)
    - Connection with other methods

Outline

1. The deterministic case
    - Problem set-up
    - Distributed proximal algorithm
    - Analysis (assumptions + convergence)
    - Connection with other methods
2. The stochastic case
    - Problem set-up
    - Data-based approach
    - Distributed data-based implementation

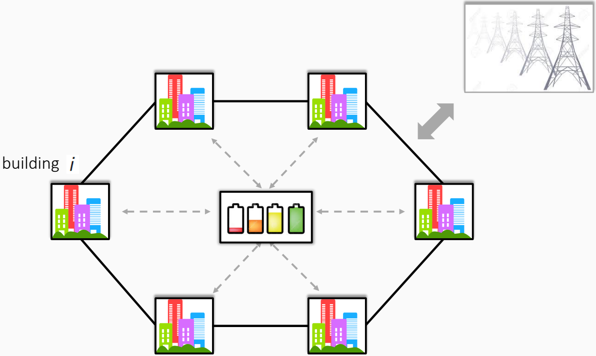
Outline

1. The deterministic case
    - Problem set-up
    - Distributed proximal algorithm
    - Analysis (assumptions + convergence)
    - Connection with other methods
2. The stochastic case
    - Problem set-up
    - Data-based approach
    - Distributed data-based implementation
3. Constraint-coupled problem set-up
    - Distributed dual decomposition algorithm
    - Discrete case

Outline

1. The deterministic case
    - Problem set-up
    - Distributed proximal algorithm
    - Analysis (assumptions + convergence)
    - Connection with other methods
2. The stochastic case
    - Problem set-up
    - Data-based approach
    - Distributed data-based implementation
3. Constraint-coupled problem set-up
    - Distributed dual decomposition algorithm
    - Discrete case
4. Summary & Future work

Building district energy management



Set-up

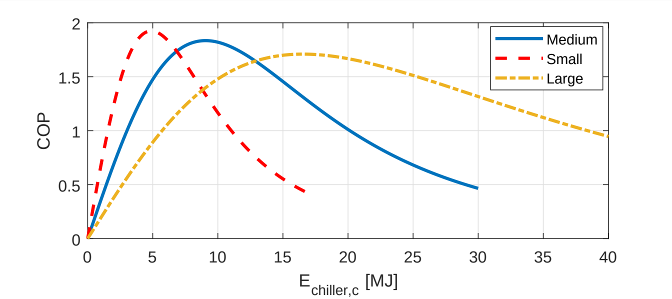
- Each building equipped with a chiller plant
- Shared cooling network that acts as a thermal storage device

Goal

Determine use of storage + zones temperature set-points to minimize the cost of the electrical energy consumption of the chillers in the district

Building district energy management

1. Chiller plant
    - Convert electrical energy into cooling energy
    - Characterized via COP (ratio between cooling energy and electrical energy)



Building district energy management

2. Building energy contribution
    - Walls-zones energy exchange – building thermal dynamics
    - Energy due to people occupancy
    - Zone thermal inertia
    - Other internal energy contribution, e.g. internal lighting, radiation through windows

Building district energy management

2. Building energy contribution
   - Walls-zones energy exchange – building thermal dynamics
   - Energy due to people occupancy
   - Zone thermal inertia
   - Other internal energy contribution, e.g. internal lighting, radiation through windows

3. Thermal storage

$$S(k+1) = \alpha S(k) - \sum_i s_i(k)$$

   - $S(k)$: Energy stored
   - $s_i(k)$: Energy exchange between building i and storage
     > 0: discharging the storage; < 0: charging
   - α: Energy losses coefficient

Building district energy management

Optimization problem

minimize Sum of costs of chillers electrical energy consumption

subject to

1. Chiller thermal energy request = Buildings energy request – Storage energy
2. Storage dynamics
3. Storage limits, chillers limits, comfort constraints

Building district energy management

Optimization problem

minimize Sum of costs of chillers electrical energy consumption

subject to

1. Chiller thermal energy request = Buildings energy request – Storage energy
2. Storage dynamics
3. Storage limits, chillers limits, comfort constraints

Compact form – x: temperature set-points, storage usage

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i X_i$$

Problem set-up

Decision-coupled problem

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i X_i$$

Problem set-up

Decision-coupled problem

$$\begin{aligned} &\text{minimize } \sum_i f_i(x) \\ &\text{subject to} \\ &x \in \bigcap_i X_i \end{aligned}$$

- local objectives f_i

Problem set-up

Decision-coupled problem

$$\begin{aligned} &\text{minimize} \sum_i f_i(x) \\ &\text{subject to} \\ &x \in \bigcap_i X_i \end{aligned}$$

- local objectives f_i
- local constraints X_i

Problem set-up

Decision-coupled problem

$$\text{minimize} \sum_i f_i(x)$$

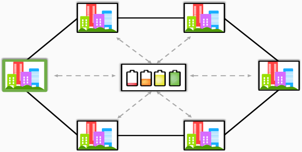
subject to

$$x \in \bigcap_i X_i$$

- local objectives f_i
- local constraints X_i
- coupled decision x

Proposed distributed algorithm

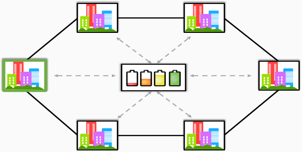
Step 1: Local problem of agent i



$$\left.\begin{aligned} &\text{minimize } f_i(x_i) + g(x_i, z_i) \\ &\text{subject to} \\ &x_i \in X_i \end{aligned}\right\} \Rightarrow x_i^*(z_i)$$

Proposed distributed algorithm

Step 1: Local problem of agent *i*

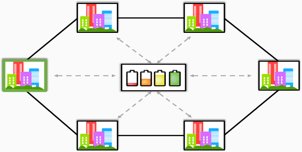


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

- x_i: "copy" of x maintained by agent i

Proposed distributed algorithm

Step 1: Local problem of agent i

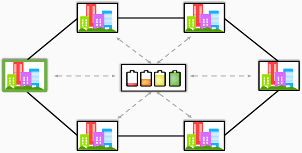


$$\left.\begin{aligned}&\text{minimize } f_i(x_i) + g(x_i, z_i)\\&\text{subject to}\\&x_i \in X_i\end{aligned}\right\} \Rightarrow x_i^*(z_i)$$

- x_i: "copy" of x maintained by agent i
- X_i: local constraint set of agent i

Proposed distributed algorithm

Step 1: Local problem of agent i

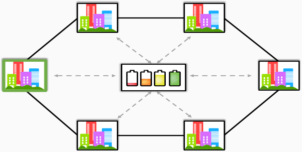


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

- x_i: "copy" of x maintained by agent i
- X_i: local constraint set of agent i
- z_i: information vector – constructed based on the info of agent's i neighbors

Proposed distributed algorithm

Step 1: Local problem of agent *i*

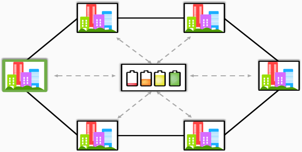


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

- x_i: "copy" of x maintained by agent i
- X_i: local constraint set of agent i
- z_i: information vector – constructed based on the info of agent's i neighbors
- Objective function
  $f_i(x_i)$: local cost/utility of agent i
  $g(x_i, z_i)$: Proxy term, penalizing disagreement with other agents

Proposed distributed algorithm

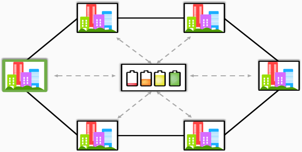
Step 1: Local problem of agent *i*



$$\left.\begin{aligned}&\text{minimize } f_i(x_i) + g(x_i, z_i)\\&\text{subject to}\\&x_i \in X_i\end{aligned}\right\} \Rightarrow x_i^*(z_i)$$

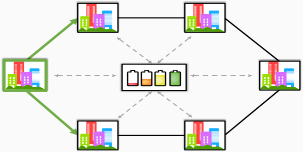
Proposed distributed algorithm

Step 1: Local problem of agent i

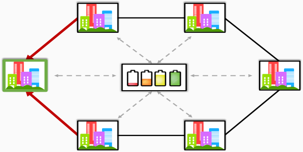


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

Step 2a: Broadcast $x_i^*(z_i)$ to neighbors

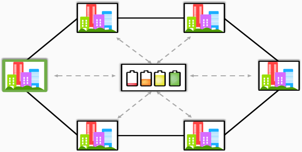


Step 2b: Receive neighbors' solutions



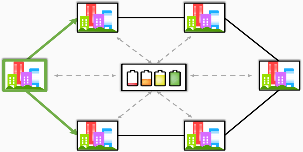
Proposed distributed algorithm

Step 1: Local problem of agent i

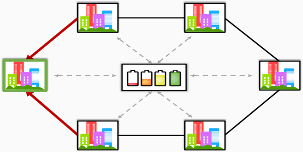


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

Step 2a: Broadcast $x_i^*(z_i)$ to neighbors



Step 2b: Receive neighbors' solutions

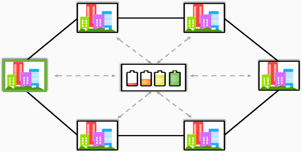


Step 3: Update z_i on the basis of information received

Go to Step 1

Proposed distributed algorithm

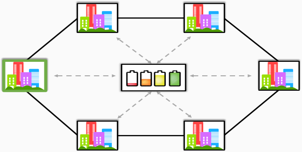
Local problem of agent i



$$\left.\begin{array}{l}\text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i\end{array}\right\} \Rightarrow x_i^*(z_i)$$

Proposed distributed algorithm

Local problem of agent i

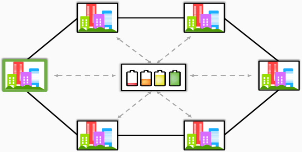


$$\left.\begin{array}{l} \text{minimize } f_i(x_i) + g(x_i, z_i) \\ \text{subject to} \\ x_i \in X_i \end{array}\right\} \Rightarrow x_i^*(z_i)$$

- Specify
  - Information vector z_i
  - Proxy term term $g(x_i, z_i)$
- Note that these terms change across algorithm iterations

Proposed distributed algorithm

Local problem of agent i at iteration $k+1$

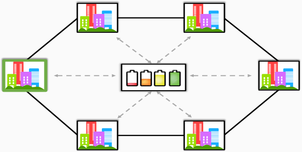


$$z_i(k) = \sum_j a_j^i(k) x_j(k)$$

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

Proposed distributed algorithm

Local problem of agent i at iteration $k+1$



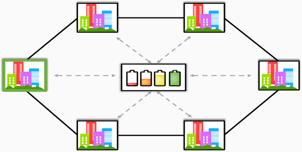
$$z_i(k) = \sum_j a_j^i(k) x_j(k)$$

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

- Information vector
  - $z_i(k) = \sum_j a_j^i(k) x_j(k)$
  - $a_j^i(k)$: how agent i weights info of agent j
- Proxy term
  - $\frac{1}{c(k)} \|x_i - z_i(k)\|^2$: deviation from (weighted) average
  - $c(k)$: trade-off between optimality and agents' disagreement

Proposed distributed algorithm

Local problem of agent i at iteration $k+1$



$$z_i(k) = \sum_j a^i_j(k) x_j(k)$$

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

- Does this algorithm converge?
- If yes, does it provide the same solution with the centralized problem (had we been able to solve it)?

Algorithm analysis

1. Convexity and compactness
    - $f_i(\cdot)$: convex for all i
    - X_i: compact, convex, non-empty interior for all i
      $\Rightarrow f_i(\cdot)$: Lipschitz continuous on X_i

Algorithm analysis

1. Convexity and compactness
    - $f_i(\cdot)$: convex for all i
    - X_i: compact, convex, non-empty interior for all i
      $\Rightarrow f_i(\cdot)$: Lipschitz continuous on X_i
2. Choice of the proxy term
    - $\{c(k)\}_k$: non-increasing
    - Should not decrease too fast

$$\sum_k c(k) = \infty$$
$$\sum_k c(k)^2 < \infty$$

    - E.g., harmonic series

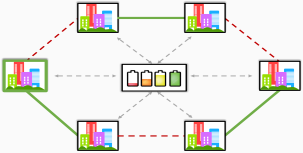
Algorithm analysis

3. Information mix
    - Weights $a^i_j(k)$: non-zero lower bound if link between $i - j$ present
      $\Rightarrow$ Info mixing at a non-diminishing rate
    - Weights $a^i_j(k)$: form a doubly stochastic matrix
      $\Rightarrow$ Agents influence each other equally in the long run

Algorithm analysis

3. Information mix
    - Weights $a^i_j(k)$: non-zero lower bound if link between $i - j$ present
      $\Rightarrow$ Info mixing at a non-diminishing rate
    - Weights $a^i_j(k)$: form a doubly stochastic matrix
      $\Rightarrow$ Agents influence each other equally in the long run

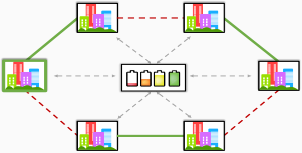
4. Network connectivity – All information flows (eventually)
    - Any pair of agents communicates infinitely often
    - Bounded intercommunication time



Algorithm analysis

3. Information mix
   - Weights $a_j^i(k)$: non-zero lower bound if link between $i - j$ present
     $\Rightarrow$ Info mixing at a non-diminishing rate
   - Weights $a_j^i(k)$: form a doubly stochastic matrix
     $\Rightarrow$ Agents influence each other equally in the long run

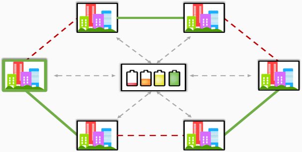
4. Network connectivity – All information flows (eventually)
   - Any pair of agents communicates infinitely often
   - Bounded intercommunication time



Algorithm analysis

3. Information mix
    - Weights $a_j^i(k)$: non-zero lower bound if link between $i - j$ present
      $\Rightarrow$ Info mixing at a non-diminishing rate
    - Weights $a_j^i(k)$: form a doubly stochastic matrix
      $\Rightarrow$ Agents influence each other equally in the long run

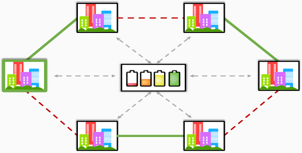
4. Network connectivity – All information flows (eventually)
    - Any pair of agents communicates infinitely often
    - Bounded intercommunication time



Algorithm analysis

3. Information mix
   - Weights $a_j^i(k)$: non-zero lower bound if link between $i - j$ present
     $\Rightarrow$ Info mixing at a non-diminishing rate
   - Weights $a_j^i(k)$: form a doubly stochastic matrix
     $\Rightarrow$ Agents influence each other equally in the long run

4. Network connectivity – All information flows (eventually)
   - Any pair of agents communicates infinitely often
   - Bounded intercommunication time



Algorithm analysis

Main result

Under the structural + network assumptions, the proposed proximal algorithm converges to some minimizer x^* of the centralized problem, i.e.,

$$\lim_{k \to \infty} \|x_i(k) - x^*\| = 0, \text{ for all } i$$

- Asymptotic agreement and optimality
- Rate no faster than $c(k)$ – "slow enough" to trade agreement and optimality

Comparison with other methods

- Proximal algorithms vs. gradient/subgradient methods

Constrained Consensus and Optimization in Multi-Agent Networks

Angelia Nedić, *Member, IEEE*, Asuman Ozdaglar, *Member, IEEE*, and Pablo A. Parrilo, *Senior Member, IEEE*

Abstract
by multipl
over a netw
general in
multiple ag
where the
objective f
where the
convex sets

Distributed Random Projection Algorithm for Convex Optimization

Soomin Lee and Angelia Nedić

***Abstract*—Random proj**
strained optimization whe
advance or the projection
is computationally prohibi
***random projection* algorith**
problems that can be used
time-varying network, wh
function and its own const
of all agents converge to th
surely. Experiments on
demonstrate good perform

Math. Program., Ser. B (2011) 129:163–195
DOI 10.1007/s10107-011-0472-0

FULL LENGTH PAPER

Incremental proximal methods for large scale convex optimization

Dimitri P. Bertsekas

Comparison with other methods

- Proximal algorithms

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

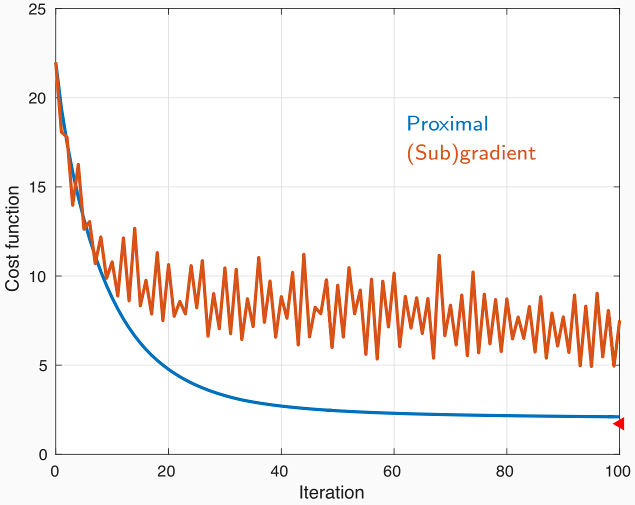
- Gradient algorithms

$$x_i(k+1) = P_{X_i}\left[z_i(k) - c(k)\nabla f_i(z_i(k))\right]$$

- Proximal algorithms allow for
    - No gradient/subgradient calculation – user can feed problem data in any solver
    - Heterogeneous constraint sets
    - No differentiability assumptions

Comparison with subgradient

Optimal power allocation in cellular networks (non-differentiable objective)
proposed solution vs. gradient-based approach



Building district problem revisited – Simulation results

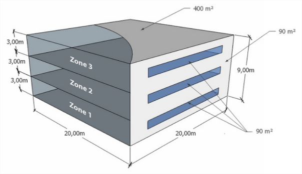
Set-up

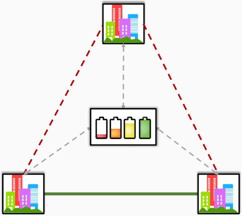
- 3 buildings - 3 zones each (different chiller per building)
- Pair-wise communication (gossip)

Building district problem revisited – Simulation results

Set-up

- 3 buildings - 3 zones each (different chiller per building)
- Pair-wise communication (gossip)

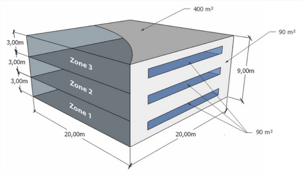


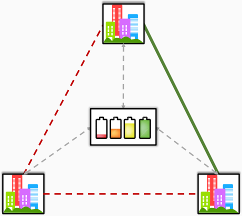


Building district problem revisited – Simulation results

Set-up

- 3 buildings - 3 zones each (different chiller per building)
- Pair-wise communication (gossip)

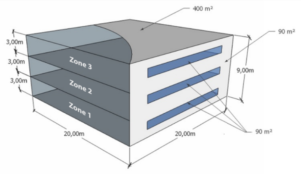


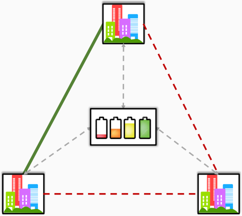


Building district problem revisited – Simulation results

Set-up

- 3 buildings - 3 zones each (different chiller per building)
- Pair-wise communication (gossip)

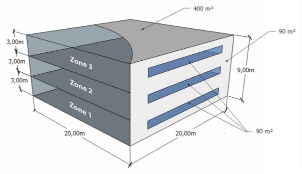


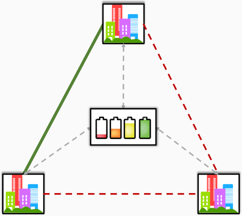


Building district problem revisited – Simulation results

Set-up

- 3 buildings - 3 zones each (different chiller per building)
- Pair-wise communication (gossip)



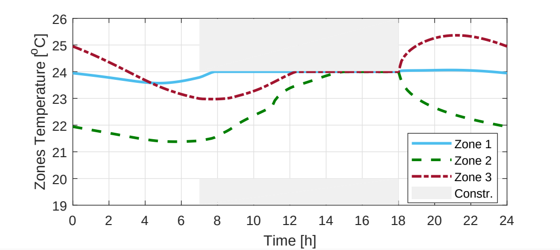


Implementation

- Simulation in MATLAB
- Optimization solver SEDUMI via the MATLAB interface YALMIP

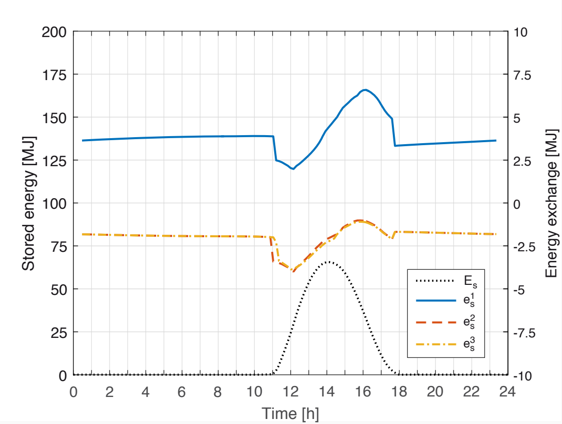
Simulation results – Temperature set-points

Optimal zone temperature profiles of building 1 (consensus solution).



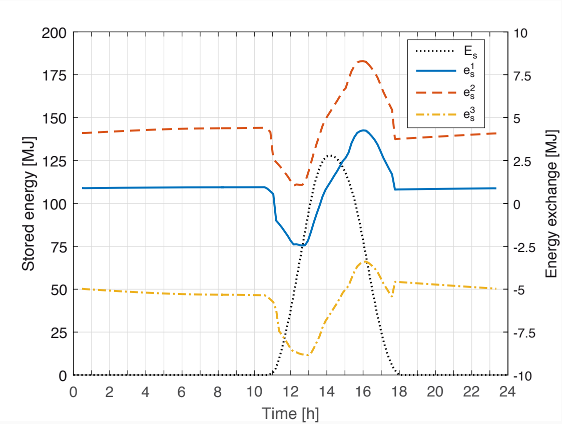
Temperature of zone 2 (middle one) is always the lowest, it acts as a passive thermal storage draining heat of the other zones through floor/ceiling.

Simulation results – Storage usage



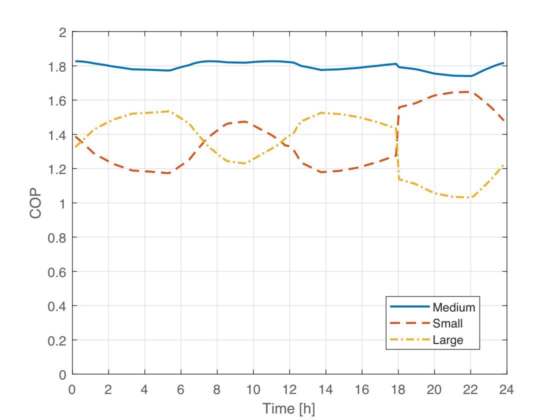
Solution computed at iteration $k = 1$ by the middle-chiller building ("blue"). The middle-chiller building uses the storage charged by the others

Simulation results – Storage usage



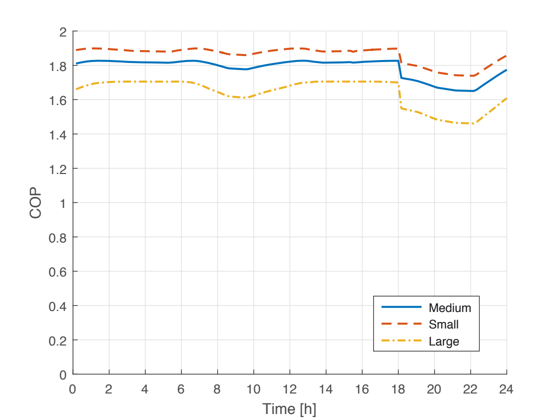
At consensus, the small-chiller building (“orange”) uses the storage charged by the others

Simulation results – Chillers usage



COP of the chillers when each building uses a fix fraction of the storage

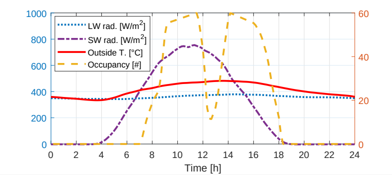
Simulation results – Chillers usage



COP of the chillers in the optimally shared storage case

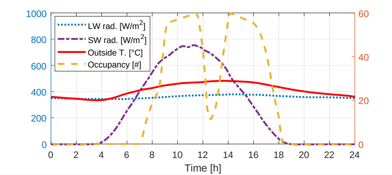
Simulation results

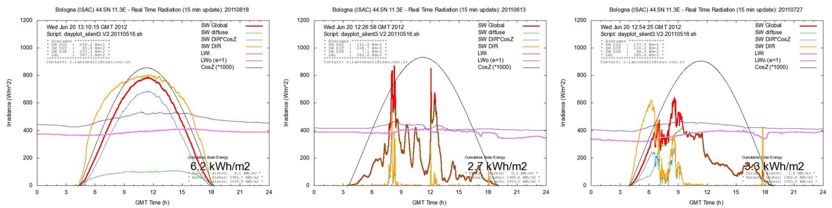
Solution computed based on nominal disturbance profiles...



Simulation results

Solution computed based on nominal disturbance profiles...





Courtesy of Istituto di Scienze dell'Atmosfera e del Clima (ISAC) - CNR

Problem set-up

Decision-coupled problem

$$\text{minimize} \ \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i X_i$$

Problem set-up

Decision-coupled problem with uncertainty

$$\text{minimize} \ \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i X_i(\delta), \ \text{for all } \delta \in \Delta$$

Problem set-up

Decision-coupled problem with uncertainty

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i X_i(\delta), \text{ for all } \delta \in \Delta$$

- Stochastic set-up
  - δ: Uncertain parameter $\delta \sim \mathbb{P}$
  - Δ: (Possibly) continuous set
  - Semi-infinite optimization program

Problem set-up

Decision-coupled problem with uncertainty

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in \Delta} X_i(\delta)$$

- Stochastic set-up
  - δ: Uncertain parameter $\delta \sim \mathbb{P}$
  - Δ: (Possibly) continuous set
  - Semi-infinite optimization program

Data-based approach

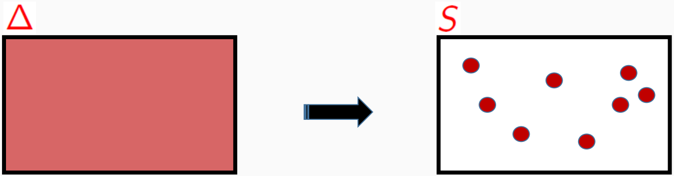
Decision-coupled problem with uncertainty

$$\text{minimize } \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

- Replace Δ with S



Data-based approach

Decision-coupled problem with uncertainty

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

Two cases:

1. Agents have the same data set S

Data-based approach

Decision-coupled problem with uncertainty

$$\text{minimize} \sum_i f_i(x)$$

$$\text{subject to}$$

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

Two cases:

1. Agents have the same data set S
2. Agents have different data sets $\{S_i\}_i$

Data-based approach

Decision-coupled problem with uncertainty

$$\text{minimize} \sum_i f_i(x)$$

$$\text{subject to}$$

$$x \in \bigcap_i \bigcap_{\delta \in S_i} X_i(\delta)$$

Two cases:

1. Agents have the same data set S
2. Agents have different data sets $\{S_i\}_i$

Data-based approach

Common data set – distributed implementation

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

Data-based approach

Common data set – distributed implementation

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

- Apply proximal algorithm with $\bigcap_{\delta \in S} X_i(\delta)$ in place of X_i

Data-based approach

Common data set – distributed implementation

$$\text{minimize } \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

- Apply proximal algorithm with $\bigcap_{\delta \in S} X_i(\delta)$ in place of X_i
- Let x_S^* denote the converged solution

Probabilistic feasibility – Common data set

Data-based program $\mathcal{P}_S$

$$\text{minimize} \sum_i f_i(x)$$

subject to $\rightarrow x^*_S$

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

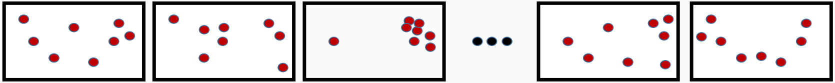
Robust program $\mathcal{P}_\Delta$

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in \Delta} X_i(\delta)$$

- Is x^*_S feasible for $\mathcal{P}_\Delta$?



Probabilistic feasibility – Common data set

Data-based program $\mathcal{P}_S$

$$\text{minimize} \sum_i f_i(x)$$

subject to $\rightarrow x_S^*$

$$x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta)$$

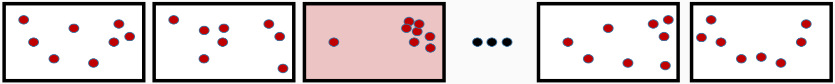
Robust program $\mathcal{P}_\Delta$

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in \Delta} X_i(\delta)$$

- Is x_S^* feasible for $\mathcal{P}_\Delta$?
- Is this true for any S?



Probabilistic feasibility – Common data set

Data-based program $\mathcal{P}_S$

$$\begin{aligned} &\text{minimize} \ \sum_i f_i(x) \\ &\text{subject to} \\ &x \in \bigcap_i \bigcap_{\delta \in S} X_i(\delta) \end{aligned} \quad \rightarrow x^*_S$$

Robust program $\mathcal{P}_\Delta$

$$\begin{aligned} &\text{minimize} \ \sum_i f_i(x) \\ &\text{subject to} \\ &x \in \bigcap_i \bigcap_{\delta \in \Delta} X_i(\delta) \end{aligned}$$

Feasibility link [Calafiore & Campi, TAC 2006]

Fix $\beta \in (0,1)$ and S. With confidence $\geq 1-\beta$, x^*_S is feasible with probability $\geq 1 - \epsilon(d, |S|, \beta)$, i.e.

$$\mathbb{P}\Big(\delta \in \Delta : \ x^*_S \notin \bigcap_i X_i(\delta)\Big) \leq \epsilon(d, |S|, \beta) \text{ with prob. } \geq 1 - \beta$$

Probabilistic feasibility – Common data set

Feasibility link

Fix $\beta \in (0,1)$ and S. With confidence $\geq 1-\beta$, x_S^* is feasible for $\mathcal{P}_\Delta$ with probability $\geq 1-\epsilon(d,|S|,\beta)$, i.e.

$$\mathbb{P}\Big(\delta \in \Delta: \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \epsilon(d,|S|,\beta) \text{ with prob. } \geq 1-\beta$$

- On which parameters does ϵ depends on?

$$\epsilon = \frac{2}{|S|}\Big(d + \ln\frac{1}{\beta}\Big)$$

  - Logarithmic in β: β can be set close to 0
  - Linear in $|S|^{-1}$: The more data the better the result
  - Linear in d: $\#$ decision variables

Data-based approach

Different data set – distributed implementation

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S_i} X_i(\delta)$$

- Apply proximal algorithm with $\bigcap_{\delta \in S_i} X_i(\delta)$ in place of X_i

Data-based approach

Different data set – distributed implementation

$$\text{minimize} \sum_i f_i(x)$$

subject to

$$x \in \bigcap_i \bigcap_{\delta \in S_i} X_i(\delta)$$

- Apply proximal algorithm with $\bigcap_{\delta \in S_i} X_i(\delta)$ in place of X_i
- Let x_S^* denote the converged solution, $S = \{S_i\}_i$

Probabilistic feasibility – Different data sets

Single-agent - a posteriori

Fix $\beta_i \in (0,1)$ and S_i. With confidence $\geq 1 - \beta_i$,

$$\mathbb{P}\Big(\delta \in \Delta : \ x_S^* \notin X_i(\delta)\Big) \leq \epsilon_i(d_i^{S_i}, |S_i|, \beta_i)$$

A posteriori result

- $d_i^{S_i}$: empirical estimate of "support" samples (wait and see)
  Changing S_i the result will change
- Complexity of $\epsilon_i(d_i^{S_i}, |S_i|, \beta_i)$ as in the previous case
- Result thanks to [Campi, Garatti & Ramponi, CDC 2015]

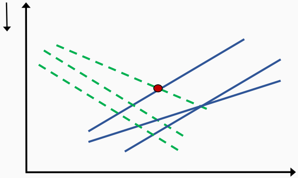
Probabilistic feasibility – Different data sets

Single-agent - a posteriori

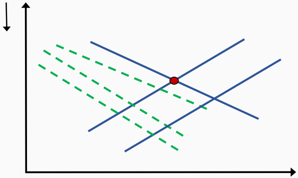
Fix $\beta_i \in (0,1)$ and S_i. With confidence $\geq 1 - \beta_i$,

$$\mathbb{P}\Big(\delta \in \Delta : \ x_S^* \notin X_i(\delta)\Big) \leq \epsilon_i(d_i^{S_i})$$

- Two-agent example, $d = 2$



$d_1^{S_1} = 1$ and $d_2^{S_2} = 1$



$d_1^{S_1} = 0$ and $d_2^{S_2} = 2$

Probabilistic feasibility – Different data sets

Multi-agent - a posteriori

Fix $\beta \in (0,1)$ and $\{S_i\}_i$. With confidence $\geq 1-\beta$,

$$\mathbb{P}\Big(\delta \in \Delta : \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \sum_i \epsilon_i(d_i^{S_i})$$

A posteriori result

- Can we turn it into an a priori statement?

Probabilistic feasibility – Different data sets

Multi-agent - a posteriori

Fix $\beta \in (0,1)$ and $\{S_i\}_i$. With confidence $\geq 1-\beta$,

$$\mathbb{P}\Big(\delta \in \Delta : \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \sum_i \epsilon_i(d_i^{S_i})$$

A posteriori result

- Can we turn it into an a priori statement?
- What is the worst-case value for $\sum_i \epsilon_i(d_i^{S_i})$ that we can "observe"?

Probabilistic feasibility – Different data sets

Multi-agent - a posteriori

Fix $\beta \in (0,1)$ and $\{S_i\}_i$. With confidence $\geq 1-\beta$,

$$\mathbb{P}\Big(\delta \in \Delta : \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \sum_i \epsilon_i(d_i^{S_i})$$

A posteriori result

- Can we turn it into an a priori statement?
- What is the worst-case value for $\sum_i \epsilon_i(d_i^{S_i})$ that we can "observe"?
- Conservative bound: $d_i^{S_i} \leq d$ for all i

Probabilistic feasibility – Different data sets

Multi-agent - a posteriori

Fix $\beta \in (0,1)$ and $\{S_i\}_i$. With confidence $\geq 1-\beta$,

$$\mathbb{P}\Big(\delta \in \Delta : \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \sum_i \epsilon_i(d_i^{S_i})$$

A posteriori result

- Can we turn it into an a priori statement?
- What is the worst-case value for $\sum_i \epsilon_i(d_i^{S_i})$ that we can "observe"?
- Conservative bound: $d_i^{S_i} \leq d$ for all i
- Sharper bound: $\sum_i d_i^{S_i} \leq d$ (# decision variables)

Probabilistic feasibility – Different data sets

Multi-agent - a priori

Fix $\beta \in (0,1)$ and $\{S_i\}_i$. With confidence $\geq 1-\beta$,

$$\mathbb{P}\Big(\delta \in \boldsymbol{\Delta} : \; x_S^* \notin \bigcap_i X_i(\delta)\Big) \leq \epsilon$$

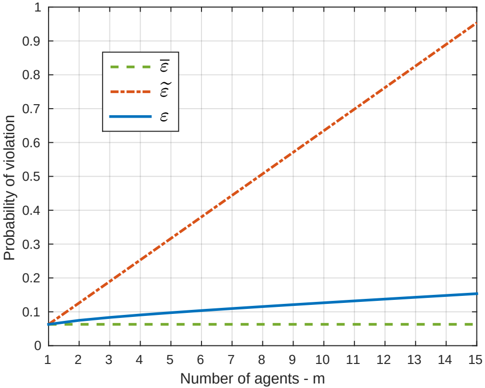
where

$$\epsilon \;=\; \text{maximize} \sum_i \epsilon_i(d_i)$$

$$\text{subject to}$$

$$\sum_i d_i \leq d$$

Common vs. different data sets



Approach using different constraint sets

- Close to the case of common data sets
- Less conservative than the worst case bound

Literature comparison

Closest approach[1]

- almost sure convergence results
  (need to sample constraints infinitely many times)

[1] S. Lee and A. Nedic, Distributed random projection algorithm for convex optimization, IEEE Journal on Selected Topics in Signal Processing 2013.

Literature comparison

Closest approach[1]

- almost sure convergence results
  (need to sample constraints infinitely many times)

Proposed solution

- weaker guarantees but with a finite number of samples

[1]S. Lee and A. Nedic, Distributed random projection algorithm for convex optimization, IEEE Journal on Selected Topics in Signal Processing 2013.

Addressed problems

$$\min_x \quad \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

Decision-coupled problem

Addressed problems

$$\min_{x} \quad \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

$$\min_{x_1,\dots,x_m} \quad \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \le 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

Decision-coupled problem

Addressed problems

$$\min_{x} \quad \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

Decision-coupled problem

$$\min_{x_1,\dots,x_m} \quad \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \leq 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i

Addressed problems

$$\min_{x} \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

$$\min_{x_1,\dots,x_m} \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \leq 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i
- local decisions x_i

Decision-coupled problem

Addressed problems

$$\min_{x} \quad \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

Decision-coupled problem

$$\min_{x_1,\dots,x_m} \quad \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \leq 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^{m} g_i(x_i) \leq 0$

Addressed problems

$$\min_{x} \quad \sum_{i=1}^{m} f_i(x)$$
$$\text{s.t.} \quad x \in \bigcap_{i=1}^{m} X_i$$

- local objectives f_i
- coupled decision x
- local constraints X_i

Decision-coupled problem

$$\min_{x_1,\dots,x_m} \quad \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \leq 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^{m} g_i(x_i) \leq 0$

Constraint-coupled problem

Proposed solution for constraint-coupled problems

$$\begin{aligned} \min_{x_1,\dots,x_m} \quad & \sum_{i=1}^{m} f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^{m} g_i(x_i) \le 0 \\ & x_i \in X_i \quad \forall i \end{aligned}$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^{m} g_i(x_i) \le 0$

Proposed solution for constraint-coupled problems

At each iteration k, agent i

$$\min_{x_1,\dots,x_m} \quad \sum_{i=1}^{m} f_i(x_i)$$
$$\text{s.t.} \quad \sum_{i=1}^{m} g_i(x_i) \leq 0$$
$$x_i \in X_i \quad \forall i$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^{m} g_i(x_i) \leq 0$

Proposed solution for constraint-coupled problems

At each iteration k, agent i

$$\ell_i(k) \leftarrow \sum_{j \in \mathcal{N}_i} a_j^i(k) \lambda_j(k)$$

$$\begin{aligned} \min_{x_1,\dots,x_m} \quad & \sum_{i=1}^m f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^m g_i(x_i) \leq 0 \\ & x_i \in X_i \quad \forall i \end{aligned}$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^m g_i(x_i) \leq 0$

Proposed solution for constraint-coupled problems

At each iteration k, agent i

$$\ell_i(k) \leftarrow \sum_{j \in \mathcal{N}_i} a_j^i(k) \lambda_j(k)$$

$$\lambda_i(k+1) \leftarrow \arg\max_{\lambda_i \geq 0} \tilde{\varphi}_i(\lambda_i)$$

where

$$\tilde{\varphi}_i(\lambda_i) = \lambda_i^\top g_i(x_i(k+1)) - \frac{1}{c(k)} \|\lambda_i - \ell_i(k)\|_2^2$$

$$\begin{aligned} \min_{x_1,\dots,x_m} \quad & \sum_{i=1}^m f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^m g_i(x_i) \leq 0 \\ & x_i \in X_i \quad \forall i \end{aligned}$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^m g_i(x_i) \leq 0$

Proposed solution for constraint-coupled problems

At each iteration k, agent i

$$\ell_i(k) \leftarrow \sum_{j \in \mathcal{N}_i} a^i_j(k)\lambda_j(k)$$

$$x_i(k+1) \leftarrow \arg\min_{x_i \in X_i} \; \tilde{f}_i(x_i)$$

$$\lambda_i(k+1) \leftarrow \arg\max_{\lambda_i \geq 0} \; \tilde{\varphi}_i(\lambda_i)$$

where

$$\tilde{f}_i(x_i) = f_i(x_i) + \ell_i(k)^\top g_i(x_i)$$

$$\tilde{\varphi}_i(\lambda_i) = \lambda_i^\top g_i(x_i(k+1)) - \frac{1}{c(k)} \|\lambda_i - \ell_i(k)\|_2^2$$

$$\begin{aligned} \min_{x_1,\ldots,x_m} \quad & \sum_{i=1}^m f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^m g_i(x_i) \leq 0 \\ & x_i \in X_i \quad \forall i \end{aligned}$$

- local objectives f_i
- local decisions x_i
- coupling constraint $\sum_{i=1}^m g_i(x_i) \leq 0$

Algorithm analysis

Main result (Convergence & optimality)

Under the structural + network assumptions, the proposed algorithm combining dual decomposition and proximal minimization converges to the set of minimizers of the centralized problem.

Algorithm analysis

Main result (Convergence & optimality)

Under the structural + network assumptions, the proposed algorithm combining dual decomposition and proximal minimization converges to the set of minimizers of the centralized problem.

Probabilistic feasibility results for the stochastic case have been developed.

Problem set-up – discrete decision variables

$$\mathcal{P}: \quad \min_{x_1,\ldots,x_m} \quad \sum_{i=1}^{m} c_i^\top x_i$$
$$\text{subject to:} \quad \sum_{i=1}^{m} A_i x_i \leq b$$
$$x_i \in X_i \quad \forall i = 1, \ldots, m$$

Features

- local decision vectors x_i
- local *linear* objectives $c_i^\top x_i$
- p coupling *linear* constraints $\sum_{i=1}^{m} A_i x_i \leq b$
- local *mixed-integer polyhedral constraint* sets X_i

Problem set-up – discrete decision variables

$$
\begin{aligned}
\mathcal{P}: \quad & \min_{x_1,\ldots,x_m} && \sum_{i=1}^{m} c_i^\top x_i \\
& \text{subject to:} && \sum_{i=1}^{m} A_i x_i \leq b \\
& && x_i \in X_i \quad \forall i = 1, \ldots, m
\end{aligned}
$$

Features

- local decision vectors x_i
- local *linear* objectives $c_i^\top x_i$
- p coupling *linear* constraints $\sum_{i=1}^{m} A_i x_i \leq b$
- local mixed-integer polyhedral constraint sets $X_i \Rightarrow$ combinatorial complexity

Constraint-coupled MILPs

The problem fits the structure of a constraint-coupled problem

Constraint-coupled MILPs

The problem fits the structure of a constraint-coupled problem

but...

It is non-convex, hence the distributed algorithms developed for convex problems have no guarantees

Constraint-coupled MILPs

The problem fits the structure of a constraint-coupled problem

but...

It is non-convex, hence the distributed algorithms developed for convex problems have no guarantees

we then aim at

1. providing a feasible (possibly sub-optimal) solution
2. quantifying the *quality* of the solution

Constraint-coupled MILPs

Goal

1. provide a feasible (possibly sub-optimal) solution
2. quantifying the *quality* of the solution

Literature

Some problem-specific approaches to recover a feasible solution

[Vujanic et al., 2016][2]

More general duality-based approach to recover a feasible solution with sub-optimality guarantees

[2] R. Vujanic, P. M. Esfahani, P. J. Goulart, S. Mariethoz, and M. Morari, A decomposition method for large scale MILPs, with performance guarantees and a power system application, Automatica, 2016

Proposed solution

Main idea of [Vujanic et al., 2016]

1. tighten the coupling constraint by a specific amount $\tilde{\rho} \geq 0$
2. obtain the dual optimal solution $\lambda^{\star}_{\tilde{\rho}}$
3. recover a feasible primal solution using $\lambda^{\star}_{\tilde{\rho}}$

Proposed solution

Main idea of [Vujanic et al., 2016]

1. tighten the coupling constraint by a specific amount $\tilde{\rho} \geq 0$
2. obtain the dual optimal solution $\lambda^{\star}_{\tilde{\rho}}$
3. recover a feasible primal solution using $\lambda^{\star}_{\tilde{\rho}}$

Proposed solution

A distributed iterative algorithm that merges:

- the procedure for solving constraint-coupled problems
- adaptive tightening of coefficient ρ based on the same idea of *[Vujanic et al., 2016]*

Theoretical results

Fact

By construction, $\rho \to \bar{\rho} \leq \tilde{\rho}$ (often $<$)

Theoretical results

Fact

By construction, $\rho \to \bar{\rho} \leq \tilde{\rho}$ (often $<$)

Theorem (Feasibility)

After a finite number of iterations, the algorithm provides a solution that is feasible for $\mathcal{P}$

Theoretical results

Fact

By construction, $\rho \to \bar{\rho} \leq \tilde{\rho}$ (often $<$)

Theorem (Feasibility)

After a finite number of iterations, the algorithm provides a solution that is feasible for $\mathcal{P}$

Theorem (Performance)

The performance are no-worse than that of [Vujanic et al., 2016] (often better)

Summary & Future work

Performance optimization of a network

- General distributed optimization framework accounting for different complexity features , i.e., heterogeneity of the agents, privacy of their local info, uncertainty, combinatorial complexity

Summary & Future work

Performance optimization of a network

- General distributed optimization framework accounting for different complexity features , i.e., heterogeneity of the agents, privacy of their local info, uncertainty, combinatorial complexity

Applications

- energy management of a building district
- power allocation in cellular networks
- plug-in electric vehicles charging scheduling

Summary & Future work

Performance optimization of a network

- General distributed optimization framework accounting for different complexity features , i.e., heterogeneity of the agents, privacy of their local info, uncertainty, combinatorial complexity

Applications

- energy management of a building district
- power allocation in cellular networks
- plug-in electric vehicles charging scheduling

What comes next?

- Convergence rate analysis
- Rolling horizon implementations
- Uncertain constraint-coupled MILP
- Application to Mixed Logical Dynamical (MLD) systems
- More applications

Main references

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

Falsone, Margellos, Garatti, Prandini (2017)
Dual decomposition for multi-agent distributed optimization with coupling constraints.
Automatica, 84:149–158.

Falsone, Margellos, Prandini (2018)
A distributed iterative algorithm for multi-agent MILPs: Finite-time feasibility and performance characterization.
IEEE Control Systems Letters, 2(4):563–568.

Credit

Alessandro Falsone

POLITECNICO
MILANO 1863

Simone Garatti

POLITECNICO
MILANO 1863

Kostas Margellos

UNIVERSITY OF
OXFORD

Credit

Work supported by the European Commission under the UnCoVerCPS project *Unifying Control and Verification of Cyber-Physical Systems*, H2020, 2015-2018