Security of distributed Model Predictive Control under False Data Injection

Rafael Accácio NOGUEIRA
rafael.accacio.nogueira@gmail.com

Seminar
École Centrale de Lyon / Laboratoire Ampère
26/05/2023 @ Écully

https://bit.ly/3g3S6X4

About me

Rafael Accácio Nogueira

Postdoctoral researcher at LAAS/CNRS

Garanteed relative localisation and anticollision scenario for autonomous vehicles

Project AutOCampus (GIS neOCampus)

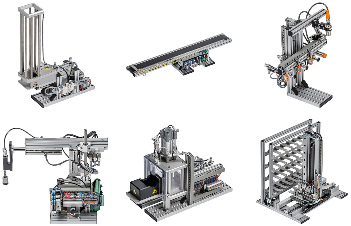
Advised by Soheib Fergani

About me

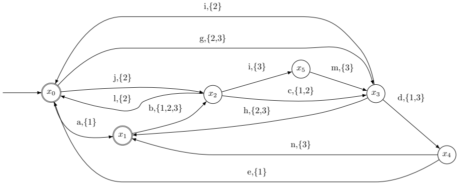
Bachelor Thesis at Escola Politécnica/UFRJ

Identification of DES for fault-diagnosis

Advised by Marcos Vicente de Brito Moreira



I/O
+
DAOCT

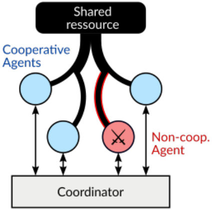


About me

Doctoral Thesis at CentraleSupélec/IETR

Security of dMPC under False Data Injection

Advised by Hervé Guéguen and Romain Bourdais



Context

Smart(er) Cities

Multiple systems interacting

Smart(er) Cities

Multiple systems interacting

Context

Smart(er) Cities

Multiple systems interacting

- Distribution:

Smart(er) Cities

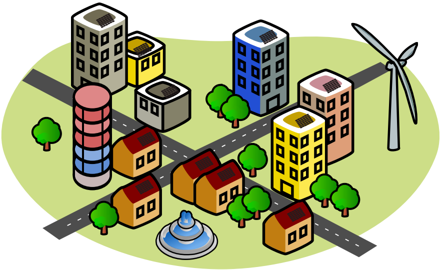
Multiple systems interacting

- Distribution:
  - Electricity

Context

Smart(er) Cities

Multiple systems interacting



- Distribution:
  - Electricity
  - Heat
  - Water

Context

Smart(er) Cities

Multiple systems interacting

- Distribution:
  - Electricity
  - Heat
  - Water
- Traffic

Context

Smart(er) Cities

Multiple systems interacting

- Distribution:
  - Electricity
  - Heat
  - Water
- Traffic

...

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption
  - Maximizer satisfaction

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption
  - Maximizer satisfaction
  - Follow a trajectory

Context

Smart(er) Cities

Multiple systems interacting under

- Technical/Comfort Constraints
- We also want
  - Minimize consumption
  - Maximizer satisfaction
  - Follow a trajectory
- Solution → MPC

Model-based Predictive Control

Brief recap

Model-based Predictive Control

Brief recap

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Brief recap

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Model-based Predictive Control

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N

$$\underset{\boldsymbol{u}[0:{\color{red}N}-1|k]}{\text{minimize}} \qquad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:{\color{red}N}-1|k])$$

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize

$$\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k])$$

Brief recap

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model

$$\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{aligned} \boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ \\ \end{aligned}\right\} \forall \xi \in \{1, \dots, N\}
\end{aligned}$$

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model
  - Other constraints to respect

$$\begin{array}{ll} \underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\ \text{subject to} & \left.\begin{array}{c} \boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leqslant 0 \\ h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0 \end{array}\right\} \begin{array}{l} \forall \xi \in \{1, \dots, N\} \\ \forall i \in \{1, \dots, m\} \\ \forall j \in \{1, \dots, p\} \end{array} \end{array}$$

Model-based Predictive Control

Brief recap

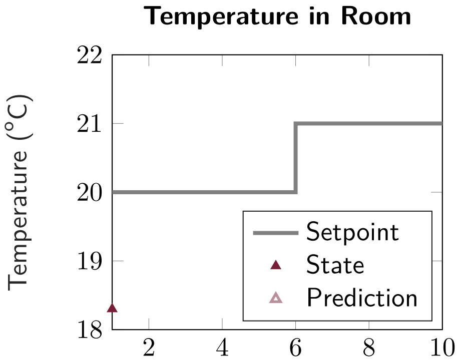
Find optimal control sequence using predictions based on a model.

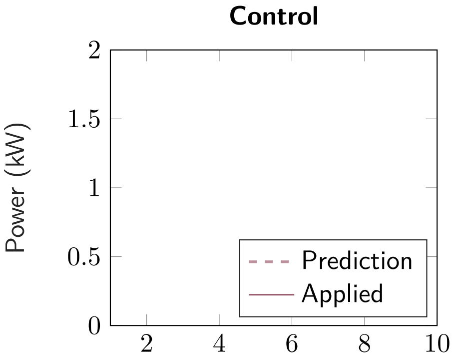
- We need an optimization problem
  - Decision variable is the control sequence calculated over horizon N
  - Objective function to optimize
  - System's Model
  - Other constraints to respect (QoS, technical restrictions, ...)

$$
\begin{array}{ll}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} & \left.\begin{array}{c}
\boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\
g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \leqslant 0 \\
h_j(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) = 0
\end{array}\right\} \begin{array}{l}
\forall \xi \in \{1, \dots, N\} \\
\forall i \in \{1, \dots, m\} \\
\forall j \in \{1, \dots, p\}
\end{array}
\end{array}
$$

Model Predictive Control

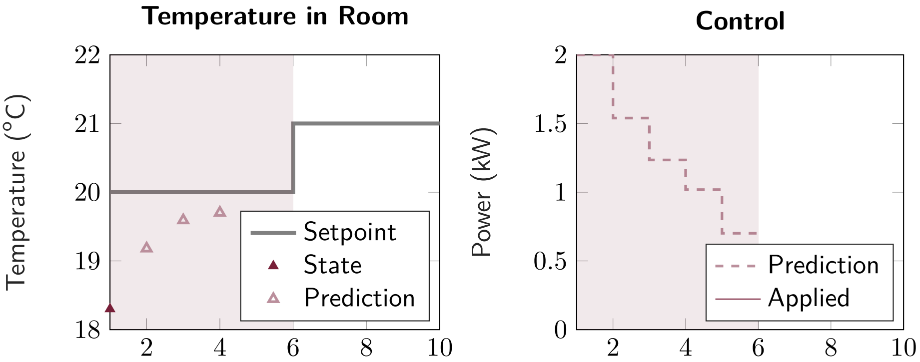
In a nutshell





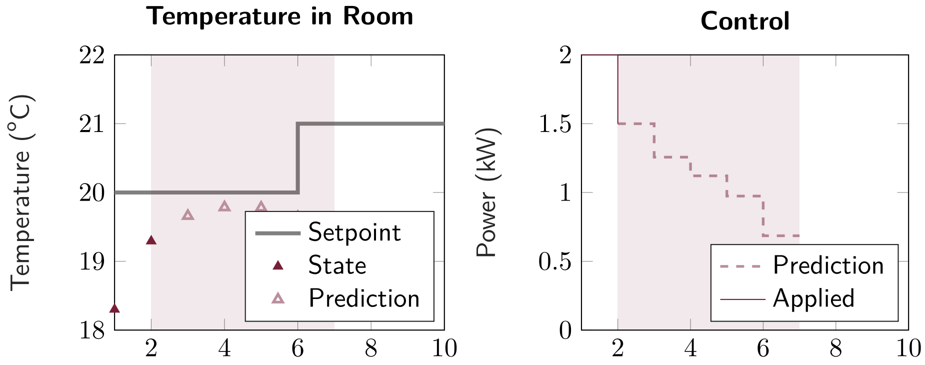
In a nutshell

Find optimal control sequence



In a nutshell

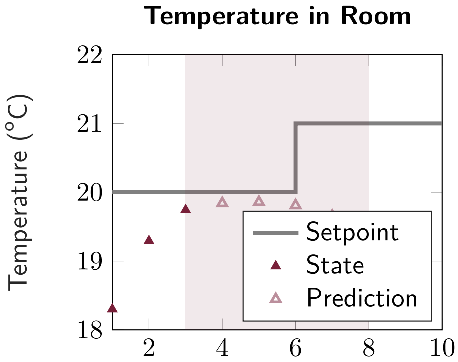
Find optimal control sequence, apply first element

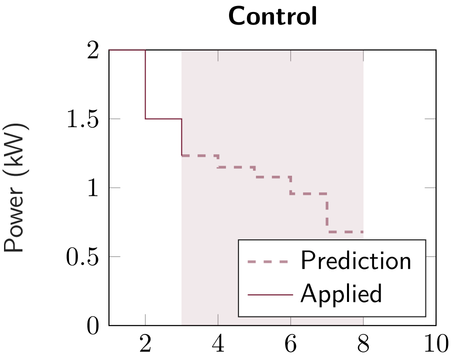


Model Predictive Control

In a nutshell

Find optimal control sequence, apply first element, rinse repeat

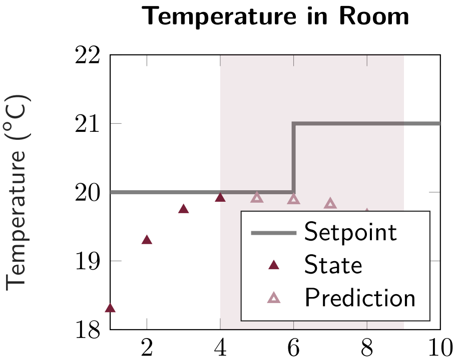


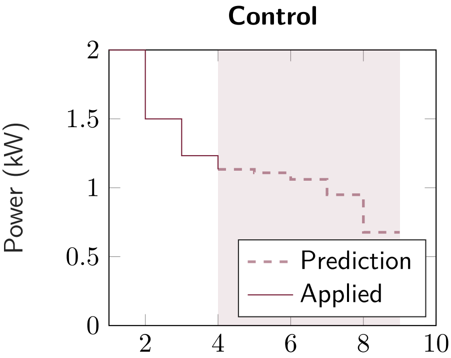


Model Predictive Control

In a nutshell

Find optimal control sequence, apply first element, rinse repeat → Receding Horizon

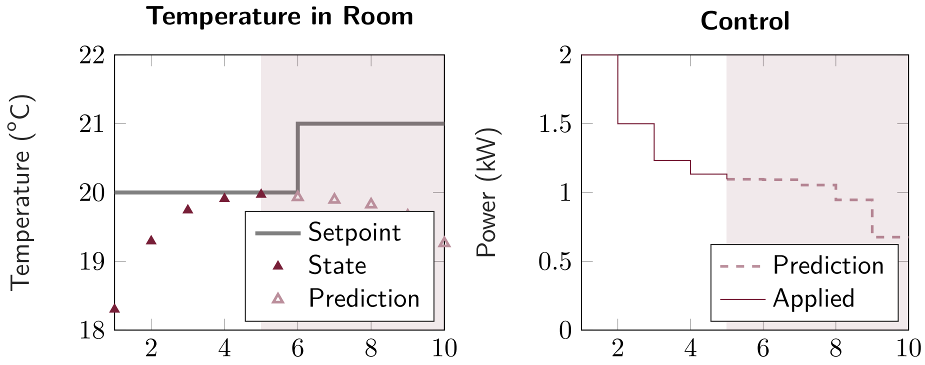




Model Predictive Control

In a nutshell

Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



Model Predictive Control

Nothing is perfect

Nothing is perfect

- Problems

Nothing is perfect

- Problems
  - Topology (Geographical distribution)

Model Predictive Control

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation
  - Flexibility (Add/remove parts)

Model Predictive Control

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation
  - Flexibility (Add/remove parts)
  - Privacy (RGPD)

Model Predictive Control

Nothing is perfect

- Problems
  - Topology (Geographical distribution)
  - Complexity of calculation
  - Flexibility (Add/remove parts)
  - Privacy (RGPD)
- Solution: Divide and Conquer (distributed MPC)

Outline

1. Decomposing the MPC

Outline

Outline

1. Decomposing the MPC
2. Attacks on the dMPC
3. Securing the dMPC

Outline

Outline

Distributed Model Predictive Control

Distributed Model Predictive Control

- We break the MPC optimization problem

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Until Convergence

Distributed Model Predictive Control

- We break the MPC optimization problem
- Make agents communicate

In other words

- Agents solve local problems
- Exchange some variables
- Variables are updated

Until Convergence

Remark

If agents exchange same variable → consensus problem

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

[1] Boyd et al., "Notes on Decomposition Methods"

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

---

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

- Cutting plane, sub-gradient methods, ...

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Many methods:

- Cutting plane, sub-gradient methods, . . .

---

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

Optimization Frameworks

Usually based on optimization decomposition methods[1]:

- Local problems with auxiliary variables
- Update auxiliary variables

Basically 2 choices[2]:

- Modify based on dual problem[3] (Solve with dual and send primal)
- Modify based on primal problem (Solve with primal and send dual)

Security/privacy properties

Many methods:

- Cutting plane, sub-gradient methods, . . .

[1] Boyd et al., "Notes on Decomposition Methods"
[2] Other approaches, but similar concepts
[3] Lagrangian, ADMM, prices, etc +1000 articles in scopus

Distributed Model Predictive Control

It is about communication

- We break the MPC optimization problem
- Make agents communicate.

Distributed Model Predictive Control

It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?

Distributed Model Predictive Control

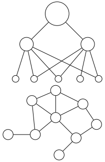
It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from

Distributed Model Predictive Control

It is about communication

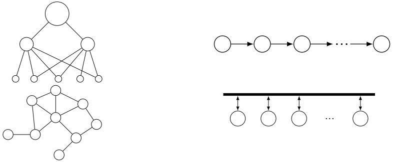
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical



Distributed Model Predictive Control

It is about communication

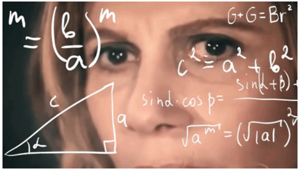
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Parallel/Sequential

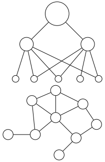


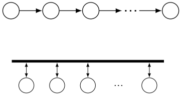
Distributed Model Predictive Control

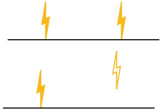
It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?
    - Many flavors to choose from
        - Hierarchical/Anarchical
        - Parallel/Sequential
        - Synchronous/Asynchronous





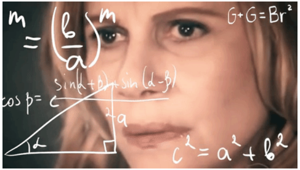


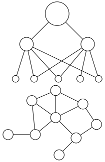


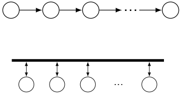
Distributed Model Predictive Control

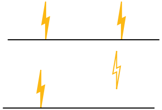
It is about communication

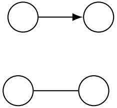
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional







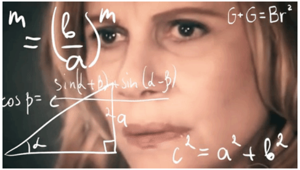


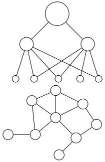


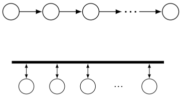
Distributed Model Predictive Control

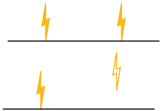
It is about communication

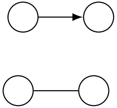
- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from[4]
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...









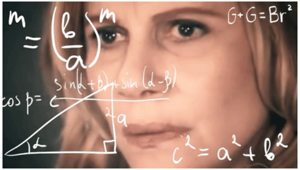


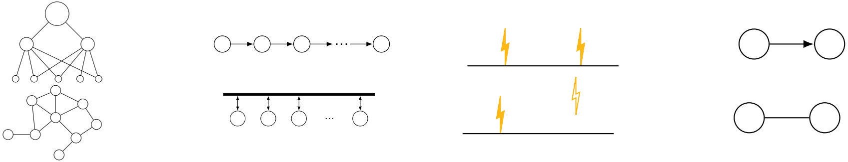
[4] José M Maestre, Negenborn, et al., Distributed Model Predictive Control made easy

Distributed Model Predictive Control

It is about communication

- We break the MPC optimization problem
- Make agents communicate. But how?
  - Many flavors to choose from[4]
    - Hierarchical/Anarchical
    - Parallel/Sequential
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...





[4] José M Maestre, Negenborn, et al., Distributed Model Predictive Control made easy

Distributed Model Predictive Control

Optimization Decomposition

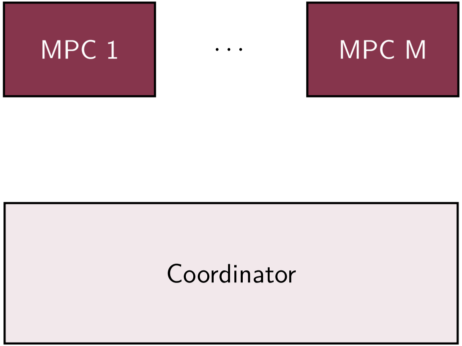
MPC

Distributed Model Predictive Control

Optimization Decomposition

Distributed Model Predictive Control

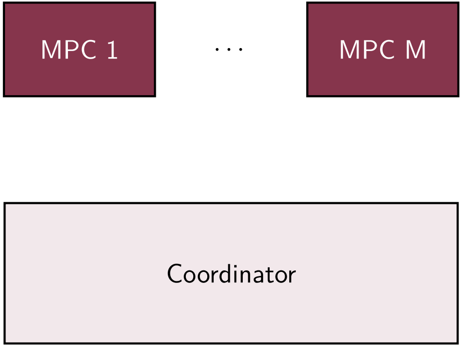
Optimization Decomposition



- Coordinator

Distributed Model Predictive Control

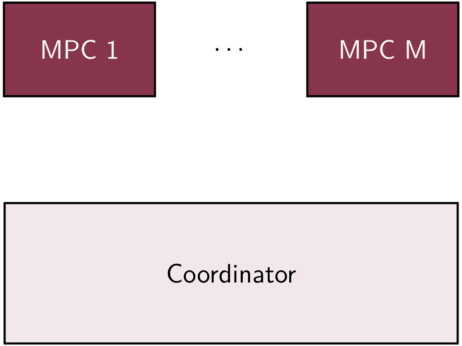
Optimization Decomposition



- Coordinator
  - Enforce global constraints

Distributed Model Predictive Control

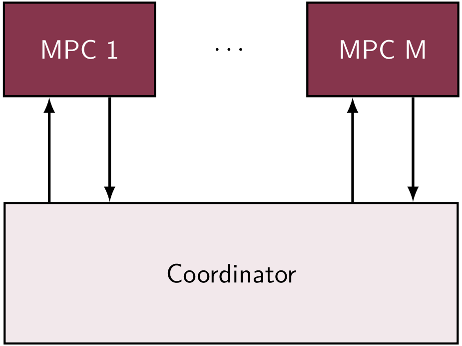
Optimization Decomposition



- Coordinator → Hierarchical
  - Enforce global constraints

Distributed Model Predictive Control

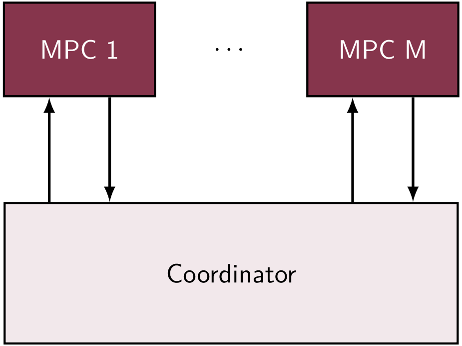
Optimization Decomposition



- Coordinator → Hierarchical
    - Enforce global constraints
- Bidirectional

Distributed Model Predictive Control

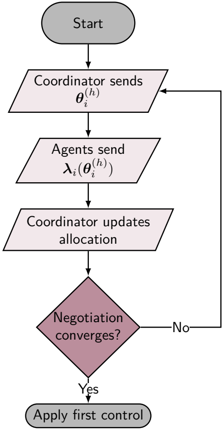
Optimization Decomposition



- Coordinator → Hierarchical
  - Enforce global constraints
- Bidirectional
- No delay → Synchronous

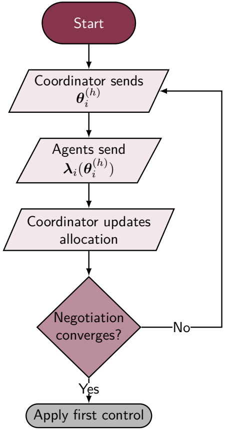
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



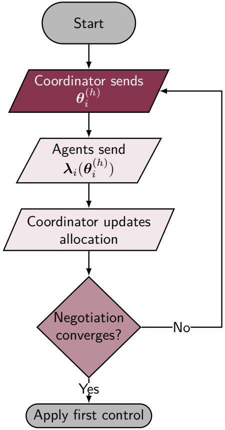
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Primal Decomposition

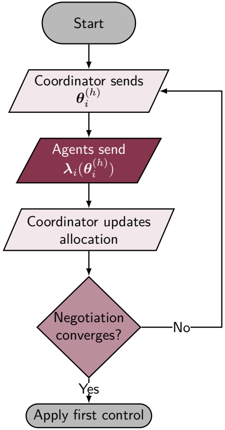
or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

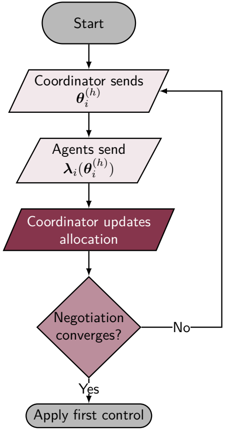


Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



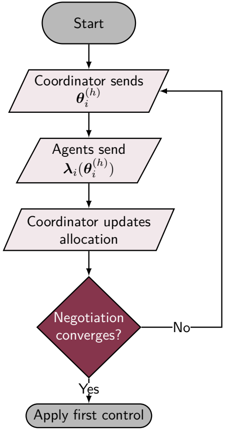
Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^+ = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



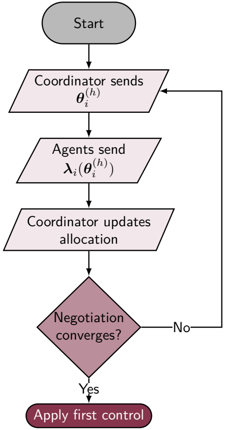
Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^{+} = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Update $\boldsymbol{\theta}_i^+ = f_i(\boldsymbol{\theta}_i, \boldsymbol{\lambda}_i)$

Primal Decomposition

In detail

$$\begin{aligned} \underset{\boldsymbol{u}_1,\dots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i,\boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i,\boldsymbol{u}_i) \preceq \boldsymbol{u}_{\mathsf{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones

$$\begin{aligned} \underset{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{\boldsymbol{u}_1,\dots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

$$
\begin{aligned}
\underset{\boldsymbol{u}_1,\dots,\boldsymbol{u}_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\
\text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}}
\end{aligned}
$$

$$\Big\downarrow \ \text{For each } i \in \mathcal{M}$$

$$
\begin{aligned}
\underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\
\text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i
\end{aligned}
$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$
4. Allocation is updated

$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)}$$

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

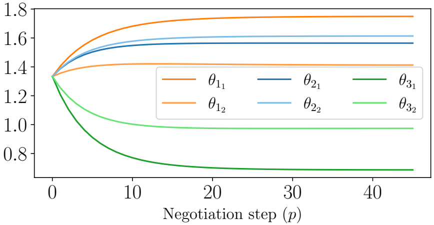
1. Allocate $\boldsymbol{\theta}_i$ for each agent
2. They solve local problems and
3. Send dual variable $\boldsymbol{\lambda}_i$
4. Allocation is updated (respect global constraint)

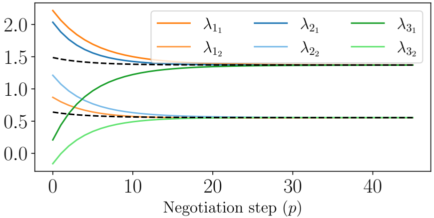
$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)})$$

Example

Until everybody is evenly[5] dissatisfied





[5] For inequality constraints dynamics are more complex

Distributed Model Predictive Control

Negotiation works if agents comply.

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

Problem recent in dMPC literature[6] (First article from 2017[7])

[6] <30 documents in scopus

[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

Problem recent in dMPC literature[6] (First article from 2017[7])

- CentraleSupélec Rennes - Expertise in MPC for Smart Buildings

[6] $<$30 documents in scopus

[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

Problem recent in dMPC literature[6] (First article from 2017[7])

- CentraleSupélec Rennes - Expertise in MPC for Smart Buildings
- Incentive Brittany Region (Sustainable Energy + cybersecurity)

[6] <30 documents in scopus
[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Distributed Model Predictive Control

Negotiation works if agents comply.
But what if some agents are ill-intentioned and attack the system?

Problem recent in dMPC literature[6] (First article from 2017[7])

- CentraleSupélec Rennes - Expertise in MPC for Smart Buildings
- Incentive Brittany Region (Sustainable Energy + cybersecurity)

- How can an agent attack?
- What are the consequences of an attack?
- Can we mitigate the effects? How?

---

[6] $<$30 documents in scopus

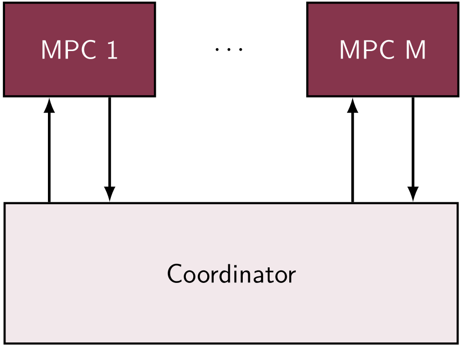
[7] Velarde, Jose Maria Maestre, H. Ishii, et al., "Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security"

Outline

2. Attacks on the dMPC

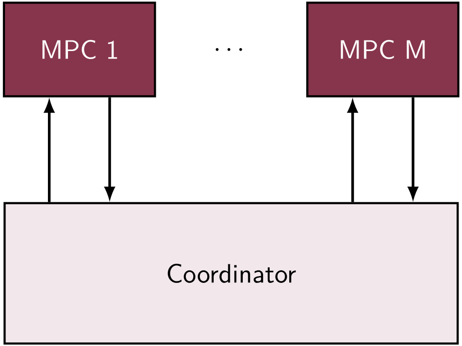
How can a non-cooperative agent attack?

Literature



How can a non-cooperative agent attack?

Literature

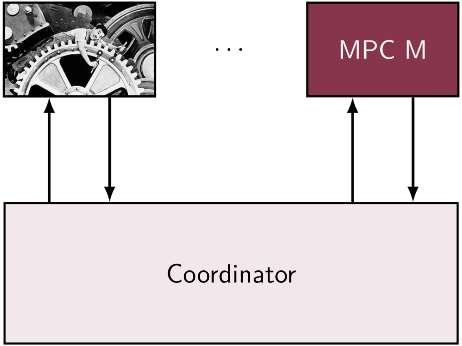


- Common attacks[8]

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Literature

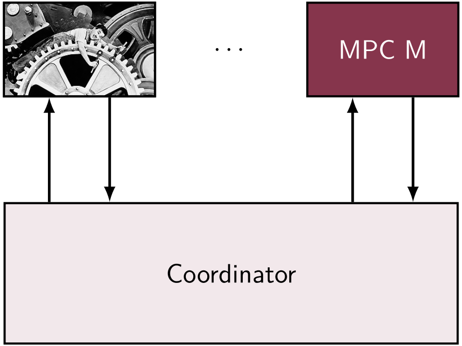


- Common attacks[8]

[8]Velarde, Jose Maria Maestre, Hideaki Ishii, et al., “Scenario-based defense mechanism for distributed model predictive control”

How can a non-cooperative agent attack?

Literature

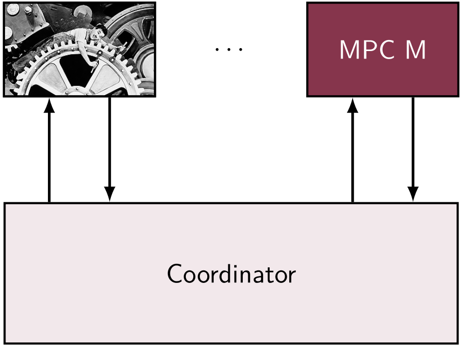


- Common attacks[8]
  - Fake objective function
  - Fake constraints
  - Use different control

[8] Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Literature



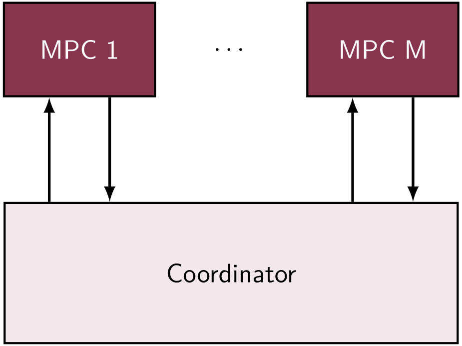
- Common attacks[8]
  - Fake objective function
  - Fake constraints
  - Use different control

  } Deception Attacks

[8]Velarde, Jose Maria Maestre, Hideaki Ishii, et al., "Scenario-based defense mechanism for distributed model predictive control"

How can a non-cooperative agent attack?

Our approach[9]

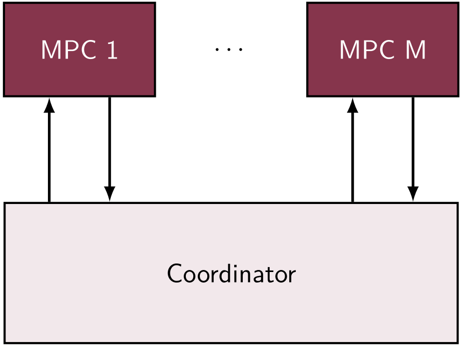


- Primal decomposition

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

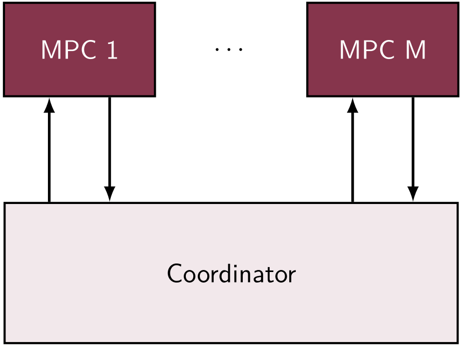


- Primal decomposition
  - Maximum resources fixed

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

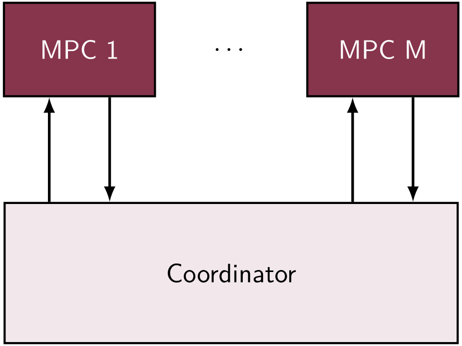


- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

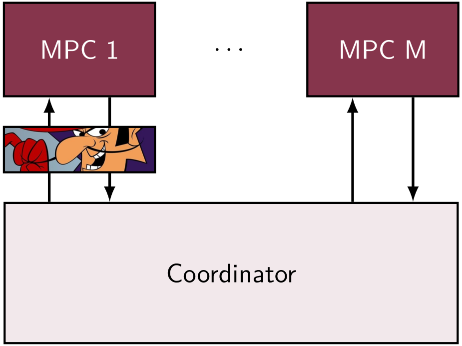


- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

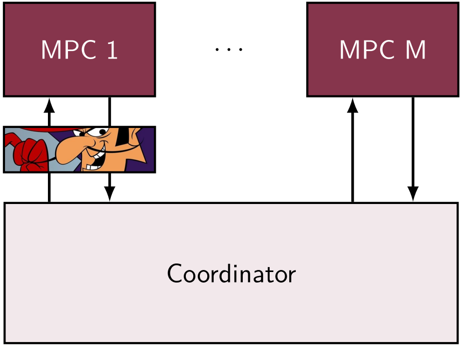


- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

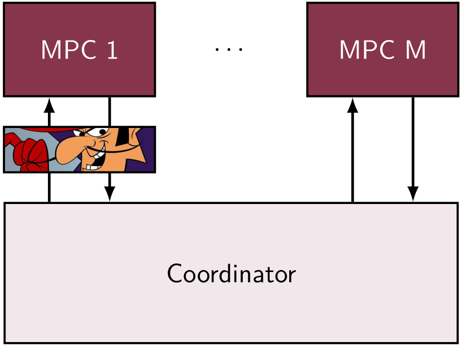


- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

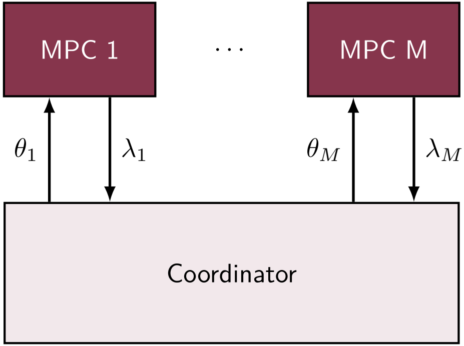


- Primal decomposition
  - Maximum resources fixed
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

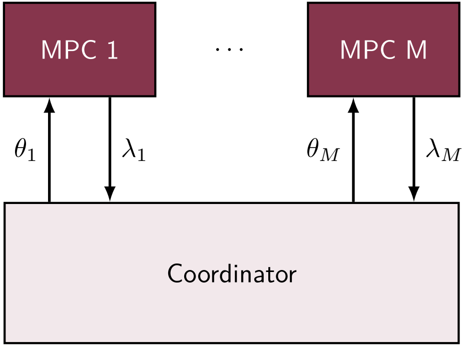


- $\boldsymbol{\lambda}_i$ is the only interface

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

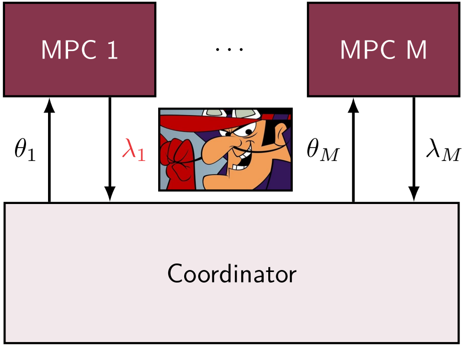


- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]

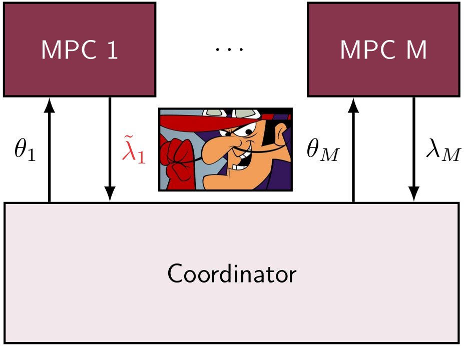


- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)
- Malicious agent modifies $\boldsymbol{\lambda}_i$

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

How can a non-cooperative agent attack?

Our approach[9]



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ obfuscate params. (+ Privacy)
- Malicious agent modifies $\boldsymbol{\lambda}_i$

$$\tilde{\boldsymbol{\lambda}}_i = \gamma_i(\boldsymbol{\lambda}_i)$$

[9] Nogueira, Bourdais, and Guéguen, "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation"

Attack model

Liar, Liar, Pants of fire

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

Attack model

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$

Attack model

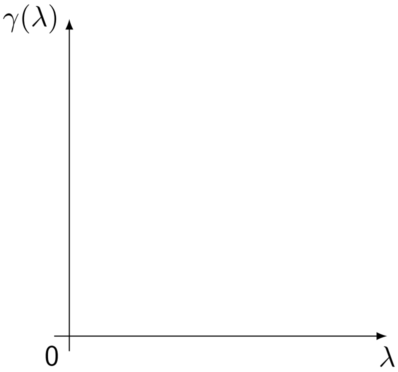
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



Attack model

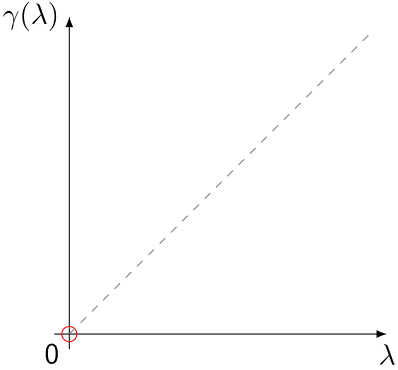
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



Attack model

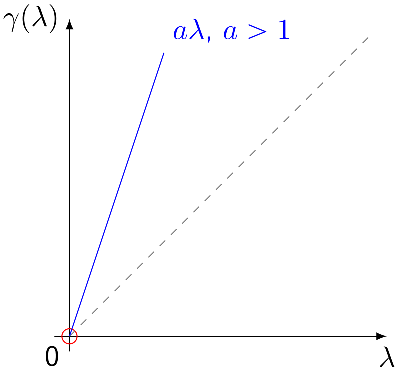
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction
- $\lambda = 0$ means complete satisfaction

Assumptions

- *Same attack during negotiation*
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

- Attack is invertible $\rightarrow \exists T_i[k]^{-1}$



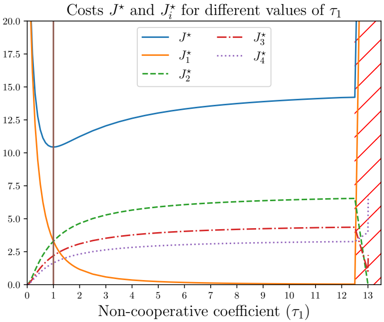
Example

Example

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

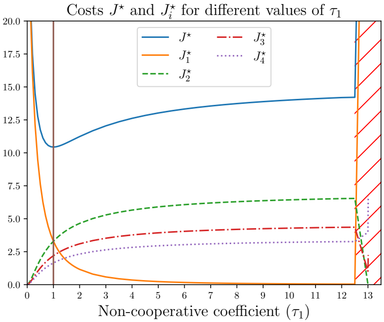
Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

Example

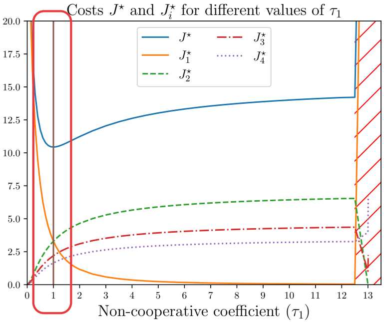


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things

Example

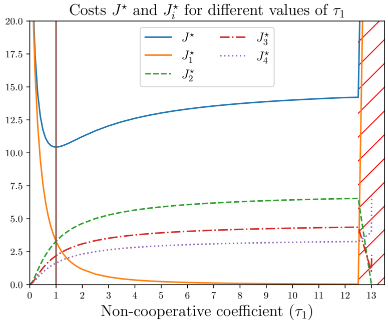


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$

Example

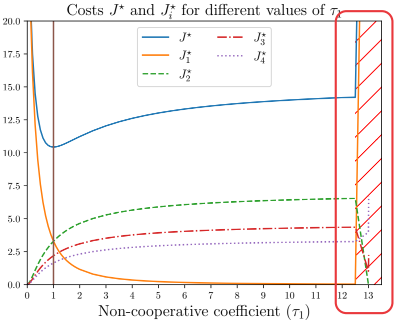


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$
  - Agent 1 benefits if τ_1 increases (inverse otherwise)

Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$
- Simulate for different τ_1 get J_i

- We can observe 3 things
  - Global minimum when $\tau_1 = 1$
  - Agent 1 benefits if τ_1 increases (inverse otherwise)
  - All collapses if too greedy

- But can we mitigate these effects?

- But can we mitigate these effects?
- Yes! (At least in some cases)

Outline

3 Securing the dMPC
- Classification
- State of Art
- Proposed Methods

Classification of mitigation techniques

Passive (Robust)

Active (Resilient)

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected
     - Detection/Isolation
     - Mitigation

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes
  1. Attack free
  2. When attack is detected
     - Detection/Isolation
     - Mitigation

State of art

Security dMPC

	Decomposition	Resilient/Robust	Detection	Mitigation
[10]	Dual	Robust (Scenario)	NA	NA
[11]	Dual	Robust (f-robust)	NA	NA
[12]	Jacobi-Gauß	–	–	–
[13]	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)

[10] José M. Maestre et al., "Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc".

[11] Velarde, José M. Maestre, et al., "Vulnerabilities in Lagrange-Based Distributed Model Predictive Control".

[12] Chanfreut, J. M. Maestre, and H. Ishii, "Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition".

[13] Ananduta et al., "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids".

Our Approach

Explore Scarcity

- Resilient
- Analytical/Learning
- Data reconstruction

Our Approach

Explore Scarcity

- Resilient
- Analytical/Learning
- Data reconstruction

Parameter Estimation

Our Approach

Explore Scarcity

- Resilient
- Analytical/Learning
- Data reconstruction

$\left.\begin{array}{l}\text{Analytical/Learning}\\ \text{Data reconstruction}\end{array}\right\}$ Parameter Estimation

- Explore Scarcity

Outline

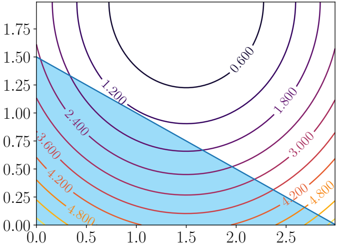
What are deprived systems?

What are deprived systems?

Systems whose optimal solution has all constraints active

What are deprived systems?

Systems whose optimal solution has all constraints active

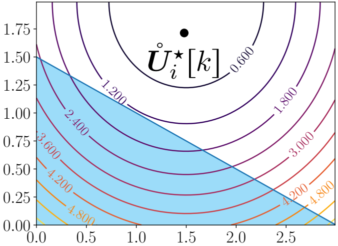


$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|^2_{H_i} + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$

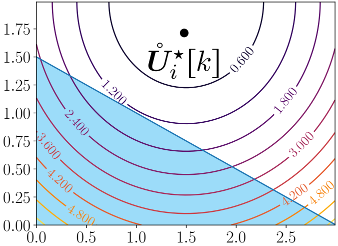


$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\|\boldsymbol{U}_i[k]\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $h_i(\mathring{\boldsymbol{U}}_i^\star[k]) \succ \boldsymbol{\theta}_i[k] \rightarrow$ Scarce resources

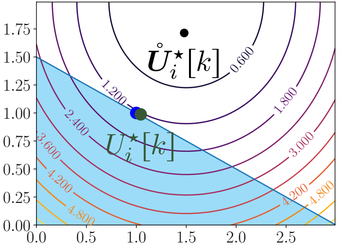


$$
\begin{array}{ll}
\underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\
\text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k]
\end{array}
$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $h_i(\mathring{\boldsymbol{U}}_i^\star[k]) \succ \boldsymbol{\theta}_i[k] \rightarrow$ Scarce resources
  - Solution projected onto boundary

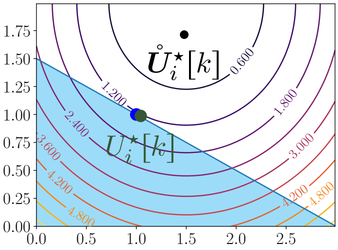


$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $h_i(\mathring{\boldsymbol{U}}_i^\star[k]) \succ \boldsymbol{\theta}_i[k] \rightarrow$ Scarce resources
  - Solution projected onto boundary
  - Same as with equality constraints[14]



$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \leq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array} \longrightarrow \begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

[14] If system can have all constraints active simultaneously ▸ see here

Analyzing Deprived Systems

Assumptions

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Solution is analytical and affine

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

- Solution is analytical and affine

$$\begin{aligned} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{aligned}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- P_i is time invariant

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

- Solution is analytical and affine

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- P_i is time invariant
- $\boldsymbol{s}_i[k]$ is time variant

Analyzing Deprived Systems

Assumptions

- *Quadratic local problems*
- *Linear inequality constraints*
- *Scarcity*

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] = \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

- Solution is analytical and affine

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

(local parameters unknown by coordinator) $\left\{ \begin{array}{l} \bullet\ P_i \text{ is time invariant} \\ \bullet\ \boldsymbol{s}_i[k] \text{ is time variant} \end{array} \right.$

Deprived Systems

Under attack!

- Normal behavior

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

$$\tilde{\boldsymbol{\lambda}}_i[k] = -T_i[k]P_i\boldsymbol{\theta}_i[k] - T_i[k]\boldsymbol{s}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$
  - Parameters modified

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$
  - Parameters modified

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

- But wait! P_i is not supposed to change!

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$
  - Parameters modified

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

- But wait! P_i is not supposed to change!
- Change $\rightarrow$ Probably an Attack!

Deprived Systems

Under attack!

- Normal behavior
  - Affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

- Under attack $\rightarrow \tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$
  - Parameters modified

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\tilde{P}_i[k]\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i[k]$$

- But wait! P_i is not supposed to change!
- Change $\rightarrow$ Probably an Attack! Let's take advantage of this!

Detection Mechanism

Detection Mechanism

- We estimate[15] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\widehat{\tilde{P}}_i[k]\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

[15] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[15] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\widehat{\tilde{P}}_i[k]\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

Assumption

We can estimate $\bar{P}_i$ from a attack free negotiation

[15] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[15] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\widehat{\tilde{P}}_i[k]\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

Assumption

We can estimate $\bar{P}_i$ from a attack free negotiation

- If $\left\|\widehat{\tilde{P}}_i[k] - \bar{P}_i\right\|_F > \epsilon_P$

[15] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[15] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\widehat{\tilde{P}}_i[k]\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

Assumption

We can estimate $\bar{P}_i$ from a attack free negotiation

- If $\left\|\widehat{\tilde{P}}_i[k] - \bar{P}_i\right\|_F > \epsilon_P \rightarrow$ Attack

[15] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[15] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

$$\tilde{\boldsymbol{\lambda}}_i[k] = -\widehat{\tilde{P}}_i[k]\boldsymbol{\theta}_i - \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

Assumption

We can estimate $\bar{P}_i$ from a attack free negotiation

- If $\left\|\widehat{\tilde{P}}_i[k] - \bar{P}_i\right\|_F > \epsilon_P \rightarrow$ Attack
- Ok, but how can we estimate $\widehat{\tilde{P}}_i[k]$?

[15] Using Recursive Least Squares for example

Estimating $\widehat{\tilde{P}}_i[k]$

Estimating $\widehat{\tilde{P}}_i[k]$

- We estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS

Estimating $\widehat{\tilde{P}}_i[k]$

- We estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails

Estimating $\widehat{\tilde{P}}_i[k]$

- We estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$

Estimating $\widehat{\widetilde{P}}_i[k]$

- We estimate $\widehat{\widetilde{P}}_i[k]$ and $\widehat{\widetilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation

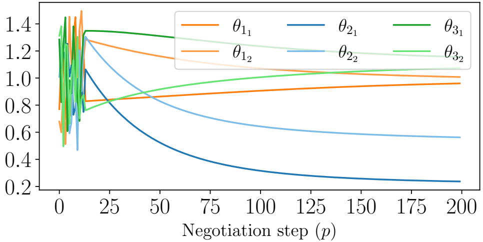
Estimating $\widehat{\tilde{P}}_i[k]$

- We estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation
- Solution: Send a random[16] sequence to increase excitation until convergence.

[16] A random signal causes persistent excitation of any order (Adaptive Control)

Estimating $\hat{\tilde{P}}_i[k]$

- We estimate $\hat{\tilde{P}}_i[k]$ and $\hat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation
- Solution: Send a random[16] sequence to increase excitation until convergence.



[16] A random signal causes persistent excitation of any order (Adaptive Control)

Classification of mitigation techniques

- Active (Resilient)
  1. Detection/Isolation ✔
  2. Mitigation

Classification of mitigation techniques

- Active (Resilient)
  1. Detection/Isolation ✔
  2. Mitigation ?

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- Now, we have $\widehat{\tilde{P}}_i[k]$

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- Now, we have $\widehat{\tilde{P}}_i[k]$
  - Since $\tilde{P}_i[k] = T_i[k]\bar{P}_i$

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- Now, we have $\widehat{\tilde{P}}_i[k]$
  - Since $\tilde{P}_i[k] = T_i[k]\bar{P}_i$
  - We can recover $T_i[k]^{-1}$

$$\widehat{T_i[k]^{-1}} = P_i \widehat{\tilde{P}}_i[k]^{-1}$$

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- Now, we have $\widehat{\tilde{P}}_i[k]$
  - Since $\tilde{P}_i[k] = T_i[k]\bar{P}_i$
  - We can recover $T_i[k]^{-1}$

$$\widehat{T_i[k]^{-1}} = P_i \widehat{\tilde{P}}_i[k]^{-1}$$

- Reconstruct $\boldsymbol{\lambda}_i$

$$\overset{\text{rec}}{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i[k]^{-1}}\widehat{\tilde{\boldsymbol{s}}}_i[k]$$

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- Now, we have $\widehat{\tilde{P}}_i[k]$
  - Since $\tilde{P}_i[k] = T_i[k]\bar{P}_i$
  - We can recover $T_i[k]^{-1}$

$$\widehat{T_i[k]^{-1}} = P_i \widehat{\tilde{P}}_i[k]^{-1}$$

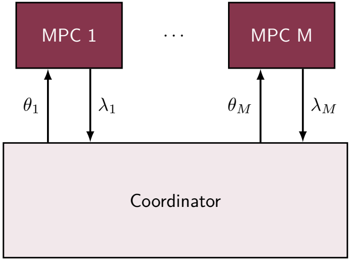
- Reconstruct $\boldsymbol{\lambda}_i$

$$\overset{\text{rec}}{\boldsymbol{\lambda}}_i = -\bar{P}_i \boldsymbol{\theta}_i - \widehat{T_i[k]^{-1}} \widehat{\tilde{\boldsymbol{s}}}_i[k]$$

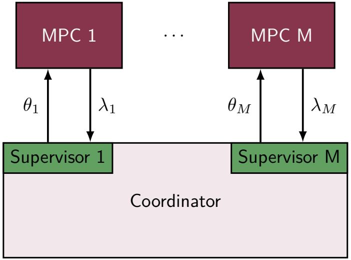
- Choose adequate version for coordination

$$\overset{\text{mod}}{\boldsymbol{\lambda}}_i = \begin{cases} \overset{\text{rec}}{\boldsymbol{\lambda}}_i, & \text{if attack detected} \\ \tilde{\boldsymbol{\lambda}}_i, & \text{otherwise} \end{cases}$$

Complete Mechanism

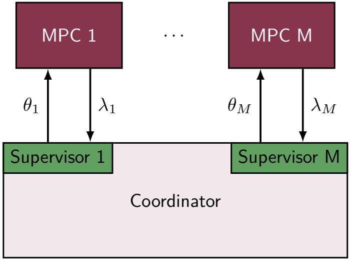


Complete Mechanism



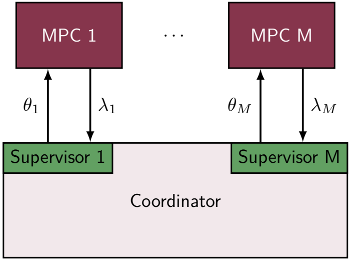
- Supervise exchanges by inquiring the agents

Complete Mechanism



- Supervise exchanges by inquiring the agents
- Estimate how they will behave

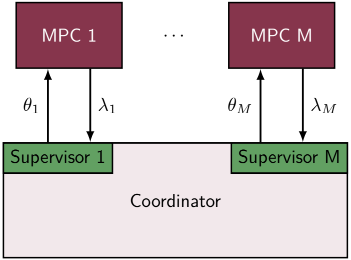
Complete Mechanism



- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

Complete Mechanism

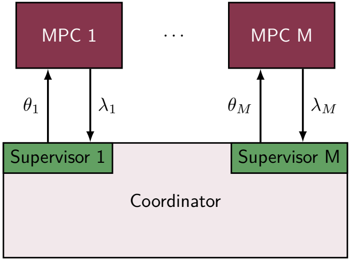


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative

Complete Mechanism



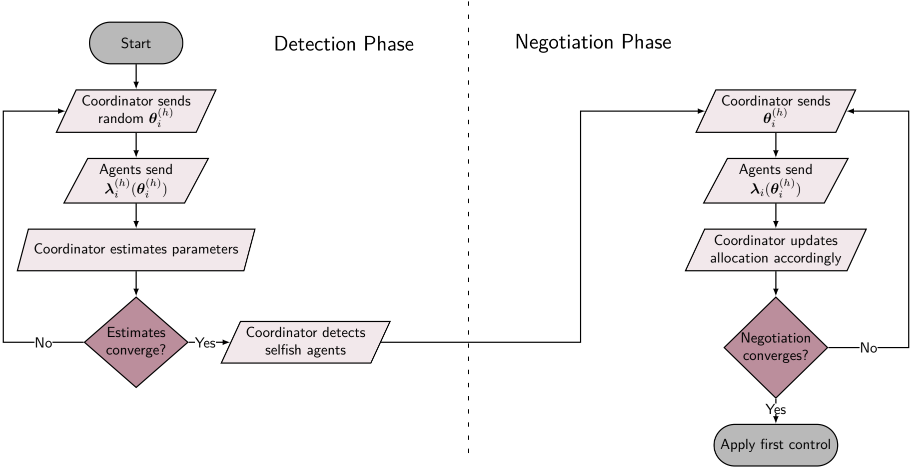
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct $\boldsymbol{\lambda}_i$ and use in negotiation

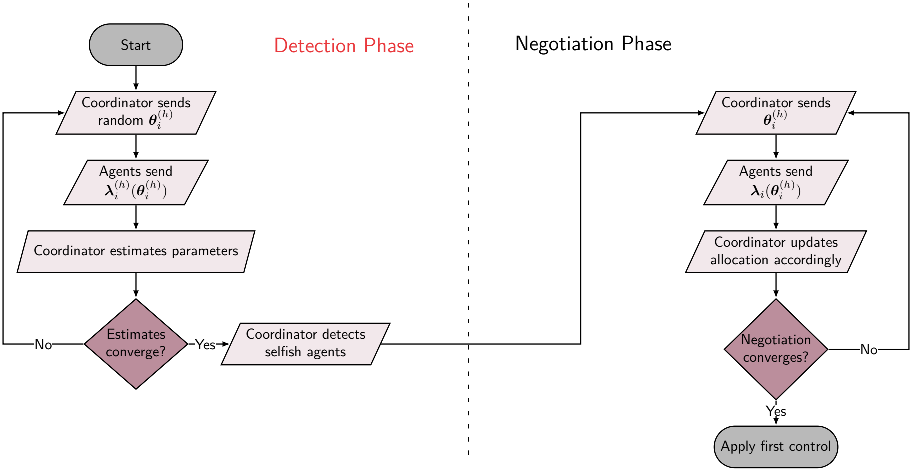
Complete algorithm

RPdMPC-DS



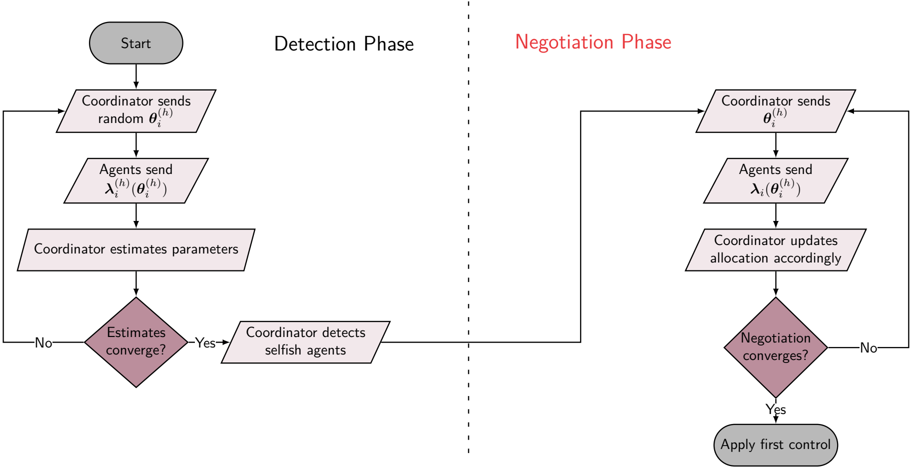
Complete algorithm

RPdMPC-DS



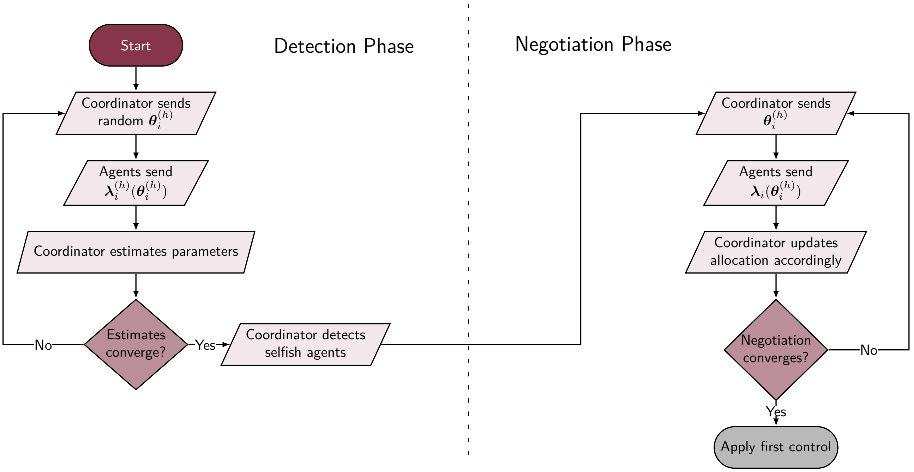
Complete algorithm

RPdMPC-DS



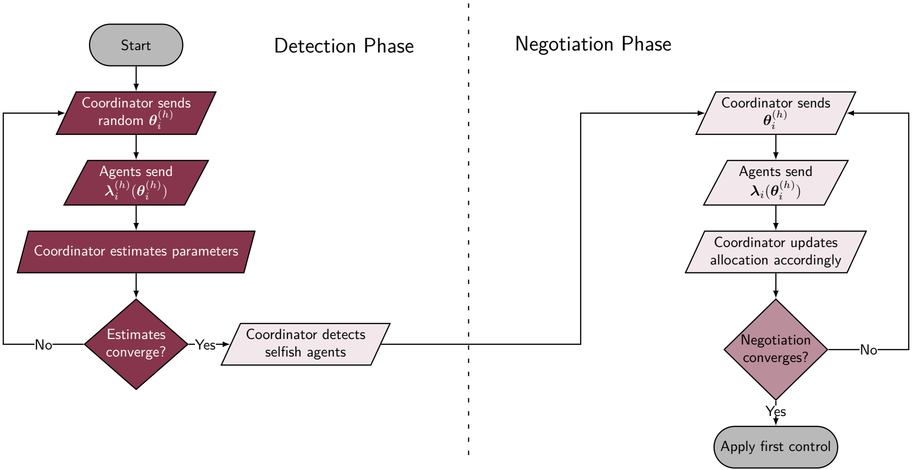
Complete algorithm

RPdMPC-DS



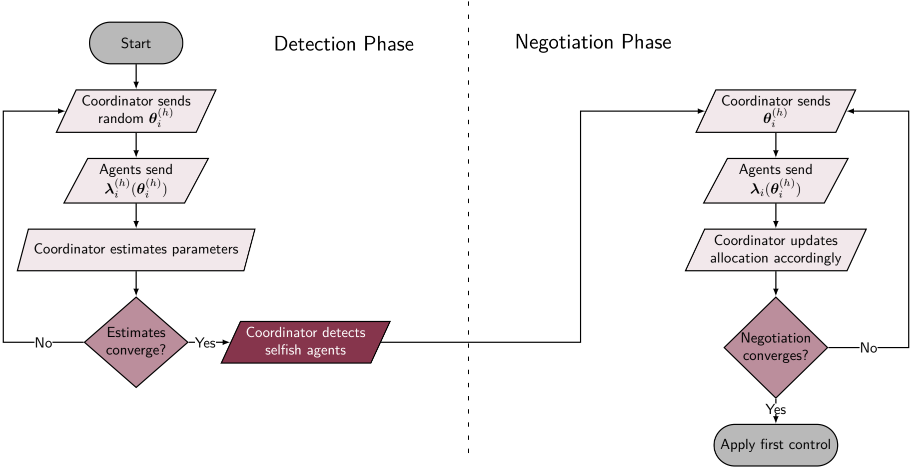
Complete algorithm

RPdMPC-DS



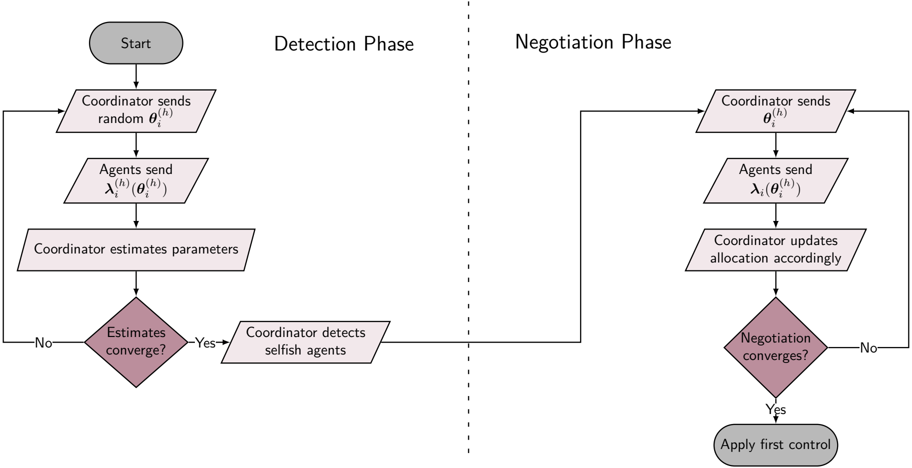
Complete algorithm

RPdMPC-DS



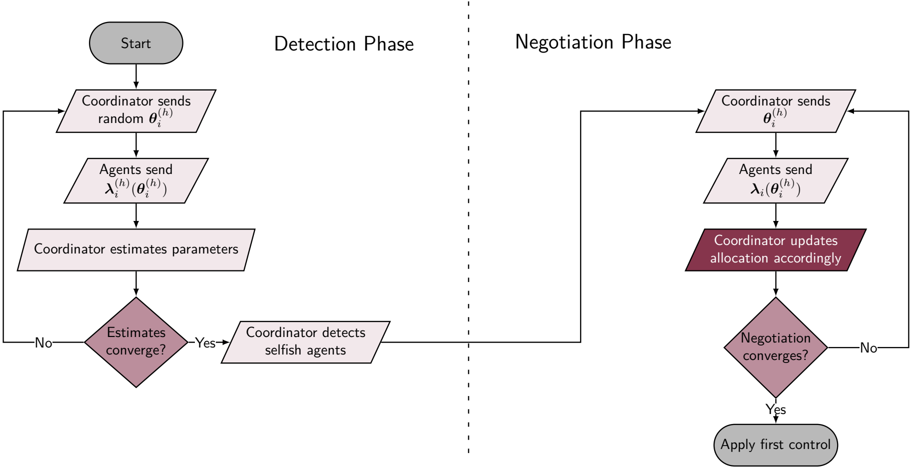
Complete algorithm

RPdMPC-DS

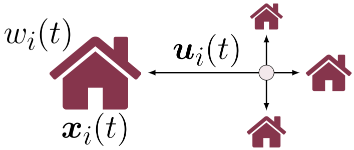


Complete algorithm

RPdMPC-DS

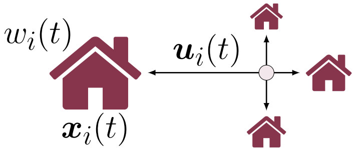


Example



District Heating Network (4 Houses)

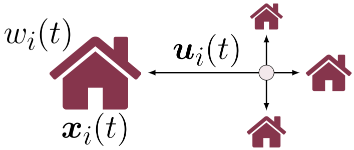
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)

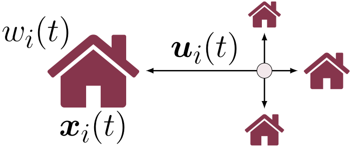
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power

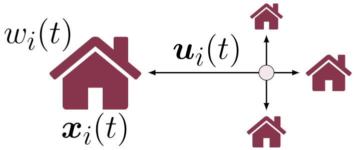
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)

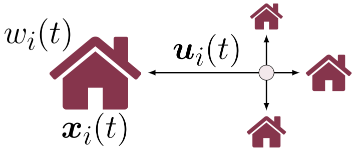
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$ ($T_s = 0.25h \rightarrow k = \{1:20\}$)
- Prediction horizon ($N = 4$)

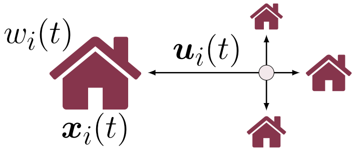
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios

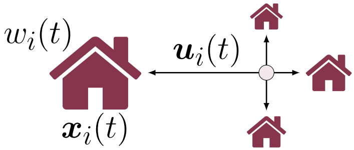
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1:20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
  - N Nominal

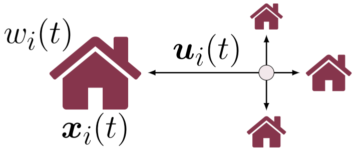
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)

Example

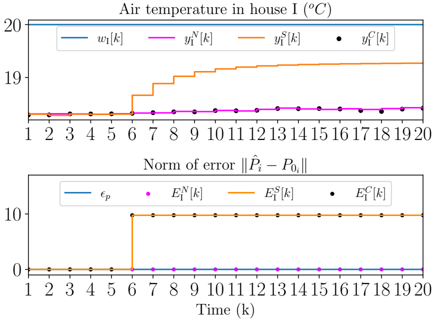


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1:20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

Results

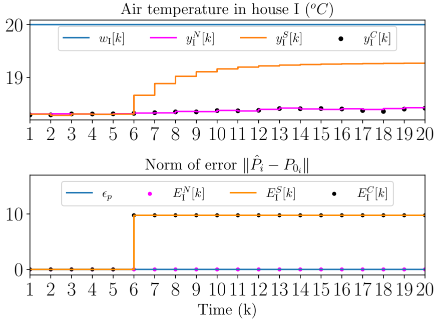
Temporal



Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selflish, C Corrected

Results

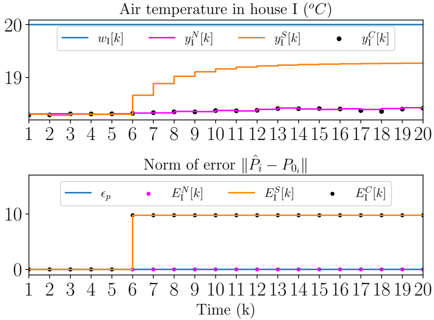
Temporal



Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selflish, C Corrected

Results

Temporal

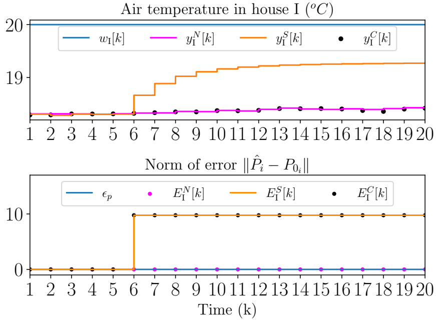


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$

Results

Temporal

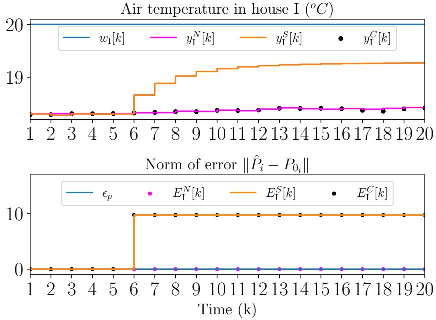


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$
- S Agent increases its comfort

Results

Temporal



Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$

S Agent increases its comfort

C Restablish behavior close to N

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.3	0.5
II	21.67	-0.55
III	17.39	-0.0
IV	17.63	-0.09
Global	3.53	0.02

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.3	0.5
II	21.67	-0.55
III	17.39	-0.0
IV	17.63	-0.09
Global	3.53	0.02

Outline

Relaxing scarcity assumption

Relaxing scarcity assumption

- Systems are not completely deprived

Relaxing scarcity assumption

- Systems are not completely deprived
  - We can't change our constraints to equality ones anymore

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

Relaxing scarcity assumption

- Systems are not completely deprived
  - We can't change our constraints to equality ones anymore
  - Nor use the simpler update equation

$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)})$$

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Now, we may have multiple

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Now, we may have multiple (Piecewise affine function)

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Now, we may have multiple (Piecewise affine function)

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

$$\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$$

Now, we may have multiple (Piecewise affine function)

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

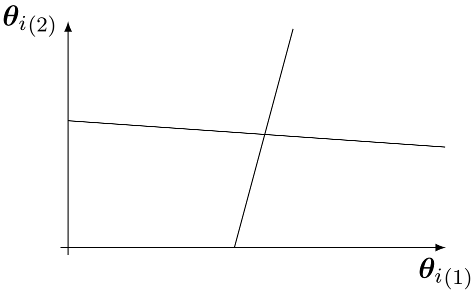
Still the $P_i^{(z)}$ are time independent

Analyzing System

Solution for $\lambda_i[k]$ (Continued)

Analyzing System

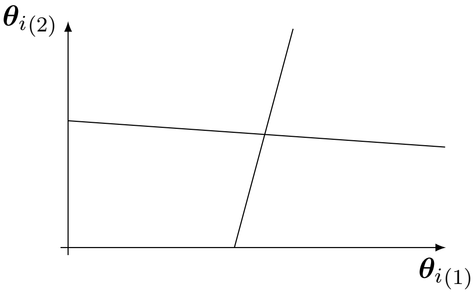
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.

Analyzing System

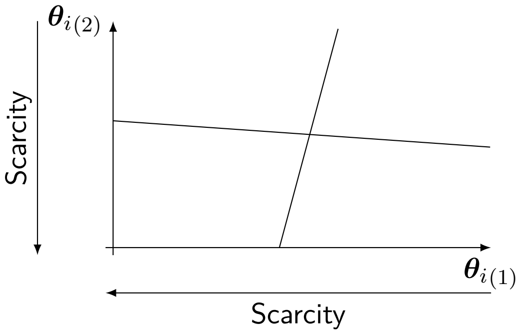
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

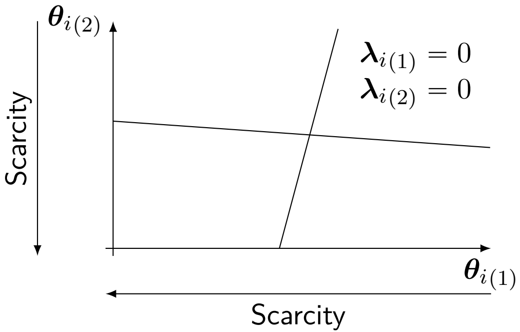
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

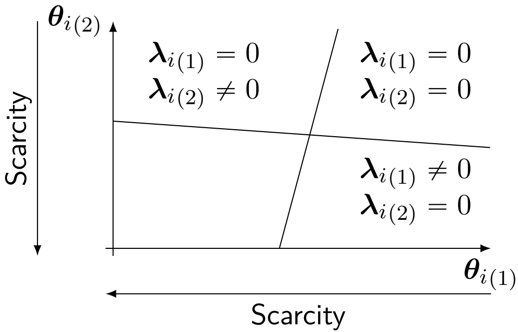
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

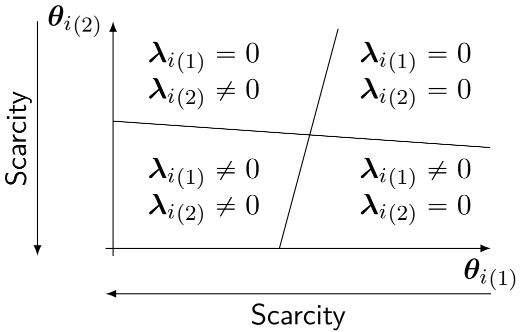
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

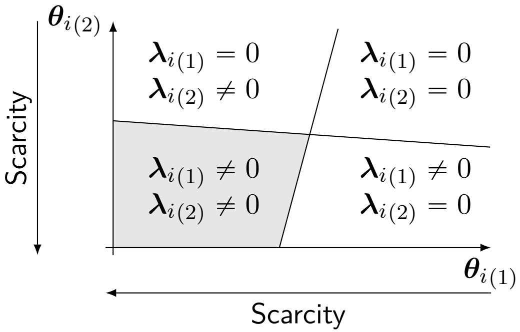
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.
Unknown by the coordinator.

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases} \quad \uparrow \text{Scarcity}$$

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

$\uparrow$ Scarcity

All constraints active $\quad -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k] \quad \rightarrow \quad -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} {\color{red}-P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k]}, & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Scarcity

All constraints active	$-P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k]$	$\rightarrow$	$-P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$
None constraints active	$-P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k]$	$\rightarrow$	$\mathbf{0}$

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Scarcity

All constraints active	$-P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k]$	$\rightarrow$	$-P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$
None constraints active	$-P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k]$	$\rightarrow$	$\mathbf{0}$

Analyzing System

Solution for $\lambda_i[k]$ (Continued) Still?

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

$\uparrow$ Scarcity

All constraints active	$-P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k]$	$\rightarrow$	$-P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$
None constraints active	$-P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k]$	$\rightarrow$	$\mathbf{0}$

Assumptions

The region $\mathcal{R}^0_{\boldsymbol{\lambda}_i} \neq \varnothing$ *and we known a point* $\overset{\varnothing}{\boldsymbol{\theta}}_i \in \mathcal{R}^0_{\boldsymbol{\lambda}_i}$

Analyzing System

Under attack!

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Parameters are modified.

$$\tilde{\boldsymbol{\lambda}}_i[k] = \begin{cases} -\widetilde{P}_i^{(0)}\boldsymbol{\theta}_i[k] - \widetilde{\boldsymbol{s}}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\widetilde{P}_i^{(Z)}\boldsymbol{\theta}_i[k] - \widetilde{\boldsymbol{s}}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Parameters are modified. But not the regions' limits

$$\tilde{\boldsymbol{\lambda}}_i[k] = \begin{cases} -\widetilde{P}_i^{(0)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\widetilde{P}_i^{(Z)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Parameters are modified. But not the regions' limits

$$\tilde{\boldsymbol{\lambda}}_i[k] = \begin{cases} -\widetilde{P}_i^{(0)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\widetilde{P}_i^{(Z)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

- If we can estimate $\widetilde{P}_i^{(0)}$ we can use same strategy than before

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Parameters are modified. But not the regions' limits

$$\tilde{\boldsymbol{\lambda}}_i[k] = \begin{cases} -\widetilde{P}_i^{(0)}\boldsymbol{\theta}_i[k] - \widetilde{\boldsymbol{s}}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\widetilde{P}_i^{(Z)}\boldsymbol{\theta}_i[k] - \widetilde{\boldsymbol{s}}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

- If we can estimate $\widetilde{P}_i^{(0)}$ we can use same strategy than before
- Problem: We don't know in which region $\boldsymbol{\theta}_i$ is

Analyzing System

Under attack!

$$\tilde{\boldsymbol{\lambda}}_i[k] = T_i[k]\boldsymbol{\lambda}_k$$

Parameters are modified. But not the regions' limits

$$\tilde{\boldsymbol{\lambda}}_i[k] = \begin{cases} -\widetilde{P}_i^{(0)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0 \\ \vdots & \vdots \\ -\widetilde{P}_i^{(Z)}\boldsymbol{\theta}_i[k] - \tilde{\boldsymbol{s}}_i^{(Z)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^Z_{\boldsymbol{\lambda}_i} \end{cases}$$

- If we can estimate $\widetilde{P}_i^{(0)}$ we can use same strategy than before
- Problem: We don't know in which region $\boldsymbol{\theta}_i$ is
- Solution: Let's force it using Artificial Scarcity

Artificial Scarcity

What you thought was way too much is not enough

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints

Artificial Scarcity

What you thought was way too much is not enough

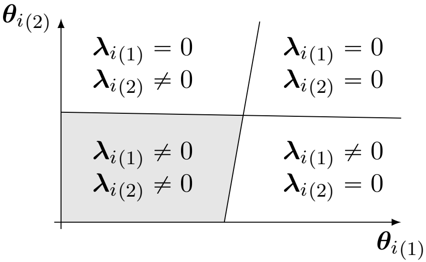
- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]

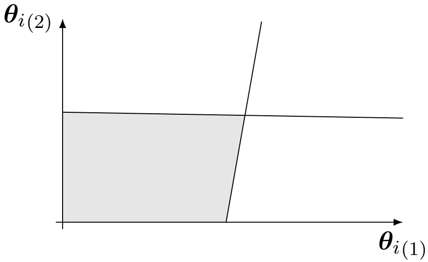


[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]

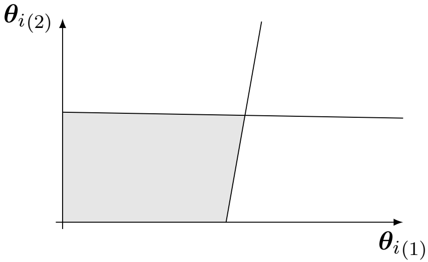


[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



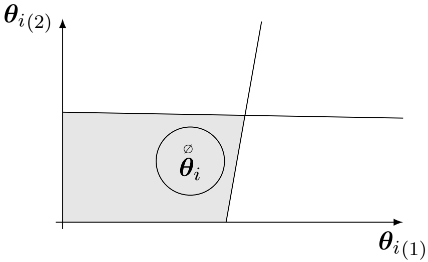
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



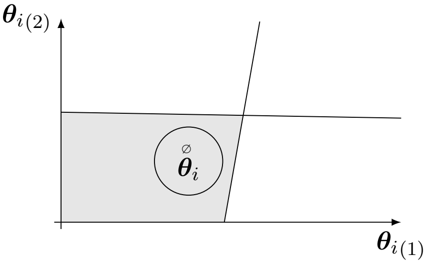
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



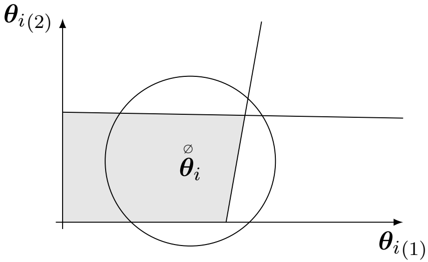
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



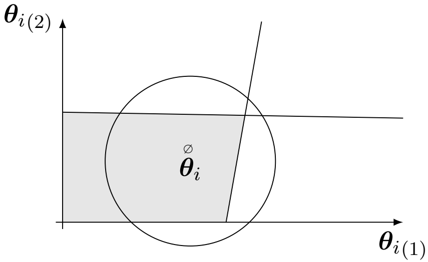
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



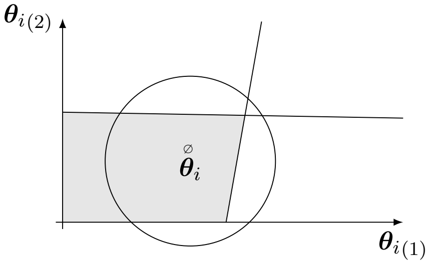
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?
  - Unfortunately we don't.

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



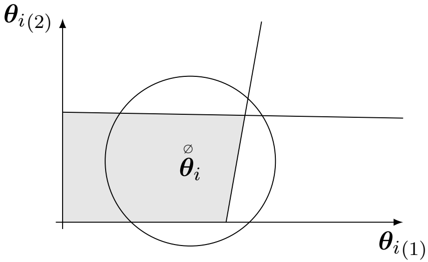
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?
  - Unfortunately we don't.
- How to estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$ nonetheless?

[17] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

What you thought was way too much is not enough

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[17]



- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?
  - Unfortunately we don't.
- How to estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$ nonetheless?
  - Expectation Maximization

[17] If we have local constraints, we suppose this point respects them.

Expectation Maximization

Expectation Maximization

- Iterative method to estimate parameters of multimodal models

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities
- At the end we have

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\tilde{P}}_i^{(z)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\tilde{P}}_i^{(z)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

[18] Such as our PWA function after using some tricks

Expectation Maximization

- Iterative method to estimate parameters of multimodal models[18]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(z)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(z)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[18] Such as our PWA function after using some tricks

Detection and Mitigation

Same same, but different

Detection and Mitigation

Same same, but different

Assumption

We estimate nominal $\bar{P}_i^{(0)}$ *from attack free negotiation*

Detection and Mitigation

Same same, but different

Assumption

We estimate nominal $\bar{P}_i^{(0)}$ from attack free negotiation

- Detection

$$\left\| \widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F \geqslant \epsilon_{P_i^{(0)}}$$

Detection and Mitigation

Same same, but different

Assumption

We estimate nominal $\bar{P}_i^{(0)}$ from attack free negotiation

- Detection

$$\left\| \widehat{\tilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F \geqslant \epsilon_{P_i^{(0)}}$$

- Mitigation

$$\widehat{T_i[k]^{-1}} = \bar{P}_i^{(0)} \widehat{\tilde{P}}_i^{(0)}[k]^{-1}.$$

Detection and Mitigation

Same same, but different

Assumption

We estimate nominal $\bar{P}_i^{(0)}$ from attack free negotiation

- Detection

$$\left\| \widehat{\tilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F \geqslant \epsilon_{P_i^{(0)}}$$

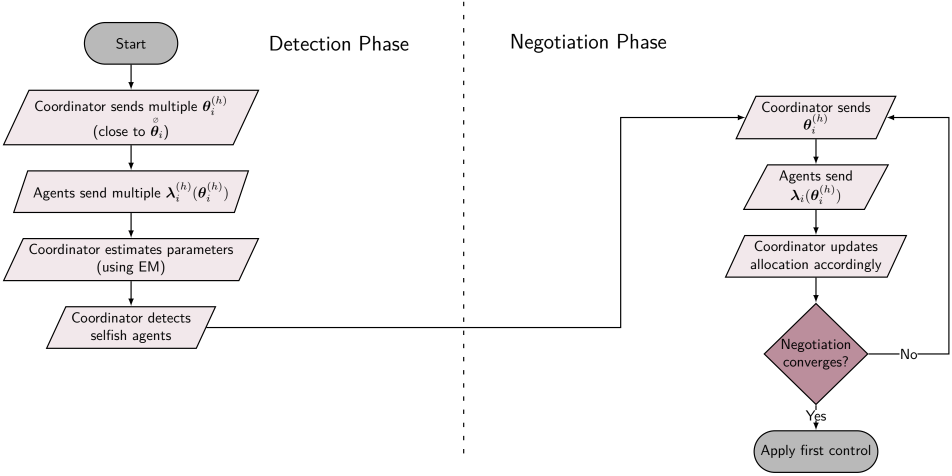
- Mitigation

$$\widehat{T_i[k]^{-1}} = \bar{P}_i^{(0)} \, \widehat{\tilde{P}}_i^{(0)}[k]^{-1}.$$

$$\overset{\text{rec}}{\boldsymbol{\lambda}_i} = \widehat{T_i[k]^{-1}} \tilde{\boldsymbol{\lambda}}_i.$$

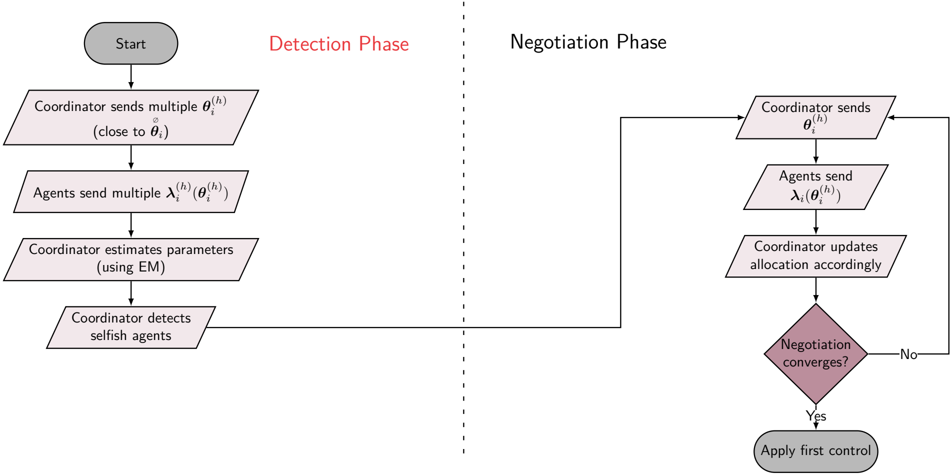
Complete algorithm

RPdMPC-AS



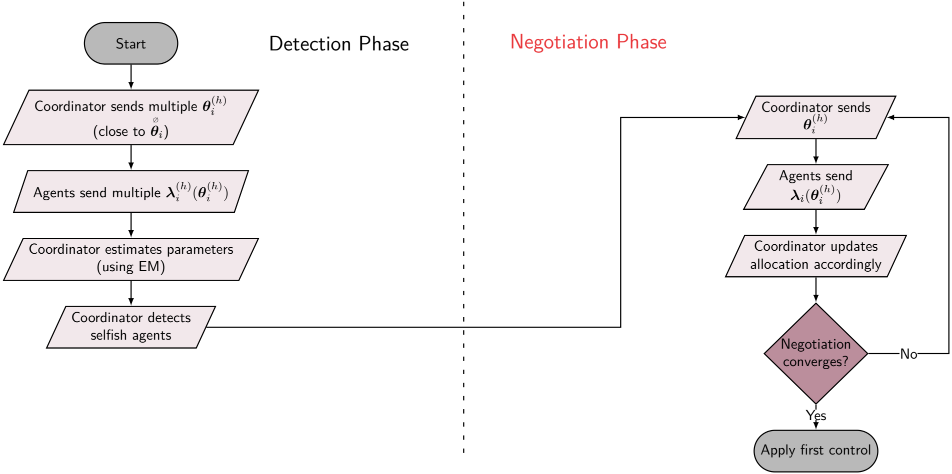
Complete algorithm

RPdMPC-AS



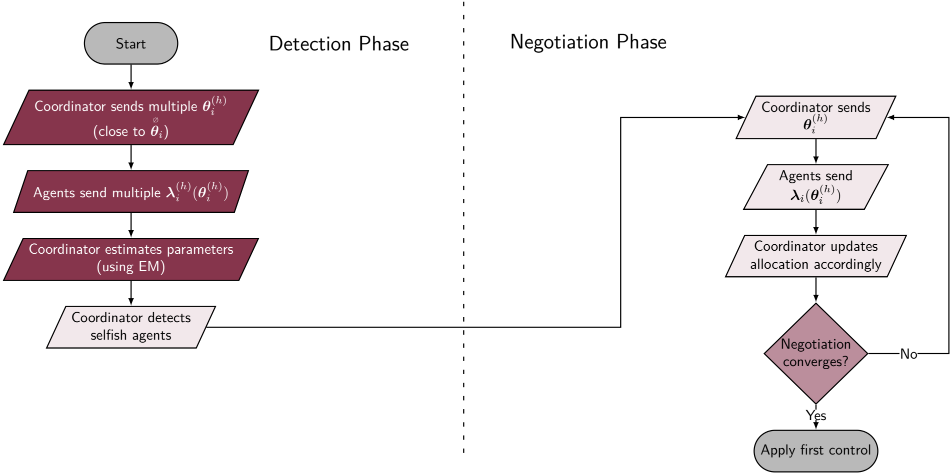
Complete algorithm

RPdMPC-AS



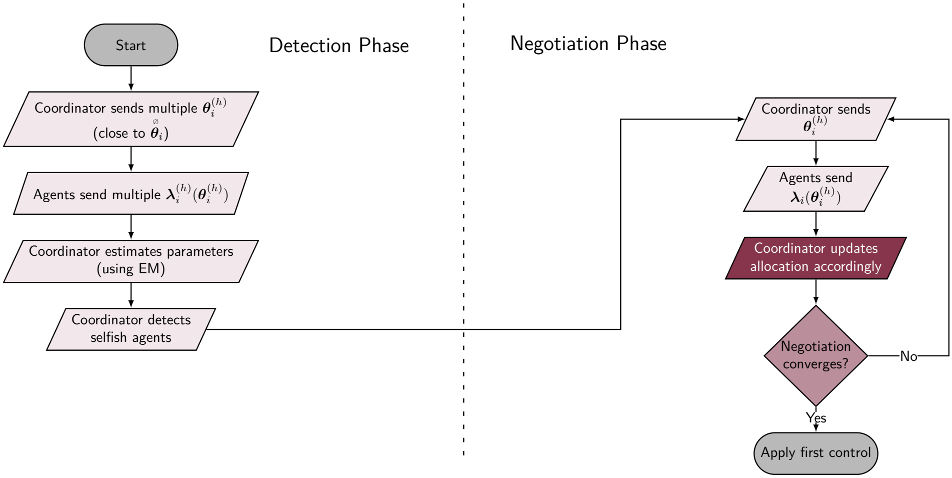
Complete algorithm

RPdMPC-AS

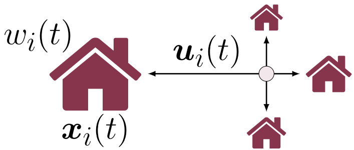


Complete algorithm

RPdMPC-AS



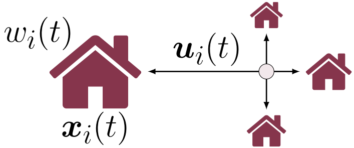
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- Not enough power
- Period of 5h ($T_s = 0.25h \rightarrow k = \{1 : 20\}$)
- Prediction horizon ($N = 4$)
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-AS)

Example

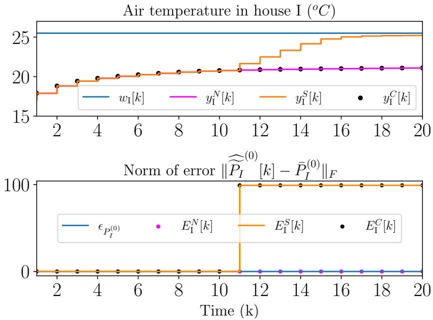


District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- ~~Not enough power~~ (Change $(\boldsymbol{x}_0, \boldsymbol{w}_0)$)
- Period of $5\mathrm{h}$ $(T_s = 0.25h \rightarrow k = \{1:20\})$
- Prediction horizon $(N = 4)$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-AS)

Results

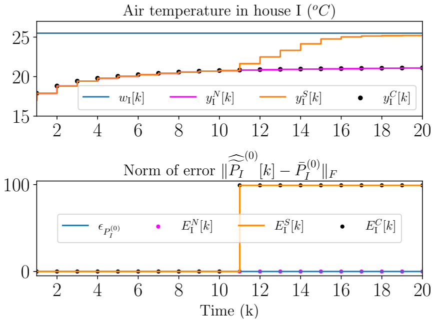
Temporal



Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish C Corrected

Results

Temporal



Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish C Corrected

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.49	$-4.12e-05$
II	35.81	$1.74e-05$
III	29.22	$2.14e-05$
IV	37.54	$1.73e-05$
Global	10.69	$-6e-07$

Too good to be true!

It's a kind of magic!

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization
    - No guarantees of achieving global optimal

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization
    - No guarantees of achieving global optimal
- Some "solutions":

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization
    - No guarantees of achieving global optimal
- Some "solutions":
  - Force some parameters to converge faster (case dependant)

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization
    - No guarantees of achieving global optimal
- Some "solutions":
  - Force some parameters to converge faster (case dependant)
  - Run multiple times with different initialization and pick best

Too good to be true!

~~It's a kind of magic!~~

- Unfortunately EM is not magic
  - Slow convergence
  - Dependency on initialization
    - No guarantees of achieving global optimal
- Some "solutions":
  - Force some parameters to converge faster (case dependant)
  - Run multiple times with different initialization and pick best
  - Associate with other methods of the same family

For Further Reading I

- Åström, K.J. and B. Wittenmark. Adaptive Control. Addison-Wesley series in electrical and computer engineering: Control engineering. Addison-Wesley, 1989. ISBN: 9780201097207. DOI: 10.1007/978-3-662-08546-2_24.
- Maestre, José M, Rudy R Negenborn, et al. Distributed Model Predictive Control made easy. Vol. 69. Springer, 2014. ISBN: 978-94-007-7005-8.
- Nogueira, Rafael Accácio. "Security of DMPC under False Data Injection". 2022CSUP0006. PhD thesis. CentraleSupélec, 2022. URL: http://www.theses.fr/2022CSUP0006.