Security of distributed Model Predictive Control under False Data Injection

Rafael Accácio NOGUEIRA

2022-12-12

IETR

https://bit.ly/3g3S6X4

CentraleSupélec

Context

“Necessity is the mother of invention”

Context

“Necessity is the mother of invention”

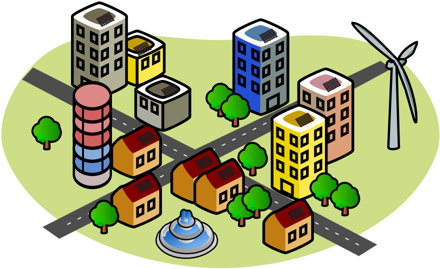
Context

"Necessity is the mother of invention"

- Electricity Distribution System

Context

" Necessity is the mother of invention "



- Electricity Distribution System
- Heat distribution
- Water distribution

Context

“Necessity is the mother of invention”

- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

Context

"Necessity is the mother of invention"

- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management

  (include your problem here)

Context

“Necessity is the mother of invention”

- Multiple systems interacting

Context

"Necessity is the mother of invention"

- Multiple systems interacting
- Coupled by constraints

CentraleSupélec

Context

❝ Necessity is the mother of invention ❞

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort

Context

"Necessity is the mother of invention"

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort
- Optimization objectives

Context

“Necessity is the mother of invention”

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort
- Optimization objectives
  - Minimize energy consumption

CentraleSupélec

Context

“Necessity is the mother of invention”

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction

Context

"Necessity is the mother of invention"

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory

CentraleSupélec

Context

“Necessity is the mother of invention”

- Multiple systems interacting
- Coupled by constraints
  - Technical/ Comfort
- Optimization objectives
  - Minimize energy consumption
  - Maximize user satisfaction
  - Follow a trajectory
- Solution → MPC

CentraleSupélec

Model-based Predictive Control

Model-based Predictive Control

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Find best control sequence using predictions based on a model.

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem

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

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

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

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

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)

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

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize

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

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize
  - System's Model (states and inputs)

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

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize
  - System's Model (states and inputs)

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

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize
  - System's Model (states and inputs)

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

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize
  - System's Model (states and inputs)
  - Other constraints to respect

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

CentraleSupélec

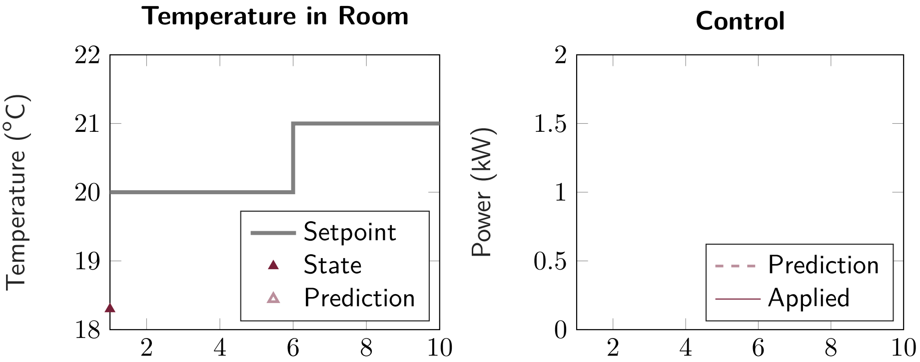
Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

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

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

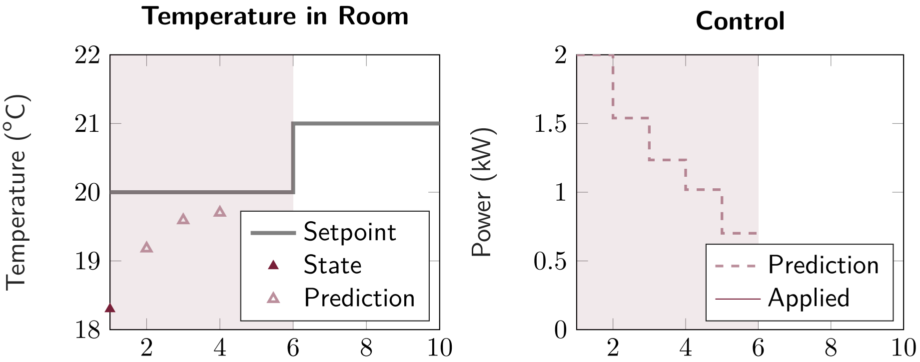
In a nutshell



Model Predictive Control

In a nutshell

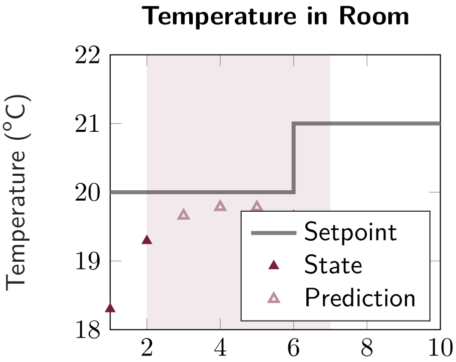
Find optimal control sequence

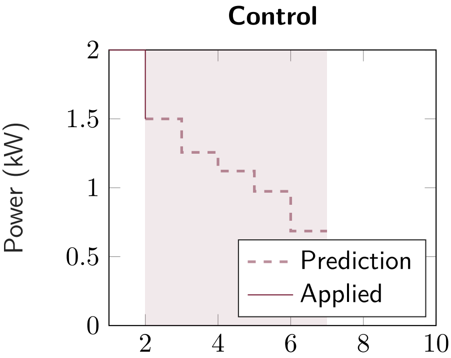


Model Predictive Control

In a nutshell

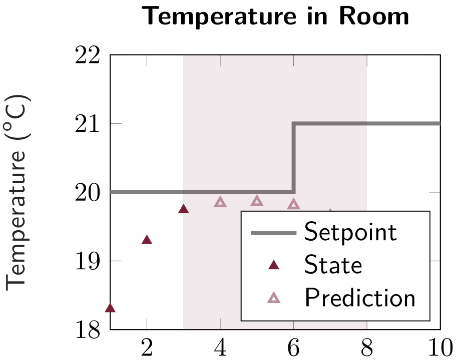
Find optimal control sequence, apply first element

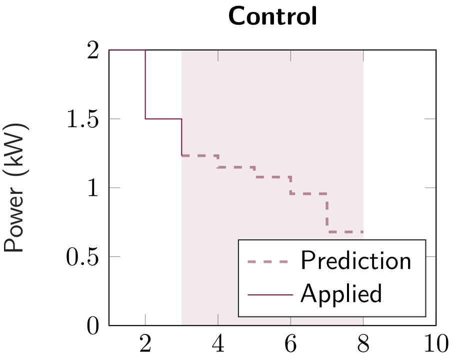




In a nutshell

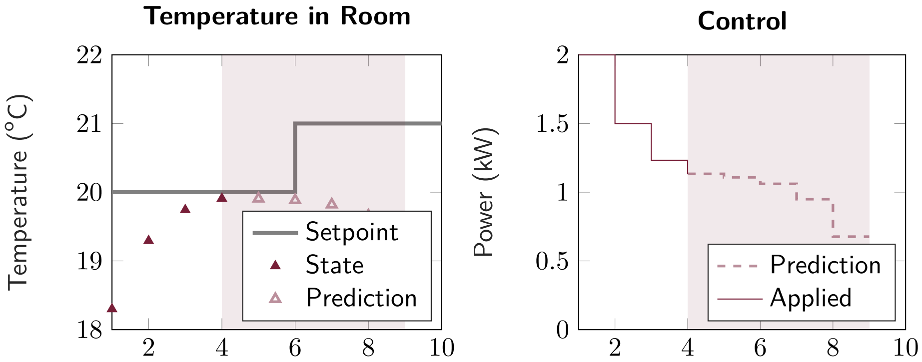
Find optimal control sequence, apply first element, rinse repeat





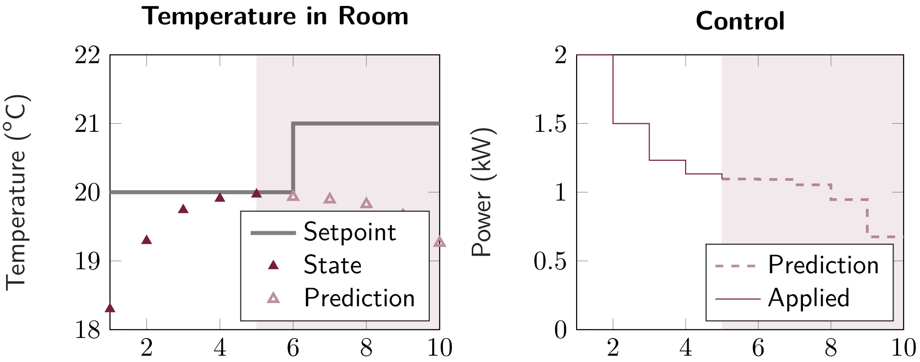
In a nutshell

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



In a nutshell

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



Model Predictive Control

Nothing is perfect

Nothing is perfect

- Problems

Model Predictive Control

Nothing is perfect

- Problems
  - Complexity of calculation

Model Predictive Control

Nothing is perfect

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

Model Predictive Control

Nothing is perfect

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

CentraleSupélec

Model Predictive Control

Nothing is perfect

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

CentraleSupélec

Model Predictive Control

Nothing is perfect

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

Model Predictive Control

Nothing is perfect

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

CentraleSupélec

Model Predictive Control

Nothing is perfect

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

Distributed Model Predictive Control

It is about communication

- We break the MPC into multiple
- Make agents communicate.

CentraleSupélec

Distributed Model Predictive Control

It is about communication

- We break the MPC into multiple
- Make agents communicate. But how?

CentraleSupélec

Distributed Model Predictive Control

It is about communication

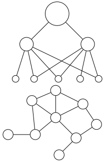
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from

CentraleSupélec

Distributed Model Predictive Control

It is about communication

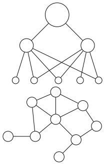
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical

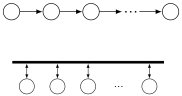


Distributed Model Predictive Control

It is about communication

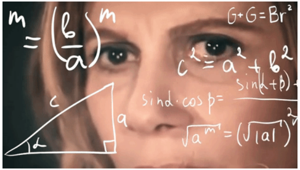
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Sequential/Parallel

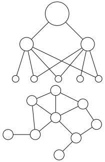


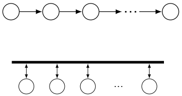


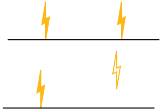
It is about communication

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







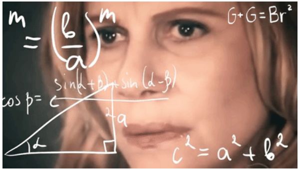


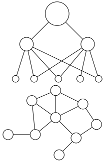
CentraleSupélec

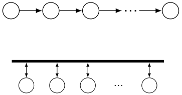
Distributed Model Predictive Control

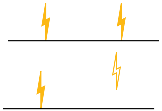
It is about communication

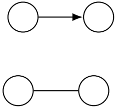
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional







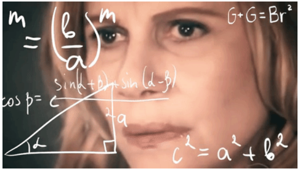


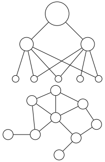


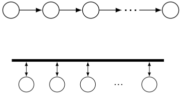
Distributed Model Predictive Control

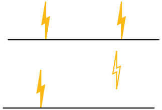
It is about communication

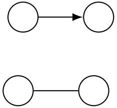
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...











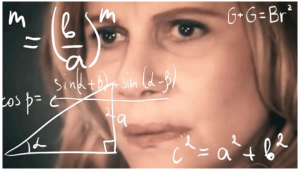
[1] Distributed Model Predictive Control made easy

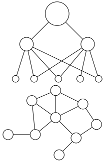
CentraleSupélec

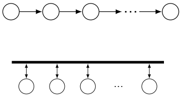
Distributed Model Predictive Control

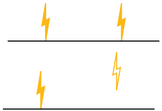
It is about communication

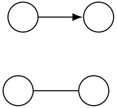
- We break the MPC into multiple
- Make agents communicate. But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...











[1] Distributed Model Predictive Control made easy

CentraleSupélec

Distributed Model Predictive Control

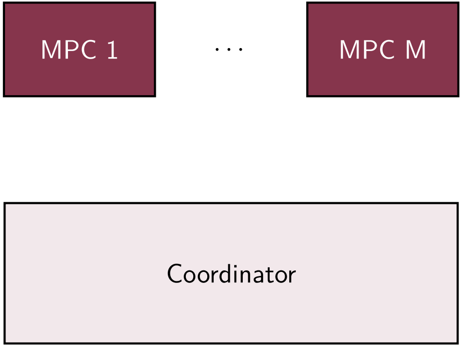
Communication Frameworks

MPC

Communication Frameworks

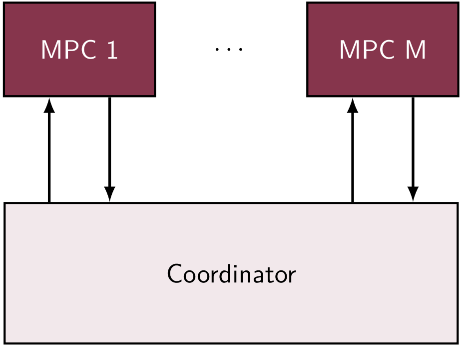
CentraleSupélec

Communication Frameworks



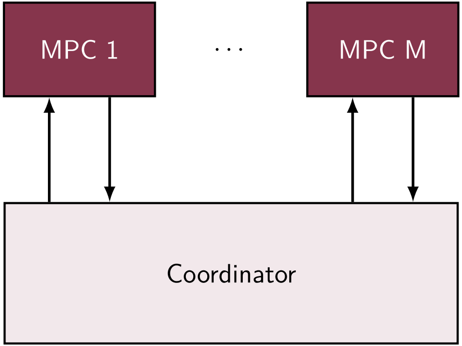
- Coordinator → Hierarchical

Communication Frameworks



- Coordinator → Hierarchical
- Bidirectional

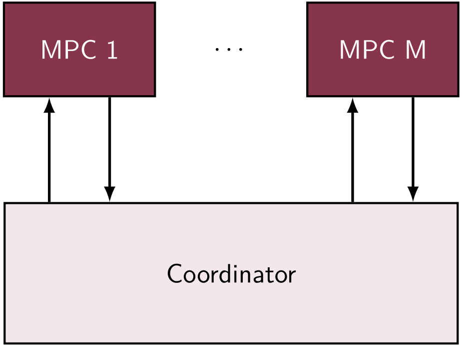
Communication Frameworks



- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous

Distributed Model Predictive Control

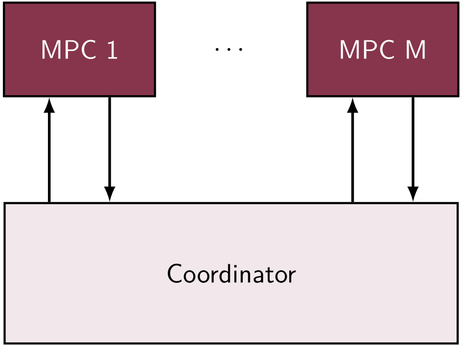
Communication Frameworks



- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous

- Agents solve local problems

Communication Frameworks

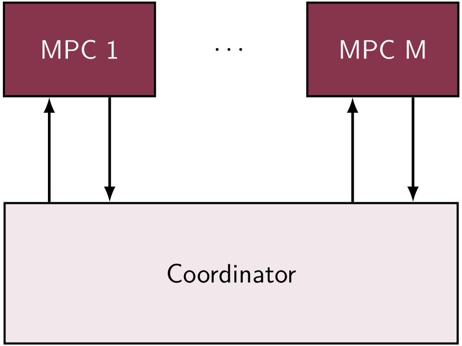


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous

- Agents solve local problems
- Variables are updated

Distributed Model Predictive Control

Communication Frameworks



- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous

- Agents solve local problems
- Variables are updated

} Until Convergence

CentraleSupélec

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?

CentraleSupélec

Distributed Model Predictive Control

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

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

Distributed Model Predictive Control

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

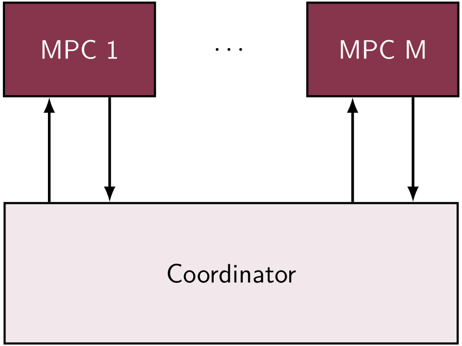
- How can an agent attack?
- What are the consequences of an attack?
- Can we mitigate the effects?

Let's have a preview!

CentraleSupélec

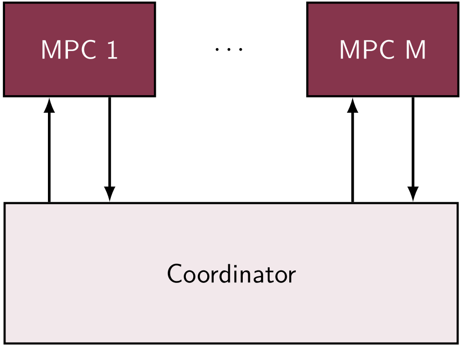
How can a non-cooperative agent attack?

Literature



How can a non-cooperative agent attack?

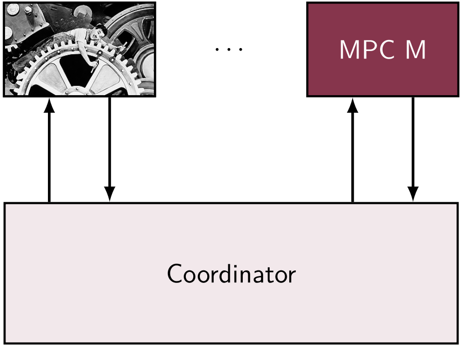
Literature



- [Vel+17a; CMI18] present attacks

How can a non-cooperative agent attack?

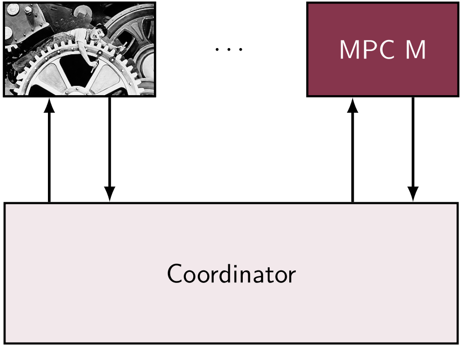
Literature



- [Vel+17a; CMI18] present attacks

How can a non-cooperative agent attack?

Literature

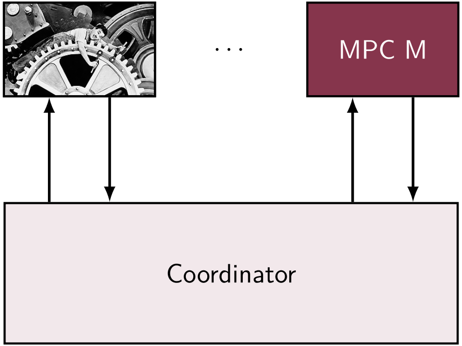


- [Vel+17a; CMI18] present attacks
  - Fake objective function
  - Fake constraints
  - Use different control

CentraleSupélec

How can a non-cooperative agent attack?

Literature

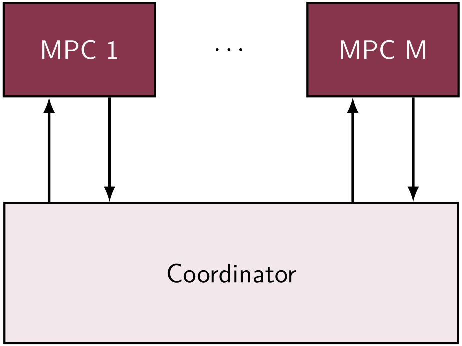


- [Vel+17a; CMI18] present attacks
  - Fake objective function
  - Fake constraints
  - Use different control

Deception Attacks

How can a non-cooperative agent attack?

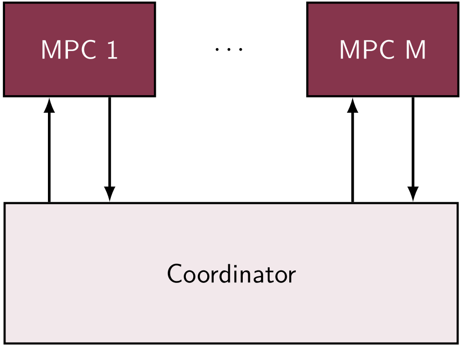
Our approach



- We are in coordinator's shoes

How can a non-cooperative agent attack?

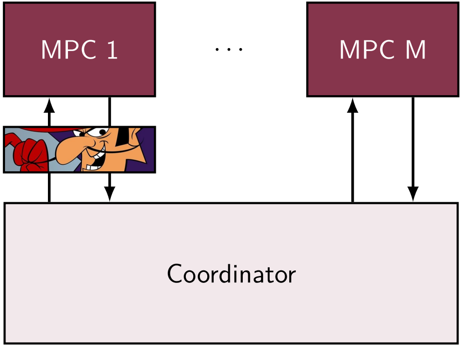
Our approach



- We are in coordinator's shoes
- What matters is the interface

How can a non-cooperative agent attack?

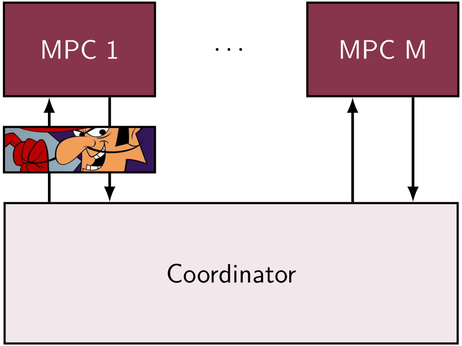
Our approach



- We are in coordinator's shoes
- What matters is the interface
    - Attacker changes communication

How can a non-cooperative agent attack?

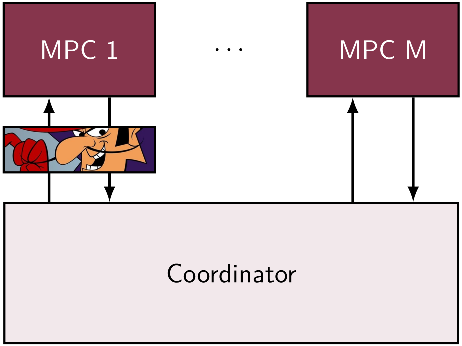
Our approach



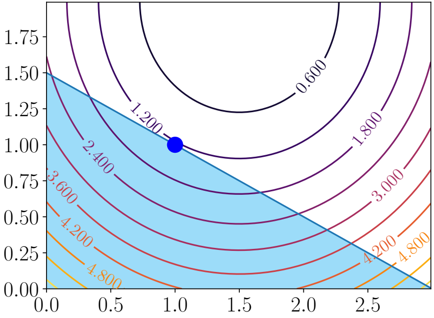
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

How can a non-cooperative agent attack?

Our approach



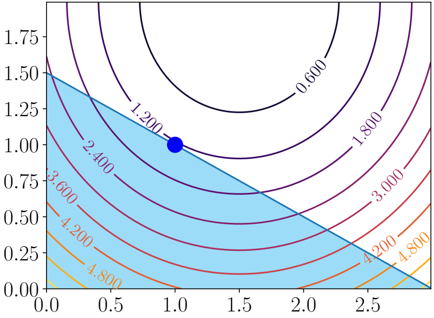
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection



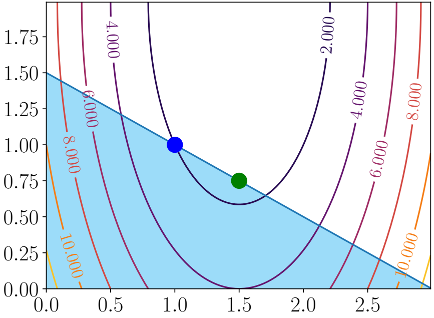
Original minimum.

Consequence of an attack

- Attack modifies optimization problem

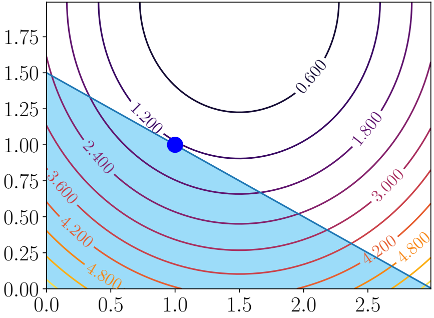


Original minimum.

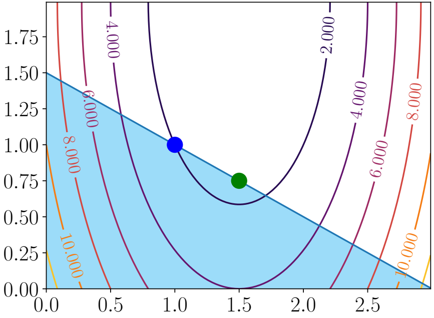


Minimum after attack.

- Attack modifies optimization problem
  - Optimum value is shifted



Original minimum.



Minimum after attack.

CentraleSupélec

Mitigating the effects

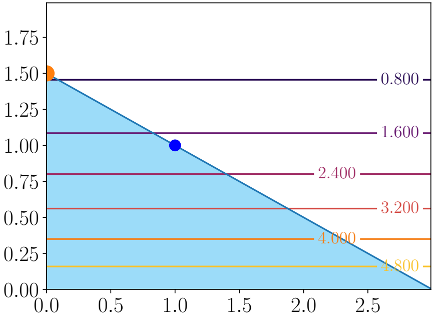
Mitigating the effects

- We can recover by

CentraleSupélec

Mitigating the effects

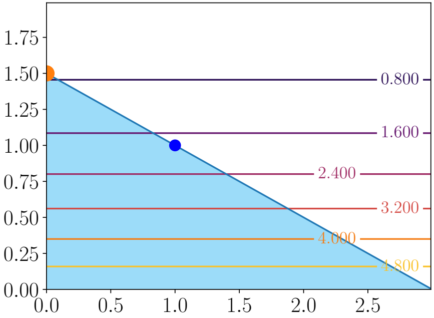
- We can recover by
  - Ignoring attacker



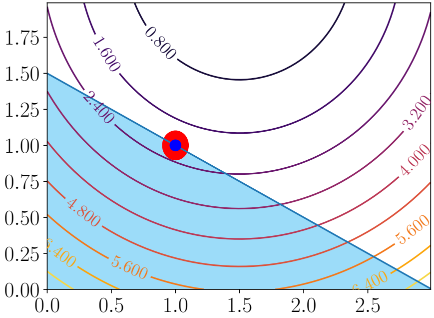
Ignore attacker.

Mitigating the effects

- We can recover by
  - Ignoring attacker
  - Recover original behavior (at least trying)



Ignore attacker.

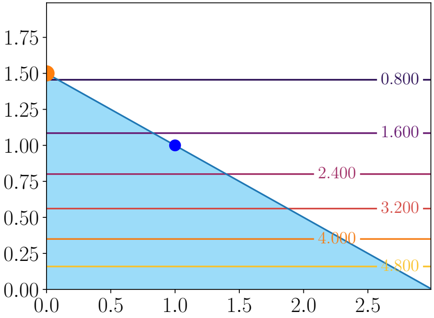


Recover original behavior.

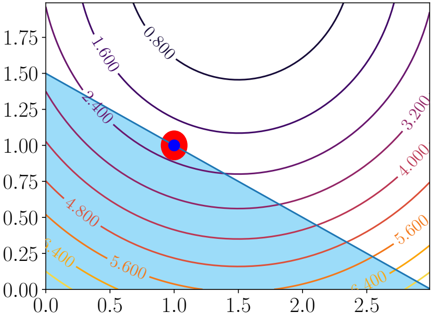
CentraleSupélec

Mitigating the effects

- We can recover by
  - Ignoring attacker
  - Recover original behavior (at least trying)



Ignore attacker.



Recover original behavior.

Classification of mitigation techniques

Passive (Robust)

Active (Resilient)

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

- 2 modes

CentraleSupélec

Classification of mitigation techniques

Passive (Robust)

- 1 mode

Active (Resilient)

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

CentraleSupélec

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

CentraleSupélec

State of art

Security dMPC

	Decomposition	Resilient/Robust
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)
[CMI18]	Jacobi-Gauß	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient

State of art

Security dMPC

	Decomposition	Resilient/Robust
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)
[CMI18]	Jacobi-Gauß	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient
Our	Primal	Resilient

CentraleSupélec

State of art

Security dMPC

	Decomposition	Resilient/Robust
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)
[CMI18]	Jacobi-Gauß	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient
Our	Primal	Resilient

CentraleSupélec

State of art

Security dMPC

	Decomposition	Resilient/Robust
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)
[CMI18]	Jacobi-Gauß	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient
Our	Primal	Resilient

State of art

Security dMPC

	Decomposition	Resilient/Robust	Detection	Mitigation
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)	NA	NA
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)	NA	NA
[CMI18]	Jacobi-Gauß	–	–	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Resilient	Active Analyt./Learn.	Data reconstruction

CentraleSupélec

State of art

Security dMPC

	Decomposition	Resilient/Robust	Detection	Mitigation
[Vel+17a] [Mae+21]	Dual	Robust (Scenario)	NA	NA
[Vel+17b] [Vel+18]	Dual	Robust (f-robust)	NA	NA
[CMI18]	Jacobi-Gauß	–	–	–
[Ana+18] [Ana+19] [Ana+20]	Dual	Resilient	Analyt./Learn.	Disconnect (Robustness)
Our	Primal	Resilient	Active Analyt./Learn.	Data reconstruction

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition

2. Resilient Primal Decomposition-based dMPC for deprived systems

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity
4. Conclusion

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity
4. Conclusion

- 1 and 2 yielded [NBG21] (SysTol'21)

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition

2. Resilient Primal Decomposition-based dMPC for deprived systems

3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity

4. Conclusion

- 1 and 2 yielded [NBG21] (SysTol'21)
- 3 yielded [Nog+22] (NecSys'22)

CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
2. Resilient Primal Decomposition-based dMPC for deprived systems
3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity
4. Conclusion

- 1 and 2 yielded [NBG21] (SysTol'21)
- 3 yielded [Nog+22] (NecSys'22)

Simon Leglaive
AIMAC Team

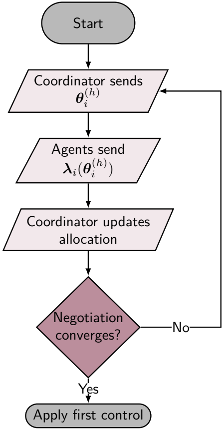
CentraleSupélec

Outline

1. Vulnerabilities in distributed MPC based on Primal Decomposition
   - What is the Primal Decomposition?
   - How can an agent attack?
   - Consequences

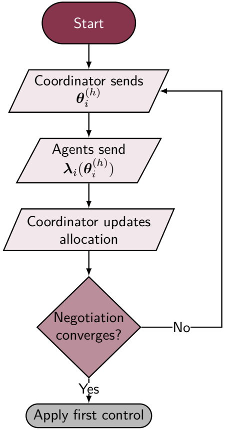
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



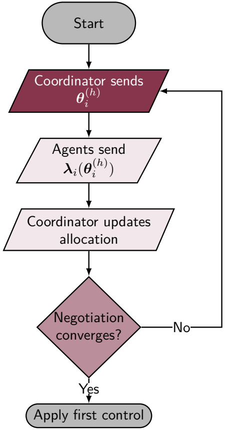
Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Primal Decomposition

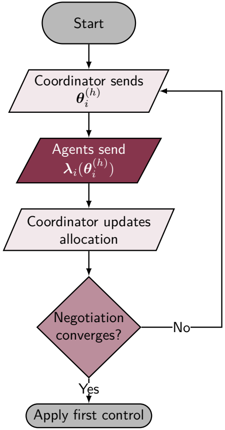
or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

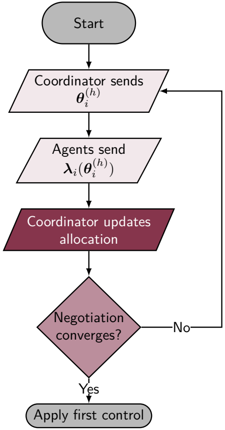


Allocation θ_i

Dissatisfaction λ_i

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



Allocation $\boldsymbol{\theta}_i$

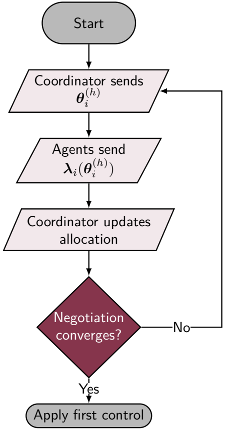
Dissatisfaction $\boldsymbol{\lambda}_i$

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation



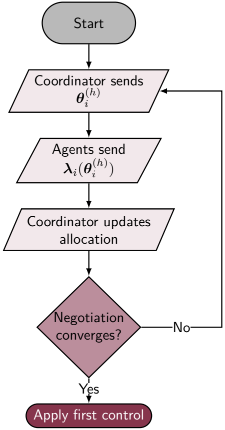
Allocation $\boldsymbol{\theta}_i$

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

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

CentraleSupélec

Primal Decomposition

In detail

- Objective is sum of local ones

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

CentraleSupélec

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

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

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

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

CentraleSupélec

Primal Decomposition

In detail

- Objective is sum of local ones
- Constraints couple variables

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

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

CentraleSupélec

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 $\boldsymbol{\lambda}_i$

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

CentraleSupélec

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

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

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

CentraleSupélec

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 (respecting global constraint)

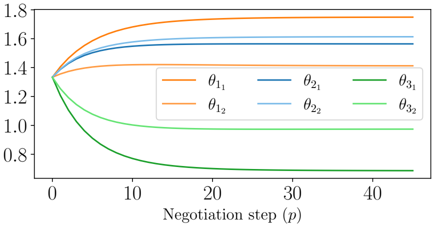
$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

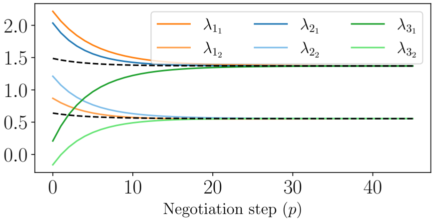
$$\boldsymbol{\theta}[k]^{(p+1)} = \mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

Example

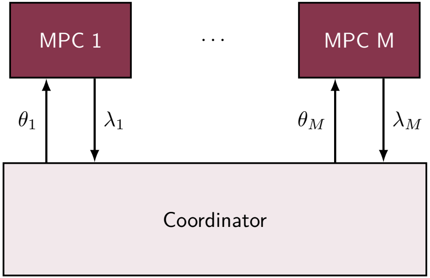
Until everybody is equally dissatisfied





How can a non-cooperative agent attack?

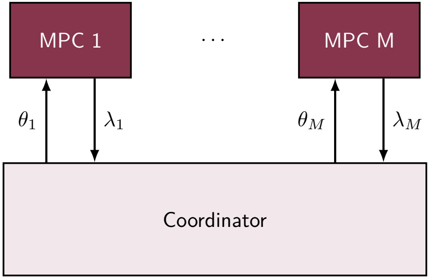
Our approach



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

How can a non-cooperative agent attack?

Our approach

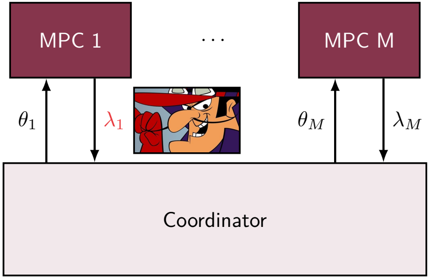


- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters

CentraleSupélec

How can a non-cooperative agent attack?

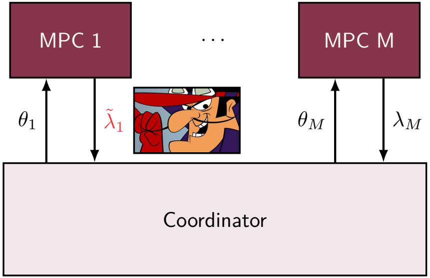
Our approach



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters
- Malicious agent modifies $\boldsymbol{\lambda}_i$

How can a non-cooperative agent attack?

Our approach



- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters
- Malicious agent modifies $\boldsymbol{\lambda}_i$

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

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire

How does an agent lie?

Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ means dissatisfaction

How does an agent lie?

Liar, Liar, Pants of fire

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

How does an agent lie?

Liar, Liar, Pants of fire

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

Assumptions

How does an agent lie?

Liar, Liar, Pants of fire

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

Assumptions

- *Same attack during negotiation*

How does an agent lie?

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*

How does an agent lie?

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*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$

How does an agent lie?

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*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i$

CentraleSupélec

How does an agent lie?

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

CentraleSupélec

How does an agent lie?

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

CentraleSupélec

How does an agent lie?

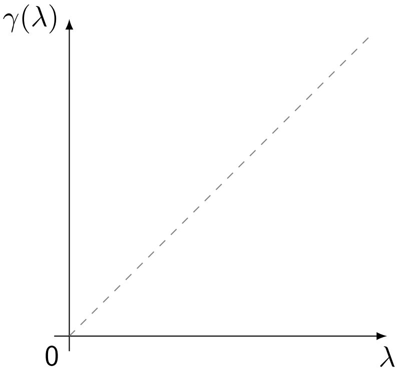
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ 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 $\rightarrow \exists T_i[k]^{-1}$



CentraleSupélec

How does an agent lie?

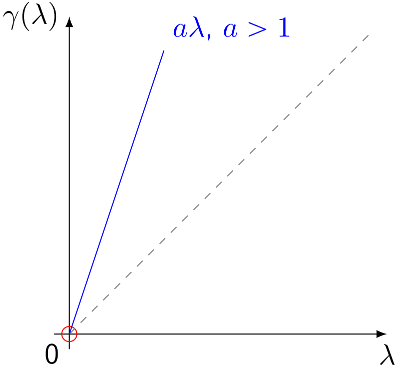
Liar, Liar, Pants of fire

- $\lambda \geqslant 0$ 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 $\rightarrow \exists T_i[k]^{-1}$



CentraleSupélec

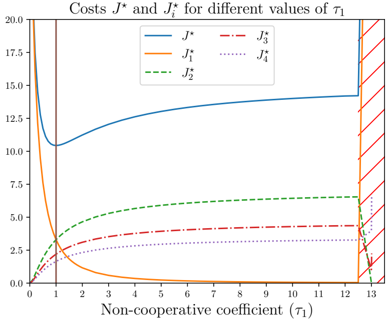
Example

Example

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

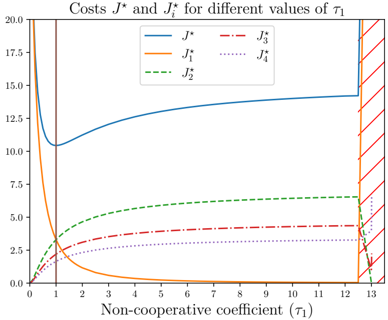
Example



4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

Example



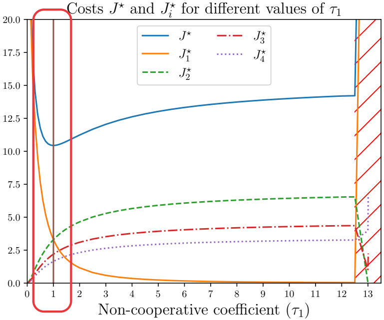
4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

- We can observe 3 things

CentraleSupélec

Example

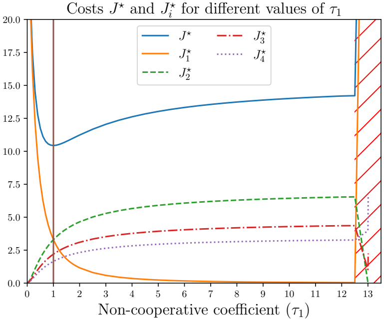


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

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

Example

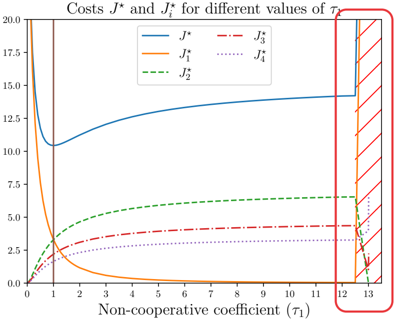


4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

- 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 $\tilde{\boldsymbol{\lambda}}_1 = \gamma_1(\boldsymbol{\lambda}_1) = \tau_1 I \boldsymbol{\lambda}_1$

- 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

CentraleSupélec

- But can we mitigate these effects?

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

Outline

2. Resilient Primal Decomposition-based dMPC for deprived systems
   - Analyzing deprived systems
   - Building an algorithm
   - Applying mechanism

CentraleSupélec

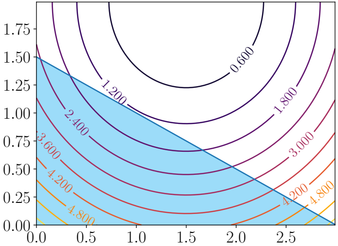
What are deprived systems?

What are deprived systems?

Systems whose optimal solution has all constraints active

What are deprived systems?

Systems whose optimal solution has all constraints active



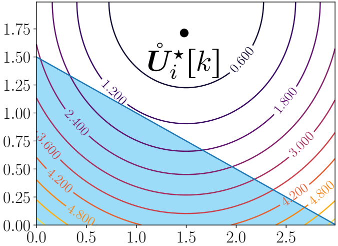
$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

CentraleSupélec

What are deprived systems?

Systems whose optimal solution has all constraints active

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

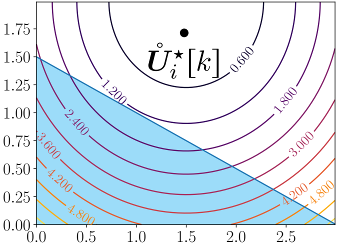


$$
\begin{aligned}
\underset{\boldsymbol{U}_i[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\
\text{subject to} \quad & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k]
\end{aligned}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

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \geq \boldsymbol{\theta}_i[k]$ → Scarce resources

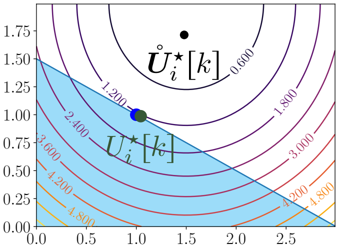


$$
\begin{array}{ll}
\underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T\boldsymbol{U}_i[k] \\
\text{subject to} & \bar{\Gamma}_i\boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k]
\end{array}
$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $\bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \geq \boldsymbol{\theta}_i[k] \rightarrow$ Scarce resources
  - Solution projected onto boundary

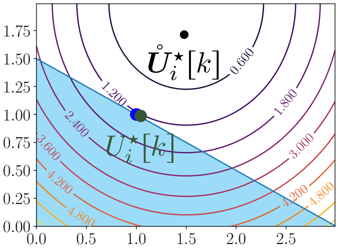


$$\begin{array}{ll} \underset{\boldsymbol{U}_i[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}_i[k]\right\|_{H_i}^2 + \boldsymbol{f}_i[k]^T \boldsymbol{U}_i[k] \\ \text{subject to} & \bar{\Gamma}_i \boldsymbol{U}_i[k] \preceq \boldsymbol{\theta}_i[k] : \boldsymbol{\lambda}_i[k] \end{array}$$

What are deprived systems?

Systems whose optimal solution has all constraints active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- $\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \geq \boldsymbol{\theta}_i[k] \rightarrow$ Scarce resources
  - Solution projected onto boundary
  - Same as with equality constraints[2]



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

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

Deprived Systems

Analysis

Assumptions

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *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

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

- P_i is time invariant

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *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

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

(local parameters unknown by coordinator) $\left\{ \right.$

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

CentraleSupélec

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

CentraleSupélec

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[3] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

[3] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[3] $\hat{P}_i[k]$ 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

[3] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[3] $\hat{P}_i[k]$ 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 $\left\| \widehat{\tilde{P}}_i[k] - \bar{P}_i \right\|_F > \epsilon_P$

[3] Using Recursive Least Squares for example

CentraleSupélec

Detection Mechanism

- We estimate[3] $\hat{P}_i[k]$ 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

[3] Using Recursive Least Squares for example

Detection Mechanism

- We estimate[3] $\hat{P}_i[k]$ 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 $\widehat{\tilde{P}}_i[k]$?

[3] Using Recursive Least Squares for example

CentraleSupélec

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

- 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 $\boldsymbol{\lambda}_i^p$

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

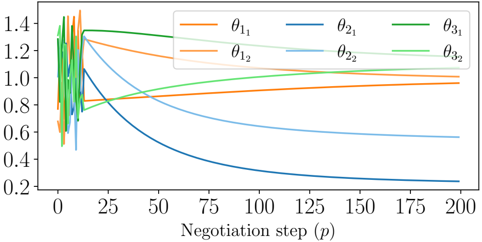
Estimating $\widehat{\tilde{P}}_i[k]$

- 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[4] sequence to increase excitation until convergence.

[4] 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[4] sequence to increase excitation until convergence.



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

Classification of mitigation techniques

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

CentraleSupélec

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 $\tilde{P}_i[k] = T_i[k]\bar{P}_i$

CentraleSupélec

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 $\boldsymbol{\lambda}_i$

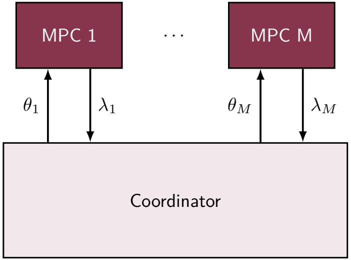
$$\overset{\text{rec}}{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i[k]^{-1}}\widehat{\tilde{\boldsymbol{s}}}_i[k]$$

- Choose adequate version for coordination

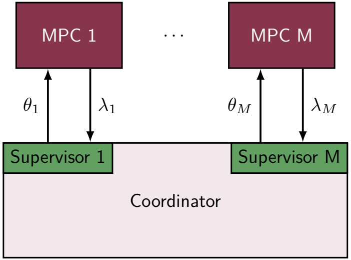
$$\overset{\text{mod}}{\boldsymbol{\lambda}}_i = \begin{cases} \overset{\text{rec}}{\boldsymbol{\lambda}}_i, & \text{if attack detected} \\ \tilde{\boldsymbol{\lambda}}_i, & \text{otherwise} \end{cases}$$

CentraleSupélec

Complete Mechanism

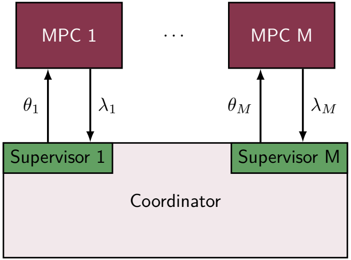


Complete Mechanism



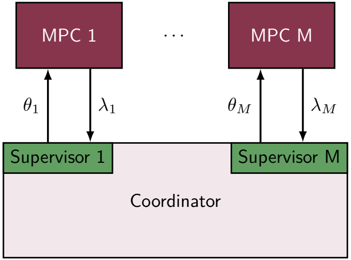
- Supervise exchanges by inquiring the agents

Complete Mechanism



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

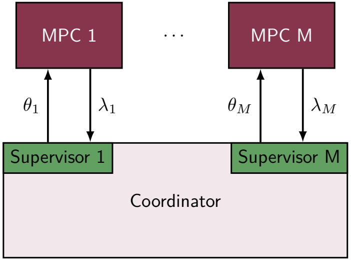
Complete Mechanism



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

Two Phases

Complete Mechanism

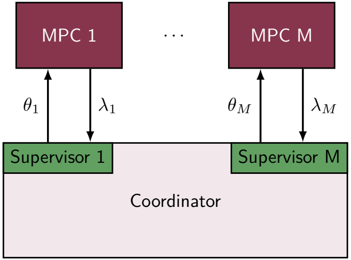


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative

Complete Mechanism



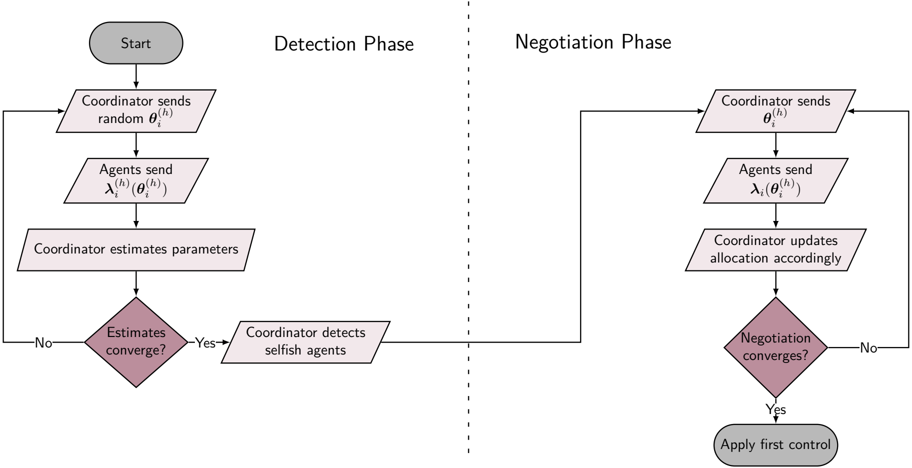
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

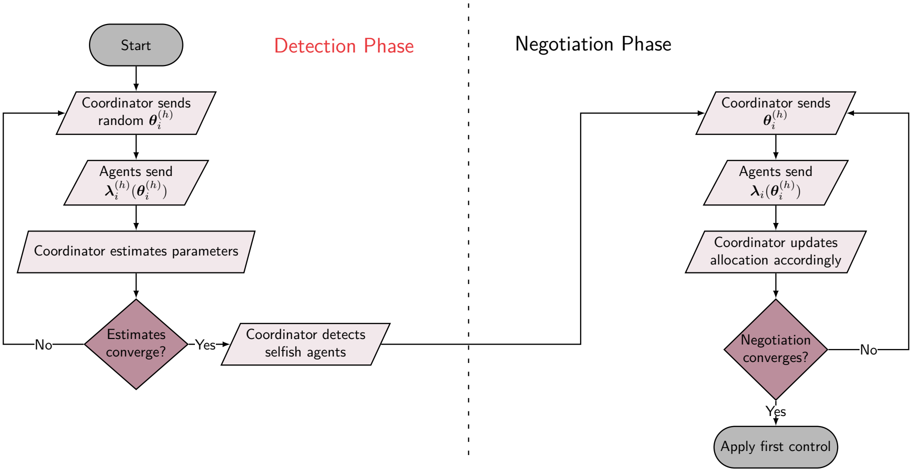
Complete algorithm

RPdMPC-DS



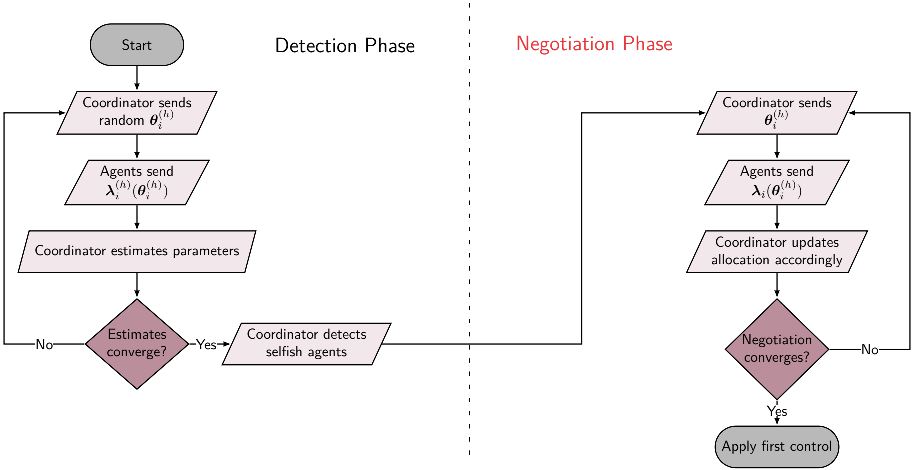
Complete algorithm

RPdMPC-DS



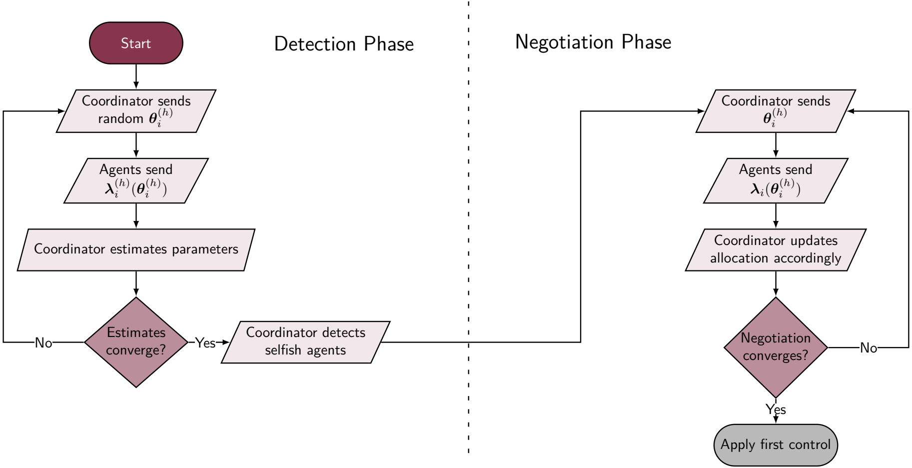
Complete algorithm

RPdMPC-DS



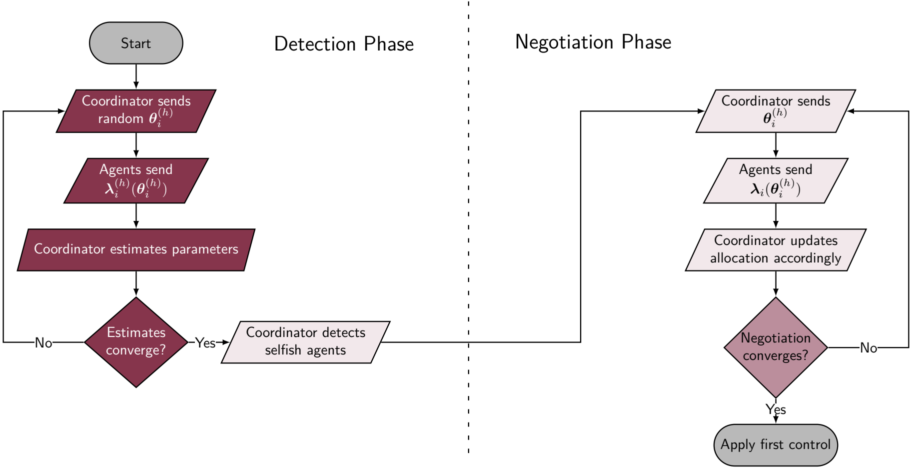
Complete algorithm

RPdMPC-DS



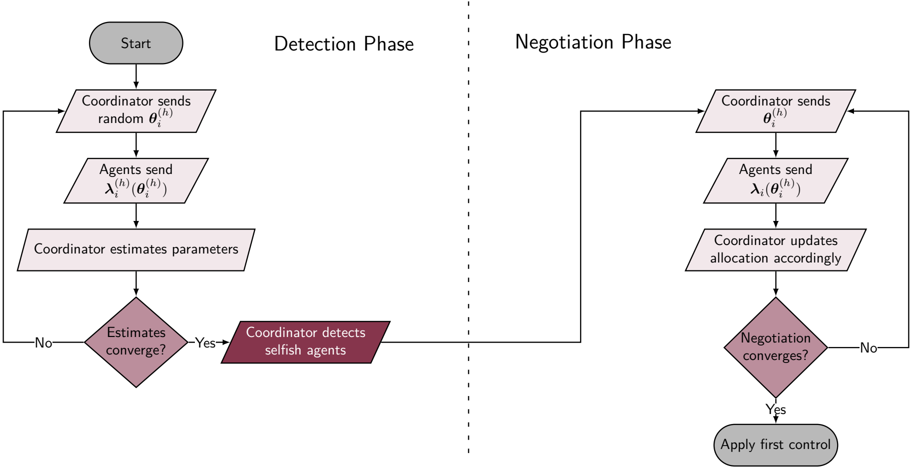
Complete algorithm

RPdMPC-DS



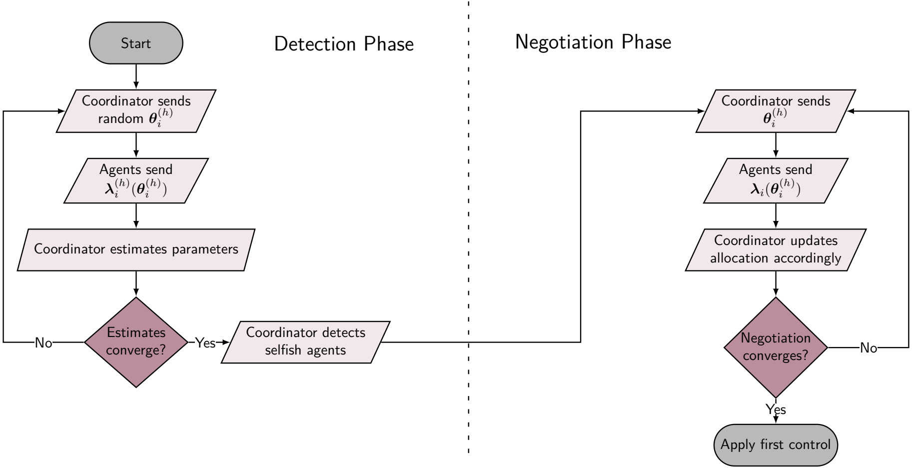
Complete algorithm

RPdMPC-DS



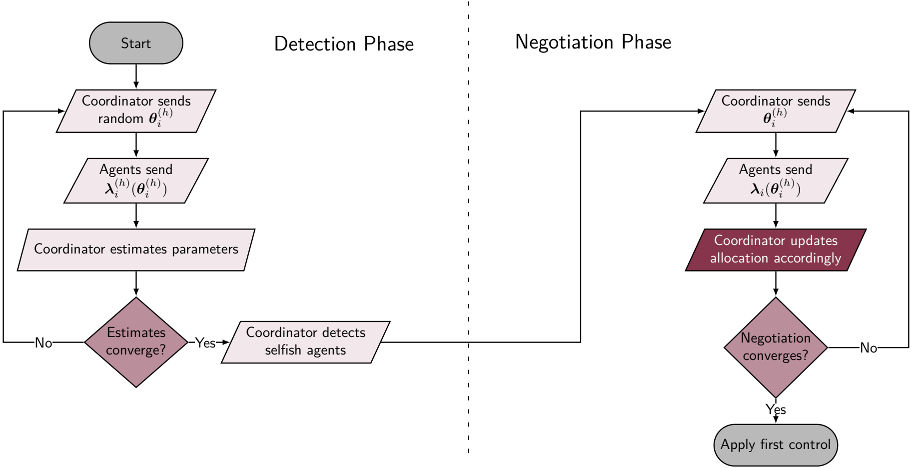
Complete algorithm

RPdMPC-DS

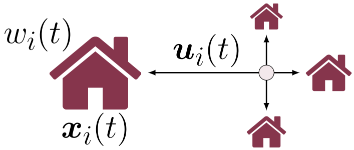


Complete algorithm

RPdMPC-DS

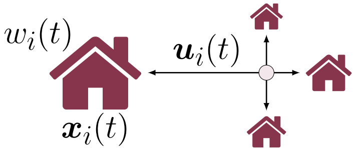


Example



District Heating Network (4 Houses)

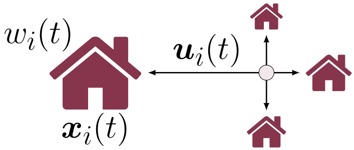
Example



District Heating Network (4 Houses)

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

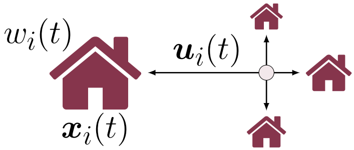
Example



District Heating Network (4 Houses)

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

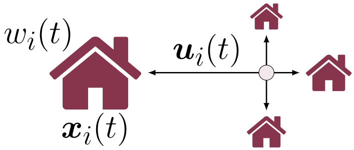
Example



District Heating Network (4 Houses)

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

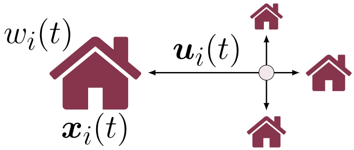
Example



District Heating Network (4 Houses)

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

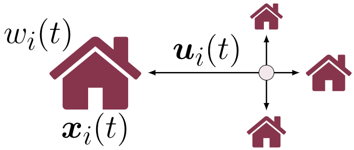
Example



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

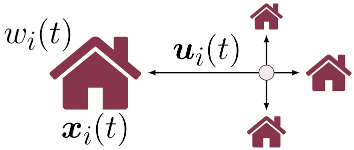
Example



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

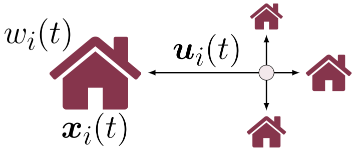
Example



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

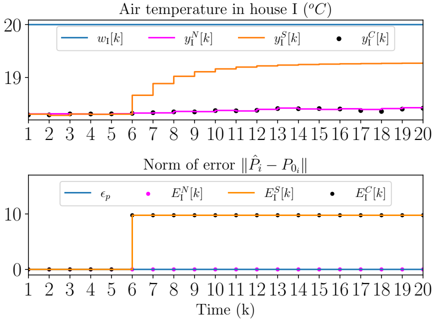


District Heating Network (4 Houses)

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

Results

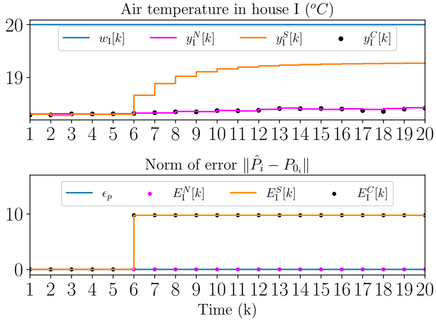
Temporal



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

Results

Temporal

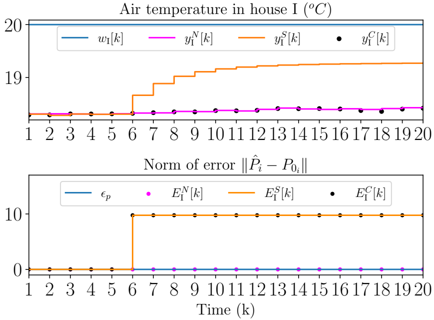


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

CentraleSupélec

Results

Temporal

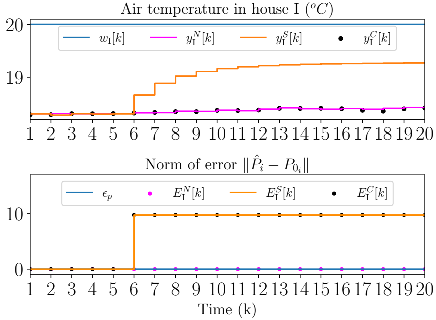


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$

Results

Temporal

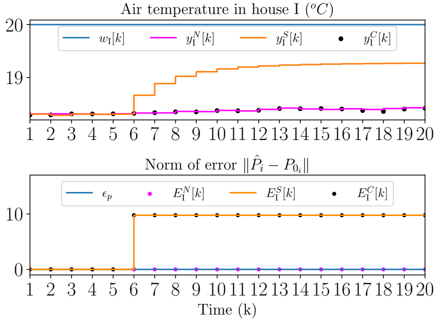


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selflish, C Corrected

- 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

CentraleSupélec

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

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

Relaxing scarcity assumption

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

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

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

CentraleSupélec

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

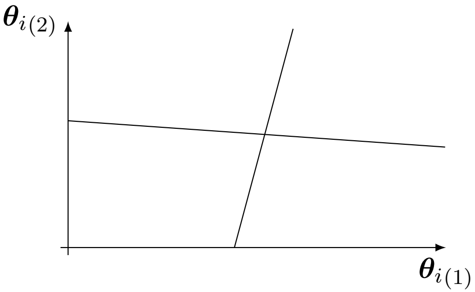
CentraleSupélec

Analyzing System

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

Analyzing System

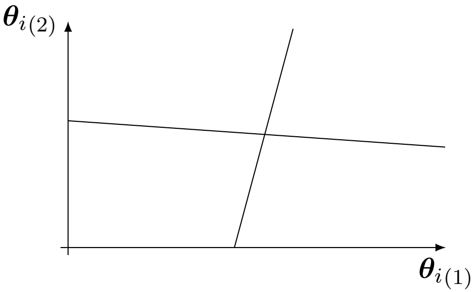
Solution for $\lambda_i[k]$ (Continued)



Separation surfaces depend on state and local parameters.

Analyzing System

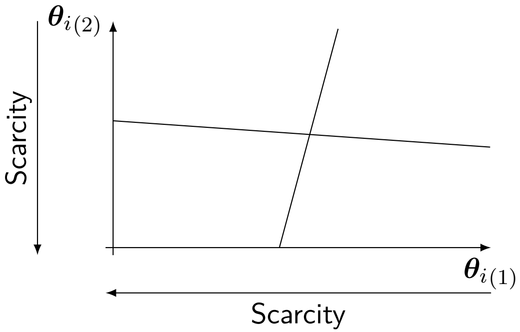
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

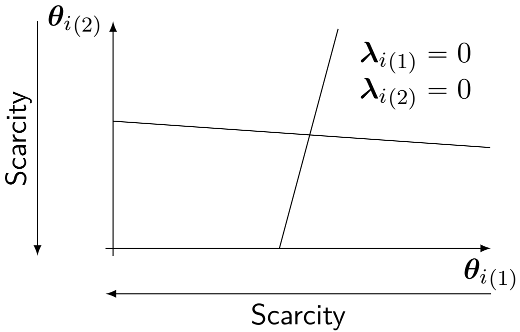
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

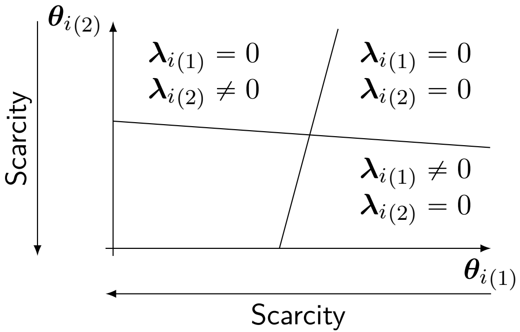
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

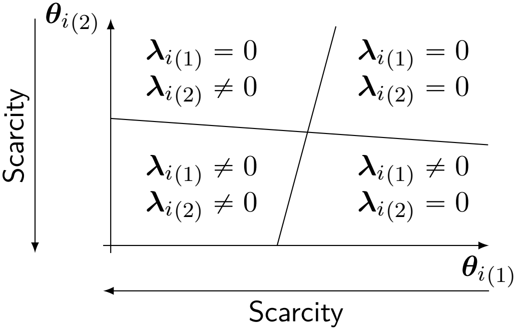
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

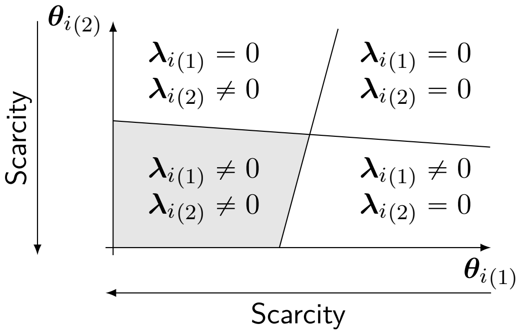
Solution for $\lambda_i[k]$ (Continued)



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?

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

Analyzing System

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

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

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?

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

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?

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

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	$-P_i^{(Z)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(Z)}[k]$	$\rightarrow$	$\mathbf{0}$

Assumptions

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

Analyzing System

Under attack!

Analyzing System

Under attack!

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

CentraleSupélec

Analyzing System

Under attack!

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

Parameters are modified.

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

CentraleSupélec

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

CentraleSupélec

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
- Solution: Let's force it using Artificial Scarcity

Artificial Scarcity

Who is it? Who is it?

Artificial Scarcity

Who is it? Who is it?

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]

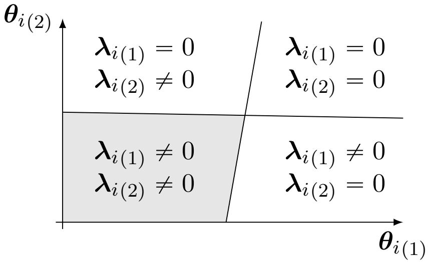
[5] If we have local constraints, we suppose this point respects them.

CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]

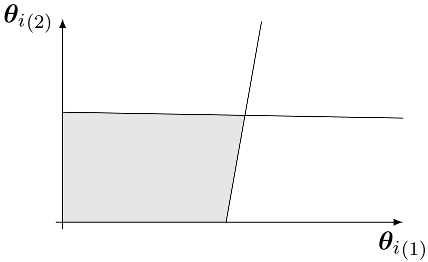


[5] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]

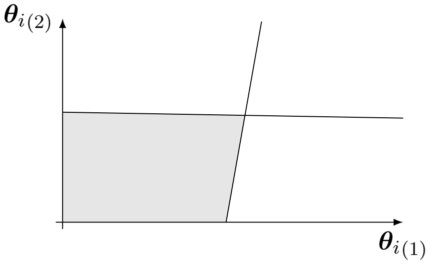


[5] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



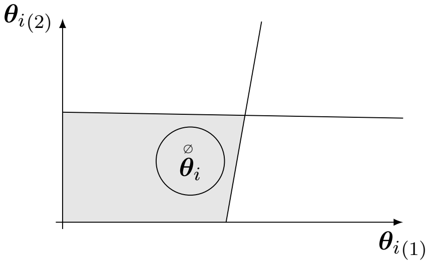
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



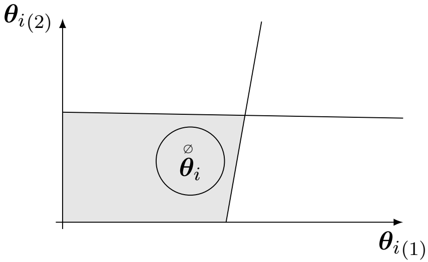
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



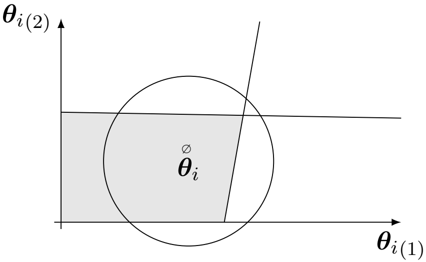
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



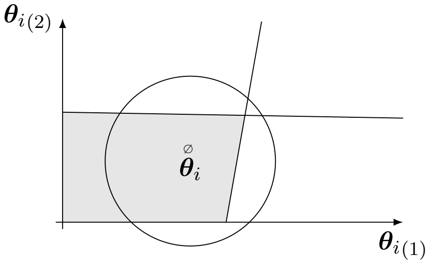
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



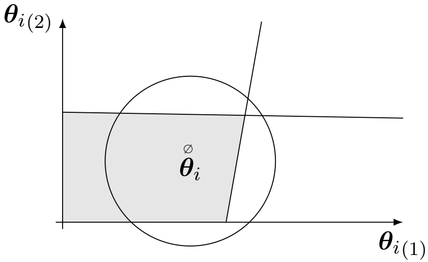
- Generate points close to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- Estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$
- How do we known the radius?
  - Unfortunately we don't.

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

Artificial Scarcity

Who is it? Who is it?

- We use the point $\overset{\varnothing}{\boldsymbol{\theta}}_i$, which activates all constraints[5]



- 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?

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

Enter Expectation Maximization

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$

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

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate

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

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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 $\tilde{\boldsymbol{\lambda}}_i^o[k]$

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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

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

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[6]
- 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., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[6] 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

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

CentraleSupélec

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

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

CentraleSupélec

Detection and Mitigation

Same same, but different

Assumption

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

- Detection

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

- Mitigation

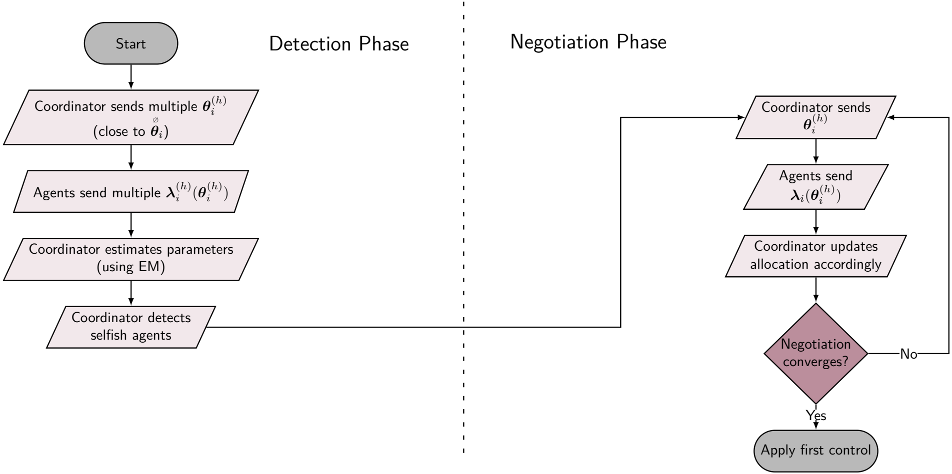
$$\widehat{T_i[k]^{-1}} = \bar{P}_i^{(0)} \widehat{\widetilde{P}}_i^{(0)}[k]^{-1}.$$

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

CentraleSupélec

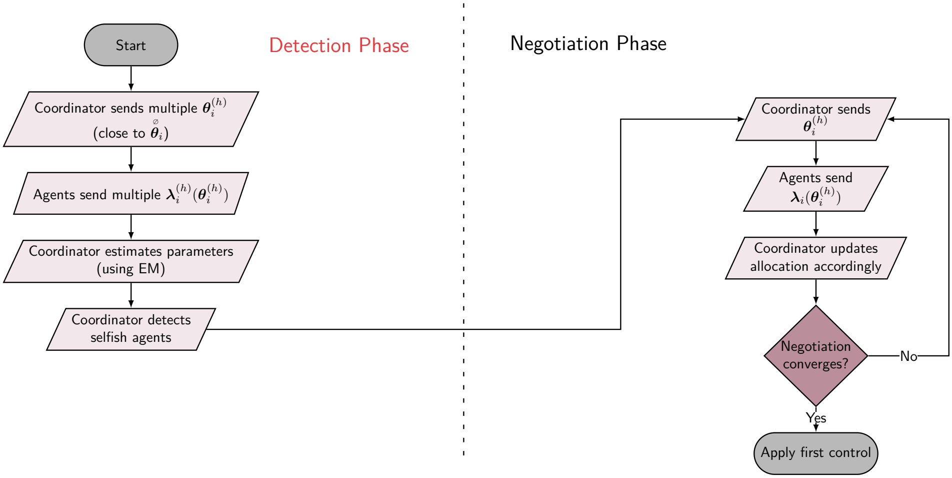
Complete algorithm

RPdMPC-AS



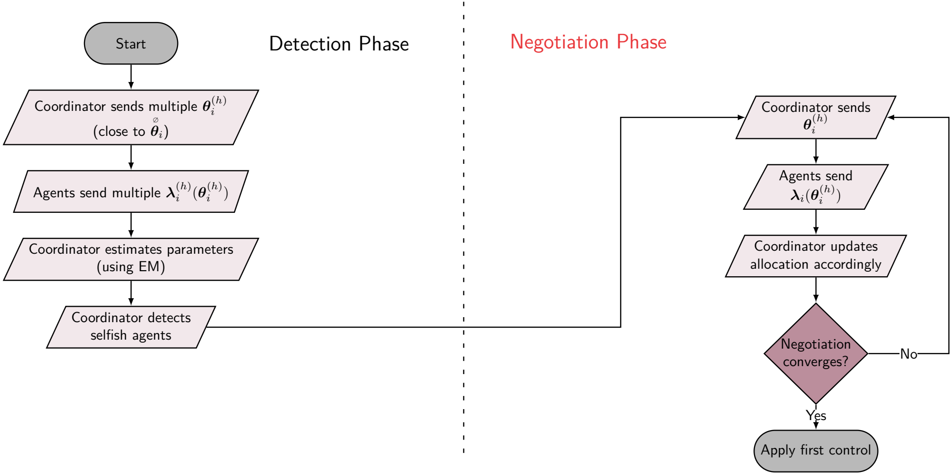
Complete algorithm

RPdMPC-AS



Complete algorithm

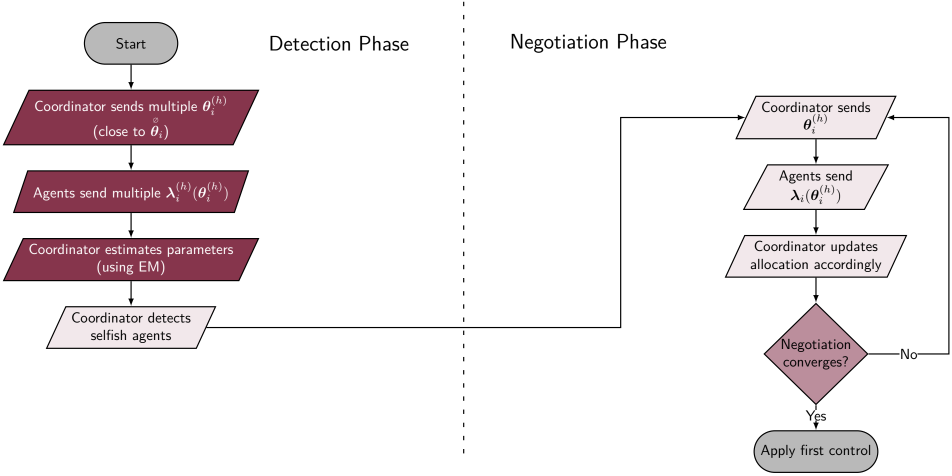
RPdMPC-AS



CentraleSupélec

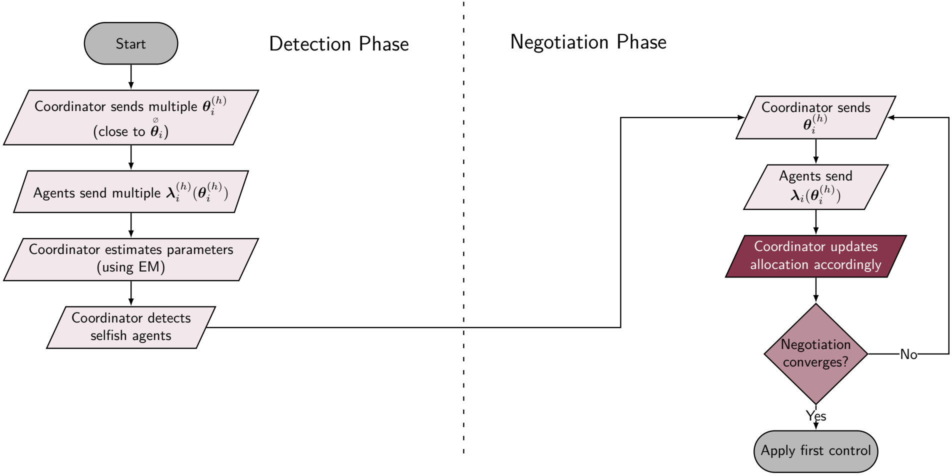
Complete algorithm

RPdMPC-AS

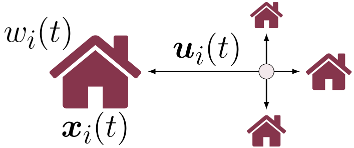


Complete algorithm

RPdMPC-AS



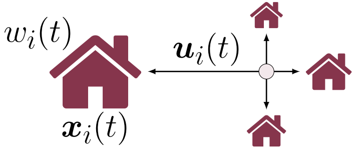
Example



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

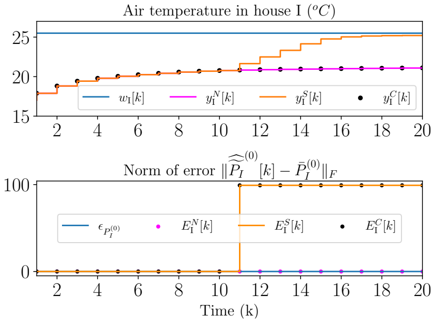


District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- ~~Not enough power~~ (Change $(\boldsymbol{x}_0, \boldsymbol{w}_0)$)
- Period of 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)

Results

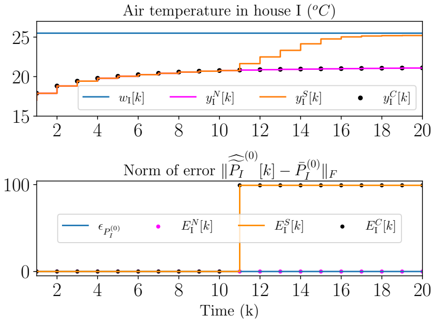
Temporal



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

CentraleSupélec

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":

CentraleSupélec

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

Outline

CentraleSupélec

Conclusion

Main takeaways

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

Conclusion

Main takeaways

- How can an agent attack? ✔
  - Attacker can change the communication to receive more ressources.
- What are the consequences of an attack?
- Can we mitigate the effects?

Conclusion

Main takeaways

- How can an agent attack? ✔
  - Attacker can change the communication to receive more ressources.
- What are the consequences of an attack? ✔
  - Suboptimality and maybe instability
- Can we mitigate the effects?

Conclusion

Main takeaways

- How can an agent attack? ✔
  - Attacker can change the communication to receive more ressources.
- What are the consequences of an attack? ✔
  - Suboptimality and maybe instability
- Can we mitigate the effects? ✔
  - Yes! By exploring the scarcity of the systems!

CentraleSupélec

Conclusion

Recap

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked

CentraleSupélec

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
  - Straightforward if system is completely deprived

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
  - Straightforward if system is completely deprived
  - If not, we try to force it artificially

CentraleSupélec

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
    1. Sensibilities are constant when there is no cheating
    2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
    - Straightforward if system is completely deprived
    - If not, we try to force it artificially
        - However, solution is PWA and we need special estimation method

CentraleSupélec

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
    1. Sensibilities are constant when there is no cheating
    2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
    - Straightforward if system is completely deprived
    - If not, we try to force it artificially
        - However, solution is PWA and we need special estimation method
- What if complete scarcity information is not available?

CentraleSupélec

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
  - Straightforward if system is completely deprived
  - If not, we try to force it artificially
    - However, solution is PWA and we need special estimation method
- What if complete scarcity information is not available? Not even artificially?

CentraleSupélec

Conclusion

Recap

- Insights from the analysis of the solutions of the optimization problems:
  1. Sensibilities are constant when there is no cheating
  2. They may change when system is attacked
- Exploiting the scarcity of resources, we find how to invert the attack
  - Straightforward if system is completely deprived
  - If not, we try to force it artificially
    - However, solution is PWA and we need special estimation method
- What if complete scarcity information is not available? Not even artificially?

Future Directions

- Reconstruction of cheating matrix with partial scarcity

Future Directions

- Reconstruction of cheating matrix with partial scarcity
- Study of robustness/Error Propagation + Add noise

CentraleSupélec

Future Directions

- Reconstruction of cheating matrix with partial scarcity
- Study of robustness/Error Propagation + Add noise
- Resilient strategy with soft constraints (QoS constraints)

CentraleSupélec

Future Directions

- Reconstruction of cheating matrix with partial scarcity
- Study of robustness/Error Propagation + Add noise
- Resilient strategy with soft constraints (QoS constraints)
- Recursive EM (or alternative method)

Future Directions

- Reconstruction of cheating matrix with partial scarcity
- Study of robustness/Error Propagation + Add noise
- Resilient strategy with soft constraints (QoS constraints)
- Recursive EM (or alternative method)
- ...

Questions? Comments?

Repository
https://github.com/Accacio/thesis

Contact
rafael.accacio.nogueira@gmail.com

For Further Reading I

K.J. Åström 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.

José M Maestre, Rudy R Negenborn, et al. Distributed Model Predictive Control made easy. Vol. 69. Springer, 2014. ISBN: 978-94-007-7005-8.

Wicak Ananduta et al. "Resilient Distributed Model Predictive Control for Energy Management of Interconnected Microgrids". In: Optimal Control Applications and Methods 41.1 (2020), pp. 146–169. DOI: 10.1002/oca.2534. URL: https://onlinelibrary.wiley.com/doi/pdf/10.1002/oca.2534.

For Further Reading II

José M. Maestre et al. “Scenario-Based Defense Mechanism Against Vulnerabilities in Lagrange-Based Dmpc”. In: Control Eng Pract 114 (2021), p. 104879. ISSN: 0967-0661. DOI: 10.1016/j.conengprac.2021.104879.

Rafael Accácio Nogueira et al. “Expectation-Maximization Based Defense Mechanism for Distributed Model Predictive Control”. In: IFAC-PapersOnLine 55.13 (2022). 9th IFAC Conference on Networked Systems NECSYS 2022, pp. 73–78. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.07.238.

Pablo Velarde et al. “Vulnerabilities in Lagrange-Based Distributed Model Predictive Control”. In: Optimal Control Applications and Methods 39.2 (Sept. 2018), pp. 601–621. DOI: 10.1002/oca.2368.

CentraleSupélec

For Further Reading III

Wicak Ananduta et al. “Resilient Distributed Energy Management for Systems of Interconnected Microgrids”. In: 2018 IEEE Conference on Decision and Control (CDC). 2018, pp. 3159–3164. DOI: 10.1109/CDC.2018.8619548.

Wicak Ananduta et al. “A Resilient Approach for Distributed MPC-Based Economic Dispatch in Interconnected Microgrids”. In: 2019 18th European Control Conference (ECC). 2019, pp. 691–696. DOI: 10.23919/ECC.2019.8796208.

P. Chanfreut, J. M. Maestre, and H. Ishii. “Vulnerabilities in Distributed Model Predictive Control based on Jacobi-Gauss Decomposition”. In: 2018 European Control Conference (ECC). June 2018, pp. 2587–2592. DOI: 10.23919/ECC.2018.8550239.

For Further Reading IV

Rafael Accácio Nogueira, Romain Bourdais, and Hervé Guéguen. "Detection and Mitigation of Corrupted Information in Distributed Model Predictive Control Based on Resource Allocation". In: 2021 5th Conference on Control and Fault-Tolerant Systems (SysTol). 2021, pp. 329–334. DOI: 10.1109/SysTol52990.2021.9595927.

Pablo Velarde et al. "Scenario-based defense mechanism for distributed model predictive control". In: 2017 IEEE 56th Annual Conference on Decision and Control (CDC). IEEE. Dec. 2017, pp. 6171–6176. DOI: 10.1109/CDC.2017.8264590.

CentraleSupélec

For Further Reading V

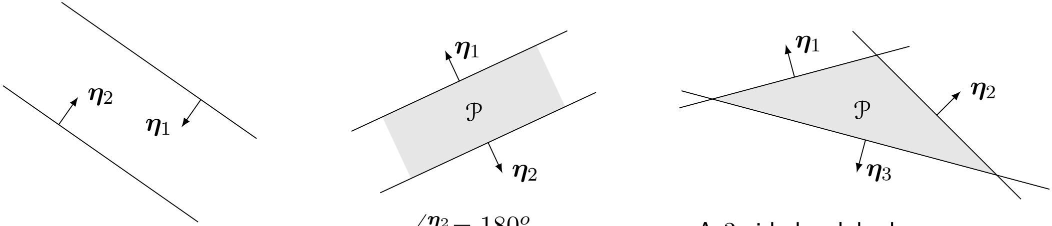
Pablo Velarde et al. “Vulnerabilities in Lagrange-Based DMPC in the Context of Cyber-Security”. In: 2017 IEEE International Conference on Autonomic Computing (ICAC). July 2017, pp. 215–220. DOI: 10.1109/ICAC.2017.53.

CentraleSupélec

Conditions

back

One way to ensure this, is to make the original constraints to form a cone.



No intersection

$\langle^{\eta_2}_{\eta_1} = 180^o$

A 3-sided polyhedron.

θ dynamics

◂ back

$$\boldsymbol{\theta}^{(p+1)} = \mathcal{A}_\theta \boldsymbol{\theta}^{(p)} + \mathcal{B}_\theta[k]$$

where

$$\mathcal{A}_\theta = \begin{bmatrix} I - \frac{M-1}{M}\rho^{(p)}P_1 & \frac{1}{M}\rho^{(p)}P_2 & \dots & \frac{1}{M}\rho^{(p)}P_M \\ \frac{1}{M}\rho^{(p)}P_1 & I - \frac{M-1}{M}\rho^{(p)}P_2 & \dots & \frac{1}{M}\rho^{(p)}P_M \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M}\rho^{(p)}P_1 & \frac{1}{M}\rho^{(p)}P_2 & \dots & I - \frac{M-1}{M}\rho^{(p)}P_M \end{bmatrix}$$

$$\mathcal{B}_\theta[k] = \begin{bmatrix} -\frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_1[k] + \frac{1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \dots - \frac{1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \\ \frac{1}{M}\rho^{(p)}\boldsymbol{s}_1[k] - \frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \dots - \frac{1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \\ \vdots \\ \frac{1}{M}\rho^{(p)}\boldsymbol{s}_1[k] + \frac{1}{M}\rho^{(p)}\boldsymbol{s}_2[k] \dots - \frac{M-1}{M}\rho^{(p)}\boldsymbol{s}_M[k] \end{bmatrix}$$

CentraleSupélec

Parameters estimated depending on Prediction Horizon N

\# constraints depend on \# global constraints c and prediction horizon N

- Number of Regions $= 2^{Nc}$
- Parameters in each region $=$ Matrix $P_i^{(z)} = (Nc)^2 +$ vector $\boldsymbol{s}_i^{(z)}[k] = Nc$
  - Total $((Nc)^2 + Nc)2^{Nc}$

Some examples

- 1 constraint
  - $N = 3 \rightarrow$ 96 elements
  - $N = 4 \rightarrow$ 320 elements

Remark

We can reduce number of elements estimated from $P_i^{(z)}$ if we assume $P_i^{(z)} \in \mathbb{S}$
New total $\rightarrow ((Nc)^2 + 3Nc)2^{Nc-1}$