Security of distributed Model Predictive Control under False Data injection

Rafael Accácio NOGUEIRA

2022-12-07

https://bit.ly/3g3S6X4

IETR

CentraleSupélec

Context

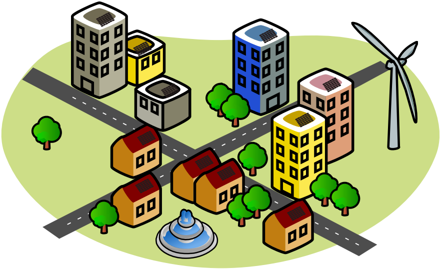
❝ Necessity is the mother of invention ❞

Context

"Necessity is the mother of invention"

Context

“Necessity is the mother of invention”



- Electricity Distribution System
- Heat distribution
- Water distribution
- Traffic management
  (include your problem here)

CentraleSupélec

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”

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

  (include your problem here)

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

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

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

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

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

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

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

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

Find best control sequence using predictions based on a model.

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

CentraleSupélec

Find best control sequence using predictions based on a model.

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

Model-based Predictive Control

Find best control sequence using predictions based on a model.

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

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
  - Objective function to optimize
  - System's Model (states and inputs)
  - Other constraints to respect

Find optimal control sequence using predictions based on a model.

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

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

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
  - Objective function to optimize
  - System's Model (states and inputs)
  - Other constraints to respect (QoS, technical restrictions, ...)

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

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

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)
  - Other constraints to respect (QoS, technical restrictions, ...)

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

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)
  - Other constraints to respect (QoS, technical restrictions, ...)

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

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)
  - Other constraints to respect (QoS, technical restrictions, ...)

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

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

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, ...)

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

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, ...)

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

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, ...)

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

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, ...)

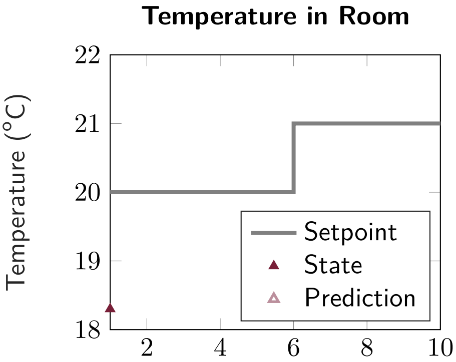
$$
\begin{aligned}
\underset{\boldsymbol{u}[0:N-1|k]}{\text{minimize}} \quad & J(\boldsymbol{x}[0|k], \boldsymbol{u}[0:N-1|k]) \\
\text{subject to} \quad & \left.\begin{array}{r} \boldsymbol{x}[\xi|k] = f(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k]) \\ g_i(\boldsymbol{x}[\xi-1|k], \boldsymbol{u}[\xi-1|k])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

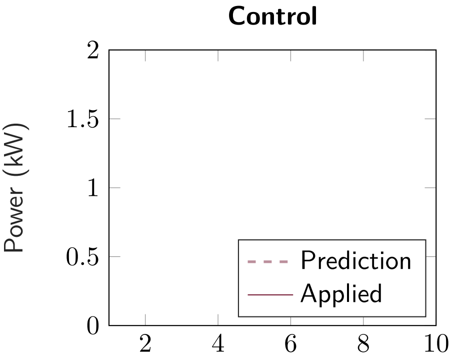
CentraleSupélec

Model Predictive Control

In a nutshell

Find optimal control sequence

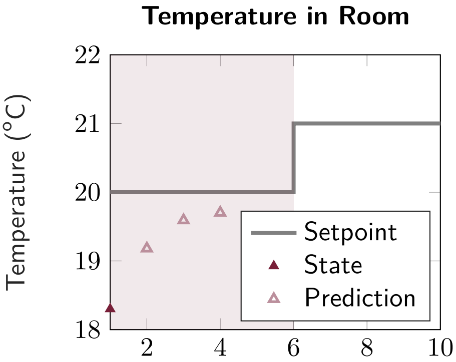


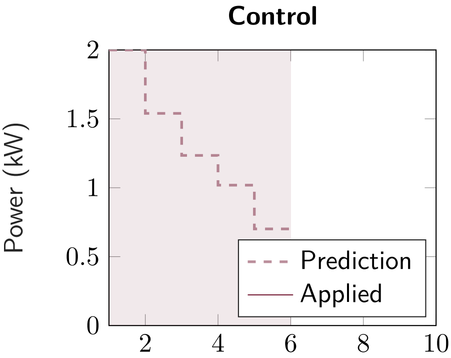


Model Predictive Control

In a nutshell

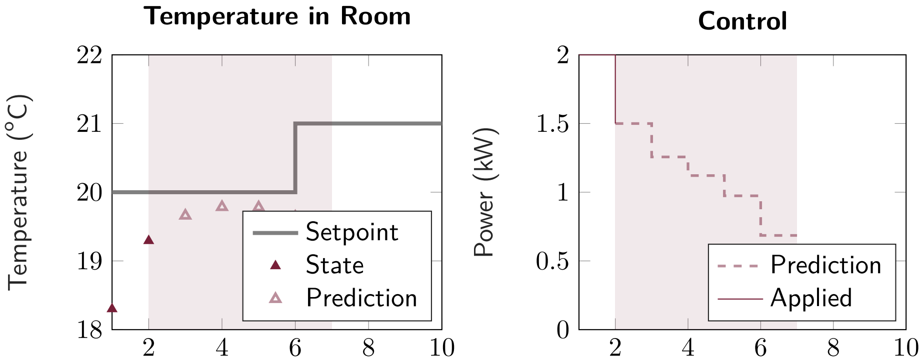
Find optimal control sequence





In a nutshell

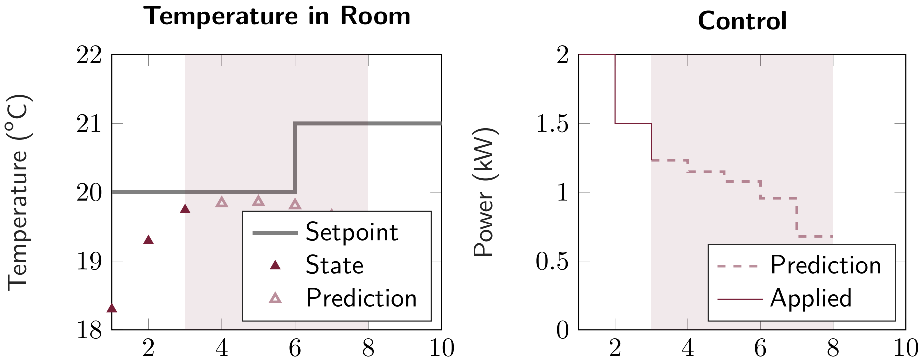
Find optimal control sequence, apply first element



Model Predictive Control

In a nutshell

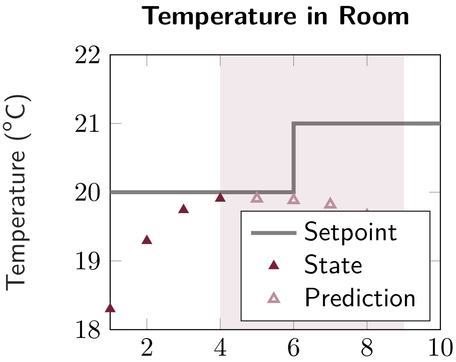
Find optimal control sequence, apply first element, rinse repeat

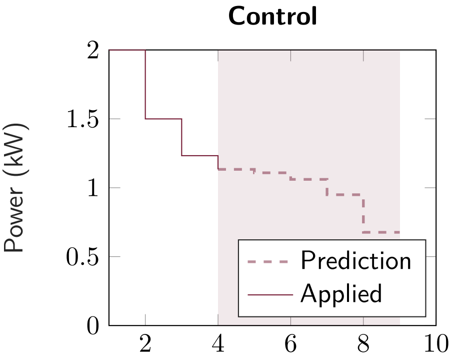


Model Predictive Control

In a nutshell

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

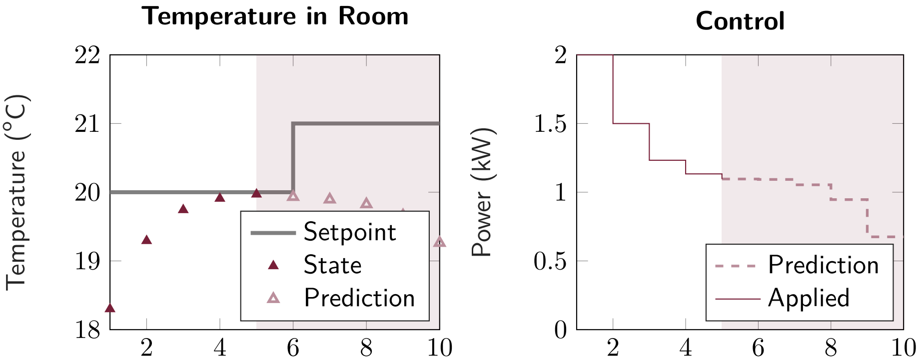




Model Predictive Control

In a nutshell

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



Model Predictive Control

Nothing is perfect

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

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 Systems Communicate

Nothing is perfect

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

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 Systems Communicate

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 Systems Communicate

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 Systems Communicate

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 Systems Communicate

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 Systems Communicate

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 Systems Communicate

CentraleSupélec

Distributed Model Predictive Control

It is about communication

- We break the MPC into multiple
- Make them Communicate
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...

[1]

Distributed Model Predictive Control

It is about communication

- We break the MPC into multiple
- Make them Communicate
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...

Distributed Model Predictive Control

It is about communication

- We break the MPC into multiple
- Make them Communicate
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...

[1]

Distributed Model Predictive Control

It is about communication

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

1

CentraleSupélec

It is about communication

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

1

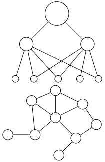
It is about communication

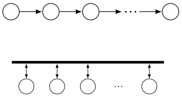
- We break the MPC into multiple
- Make them Communicate , But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...

1

It is about communication

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

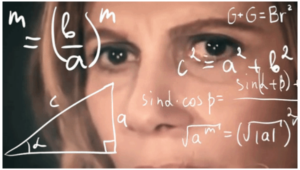


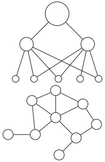


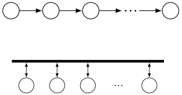
1

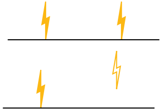
It is about communication

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









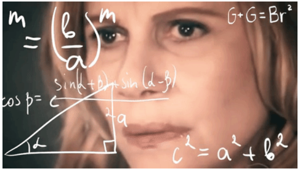
1

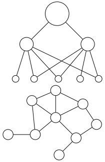
CentraleSupélec

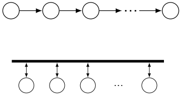
Distributed Model Predictive Control

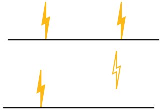
It is about communication

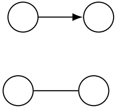
- We break the MPC into multiple
- Make them Communicate , But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...











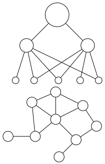
1

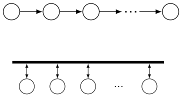
CentraleSupélec

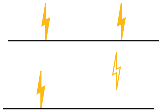
Distributed Model Predictive Control

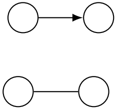
It is about communication

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







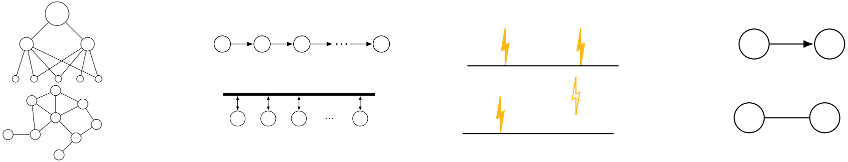


[1] Distributed Model Predictive Control made easy

Distributed Model Predictive Control

It is about communication

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

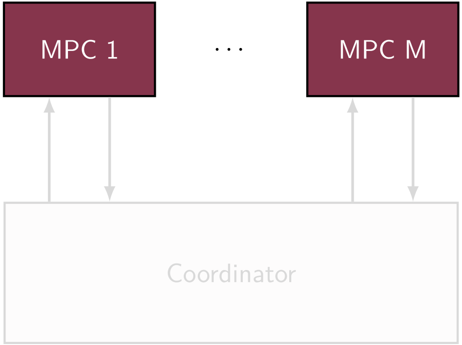


[1] Distributed Model Predictive Control made easy

CentraleSupélec

Distributed Model Predictive Control

Optimization Frameworks

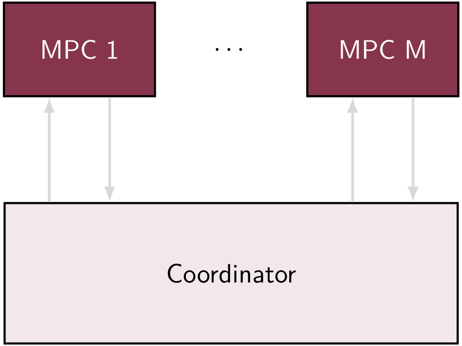


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

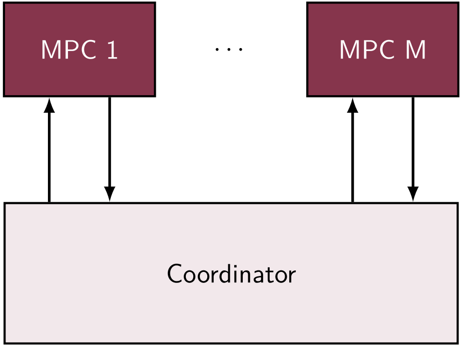


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

$\Big\}$ Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

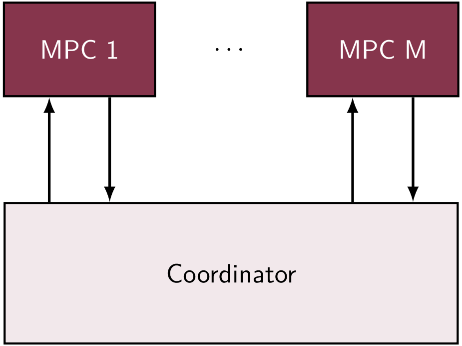


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

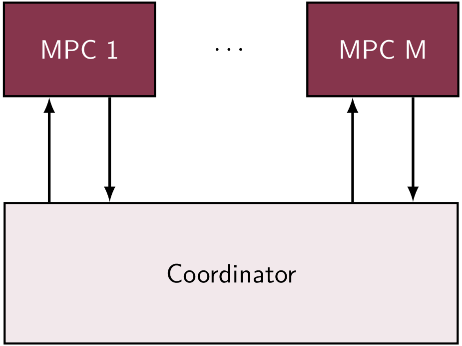


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks



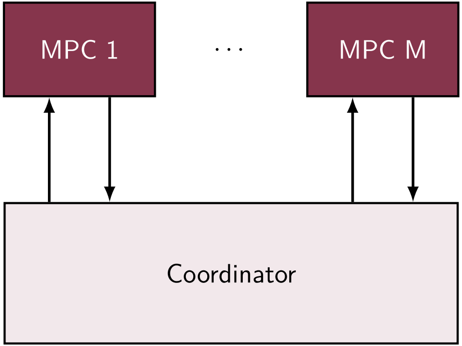
- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

CentraleSupélec

Distributed Model Predictive Control

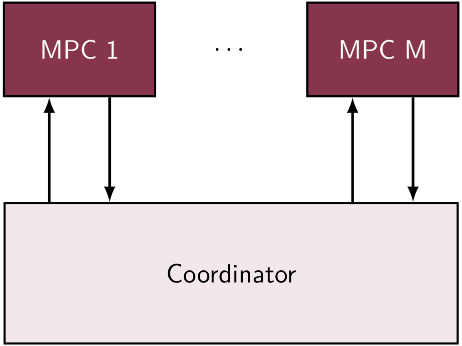
Optimization Frameworks



- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Optimization Frameworks



- Coordinator $\longrightarrow$ Hierarchical
- Bidirectional
- No delay $\longrightarrow$ Synchronous
- Agents solve local problems
- Variables are updated

$\left.\begin{array}{l}\text{Agents solve local problems}\\ \text{Variables are updated}\end{array}\right\}$ Until Convergence

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?

Let's have a preview!

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?

Let's have a preview!

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!

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

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!

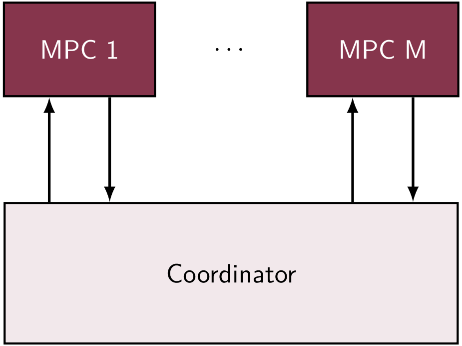
Distributed Model Predictive Control

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!

How can a non-cooperative agent attack?

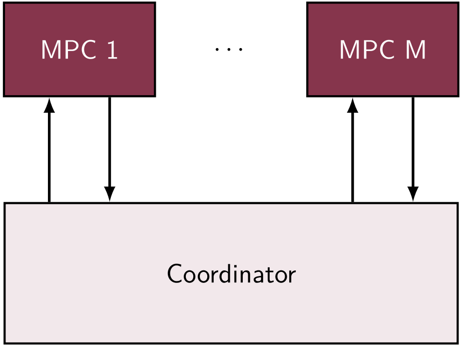


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

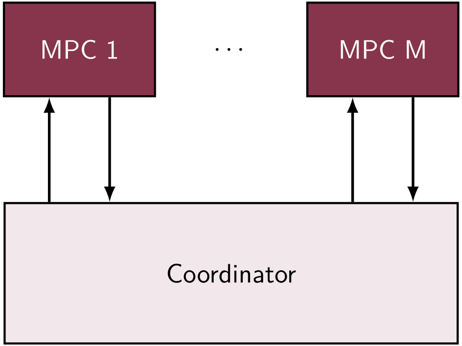


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

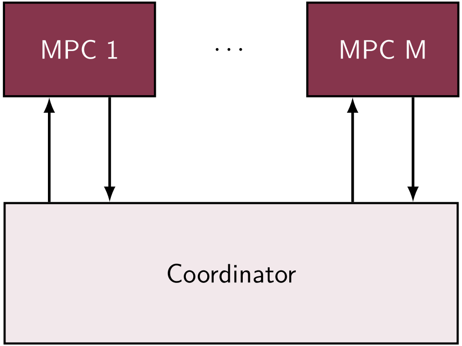


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

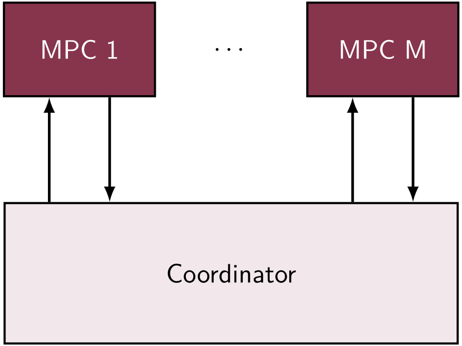


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

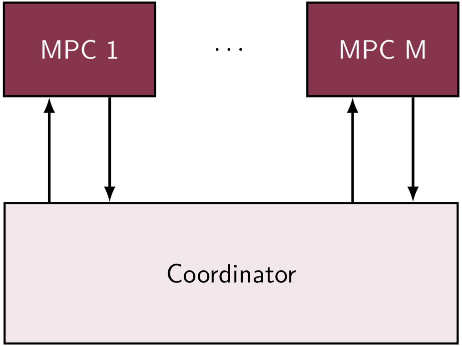


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

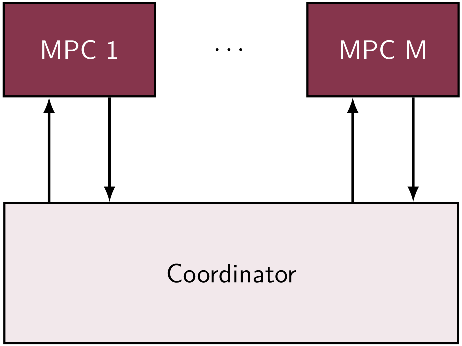


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

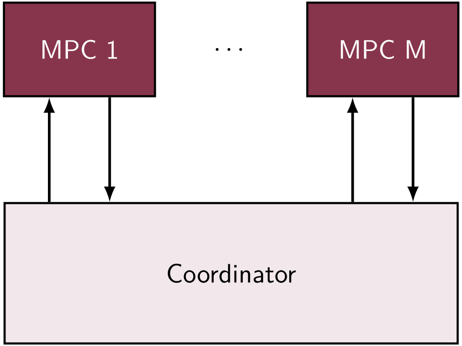


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

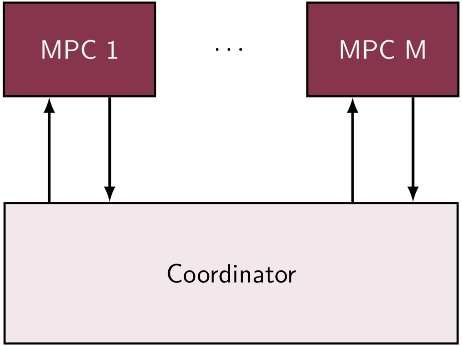


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

$\left.\right\}$ Deception Attacks

How can a non-cooperative agent attack?

Literature

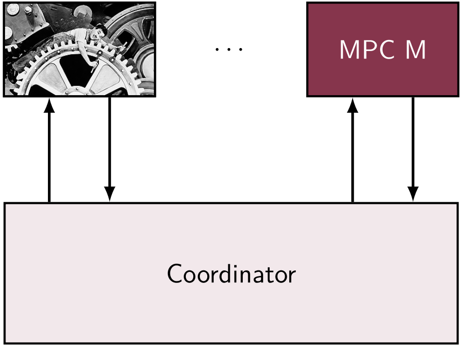


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

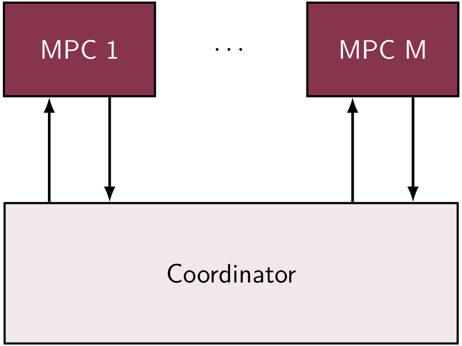


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks (Internal change)

How can a non-cooperative agent attack?

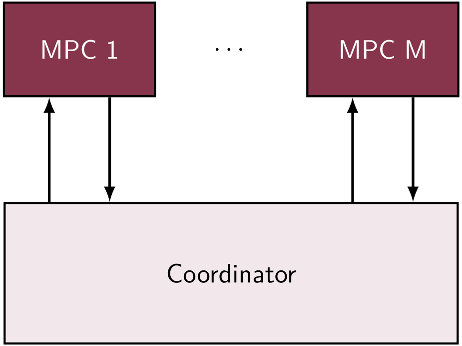
Our approach



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

How can a non-cooperative agent attack?

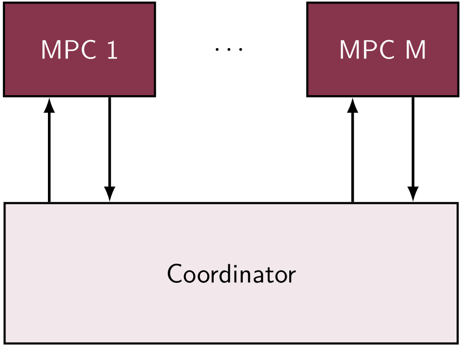
Our approach



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

How can a non-cooperative agent attack?

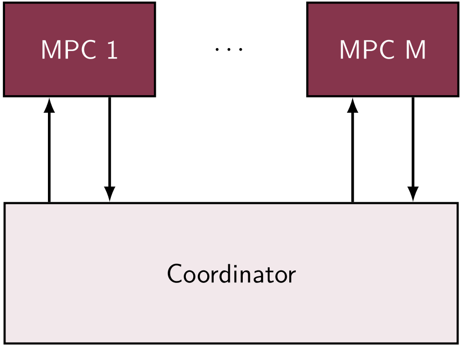
Our approach



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

How can a non-cooperative agent attack?

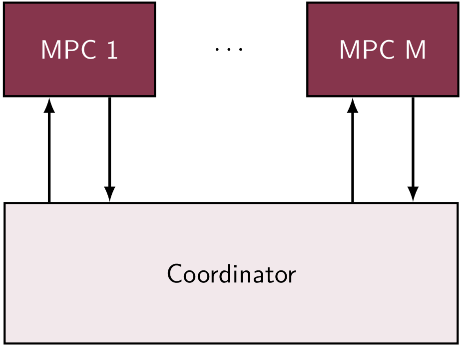
Our approach



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

How can a non-cooperative agent attack?

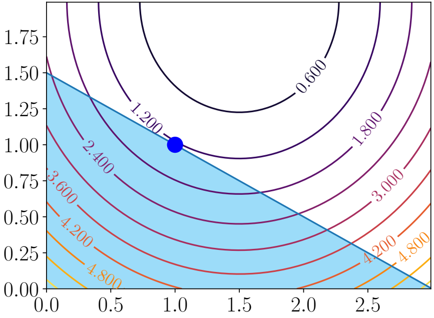
Our approach



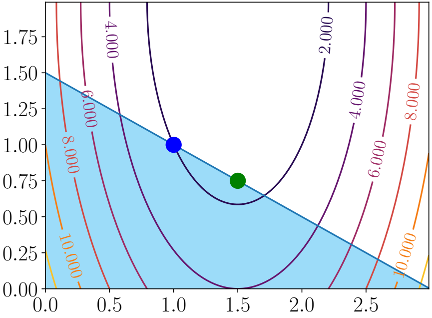
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

Consequence of an attack

- Attack modifies optimization problem
  - Optimum value is shifted



Original minimum.

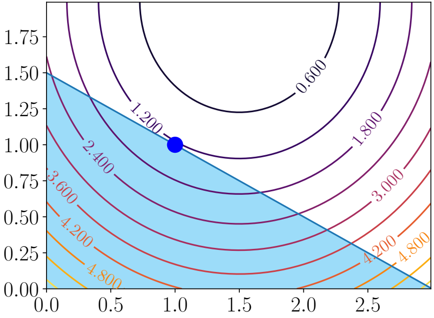


Minimum after attack.

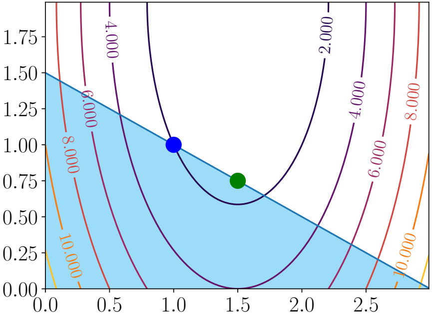
CentraleSupélec

Consequence of an attack

- Attack modifies optimization problem
  - Optimum value is shifted



Original minimum.

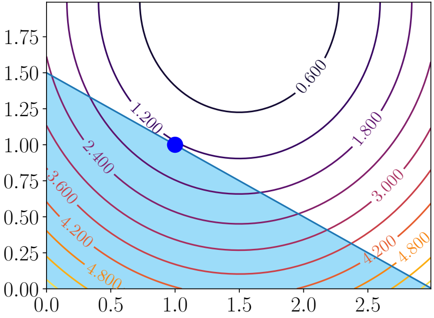


Minimum after attack.

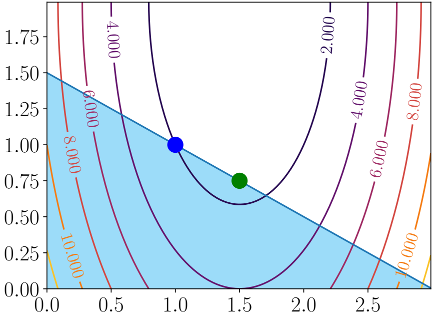
CentraleSupélec

Consequence of an attack

- Attack modifies optimization problem
    - Optimum value is shifted



Original minimum.



Minimum after attack.

Mitigating the effects

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

CentraleSupélec

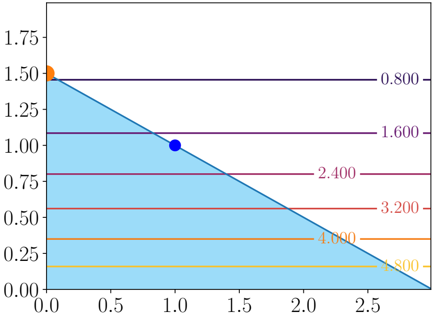
Mitigating the effects

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

CentraleSupélec

Mitigating the effects

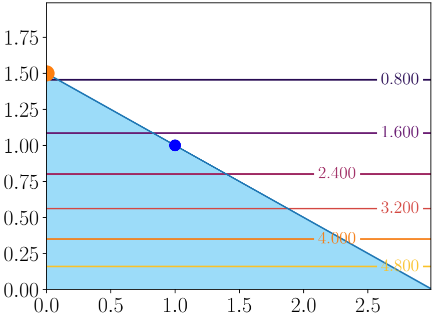
- We can recover by
  - Ignoring attacker
  - Recuperating original behavior (at least trying)



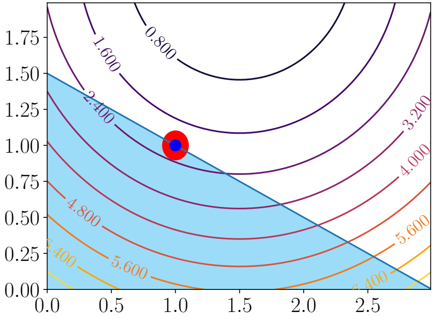
Ignore attacker.

Mitigating the effects

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



Ignore attacker.

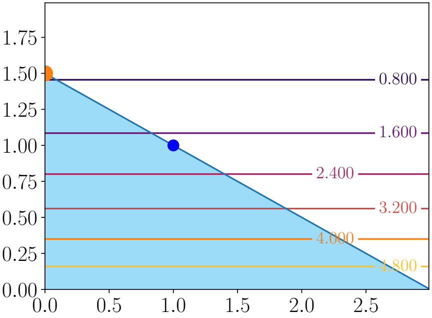


Recover original behavior.

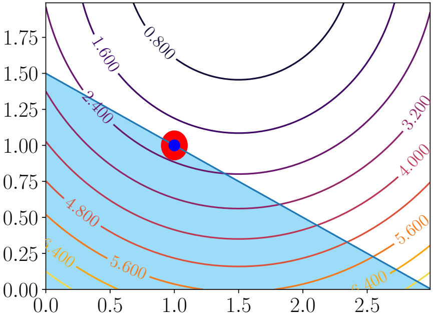
CentraleSupélec

Mitigating the effects

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



Ignore attacker.



Recover original behavior.

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

Attack free

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes $\begin{cases} \text{Attack free} \\ \text{When attack detected} \end{cases}$
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes $\begin{cases} \text{Attack free} \\ \text{When attack detected} \end{cases}$
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes $\begin{cases} \text{Attack free} \\ \text{When attack detected} \end{cases}$
  1. Detection/Isolation
  2. Mitigation

CentraleSupélec

Classification of mitigation techniques

- Passive (Robust) - 1 mode
- Active (Resilient) - 2 modes
  1. Detection/Isolation
  2. Mitigation

$\begin{cases} \text{Attack free} \\ \text{When attack detected} \end{cases}$

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

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

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

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

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
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

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

Outline

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

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

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

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

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- 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 $i \in \mathcal{M}$

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{u_1,\dots,u_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(x_i,u_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} h_i(x_i,u_i) \preceq u_{\text{total}} \end{aligned}$$

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

$$\begin{aligned} \underset{u_i}{\text{minimize}} \quad & J_i(x_i,u_i) \\ \text{s. t.} \quad & h_i(x_i,u_i) \preceq \theta_i : \lambda_i \end{aligned}$$

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{u_1,\dots,u_M}{\text{minimize}} \quad & \sum_{i \in \mathcal{M}} J_i(x_i, u_i) \\ \text{s.t.} \quad & \sum_{i \in \mathcal{M}} h_i(x_i, u_i) \preceq u_{\text{total}} \end{aligned}$$

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

$$\begin{aligned} \underset{u_i}{\text{minimize}} \quad & J_i(x_i, u_i) \\ \text{s. t.} \quad & h_i(x_i, u_i) \preceq \theta_i : \lambda_i \end{aligned}$$

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- 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 $i \in \mathcal{M}$

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
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

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{u_1,\dots,u_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

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

1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

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

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

$$\begin{aligned} \underset{u_1,\ldots,u_M}{\text{minimize}} \quad & \sum_{i\in\mathcal{M}} J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s.t.} \quad & \sum_{i\in\mathcal{M}} \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{u}_{\text{total}} \end{aligned}$$

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

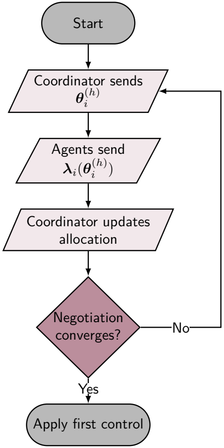
1. Allocate a part for each agent
2. They solve local problems and
3. communicate how dissatisfied
4. Allocation is updated (respecting global constraint)

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

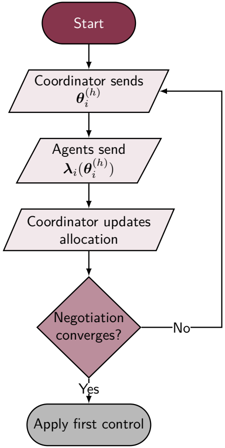
$$\boldsymbol{\theta}[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

Quantity Decomposition | Resource Allocation

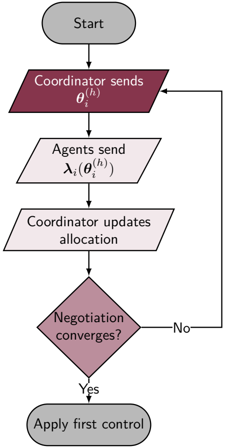


Quantity Decomposition | Resource Allocation



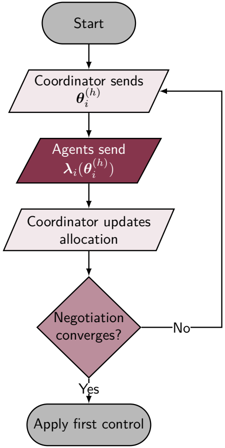
CentraleSupélec

Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

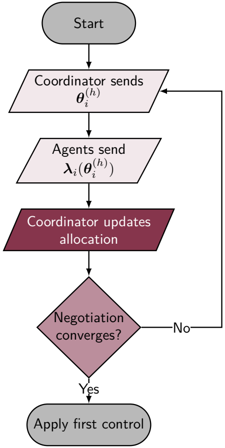
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

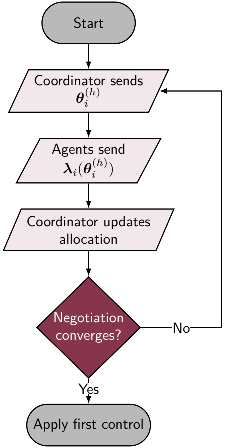
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

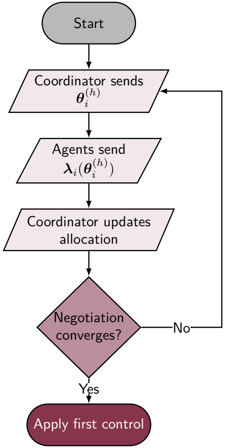
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Quantity Decomposition | Resource Allocation

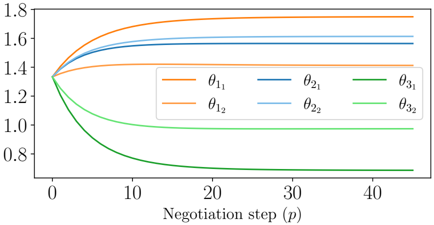


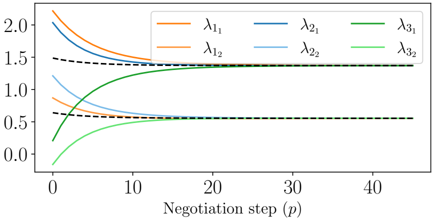
Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Quantity Decomposition | Resource Allocation

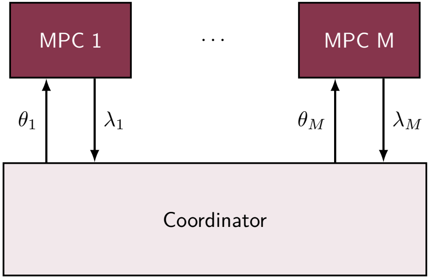
Until everybody is equally dissatisfied





How can a non-cooperative agent attack?

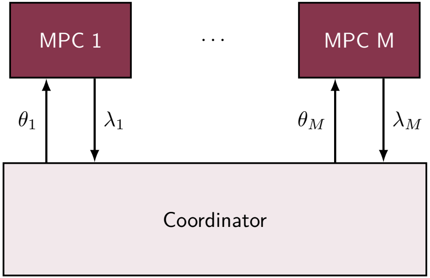
Our approach



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

How can a non-cooperative agent attack?

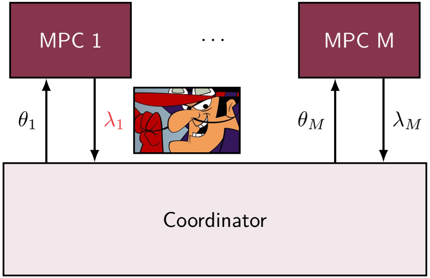
Our approach



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

How can a non-cooperative agent attack?

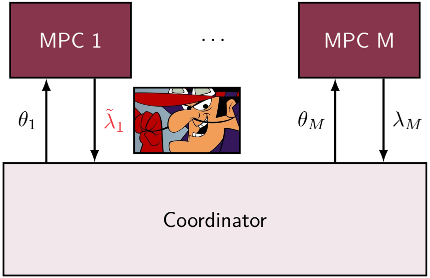
Our approach



- $\boldsymbol{\lambda}_i$ 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

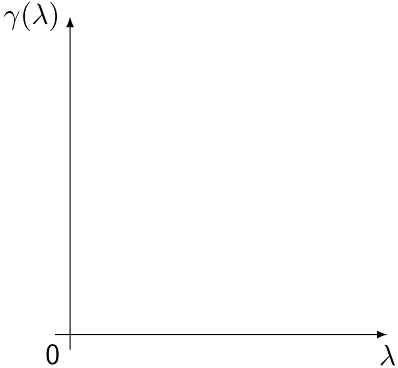


- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters
- Malicious agent modifies $\boldsymbol{\lambda}_i$

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

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

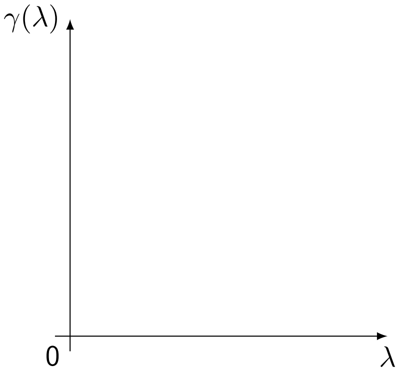
- *Attacker is not naïve*
  $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

- *Attacker is not naïve*

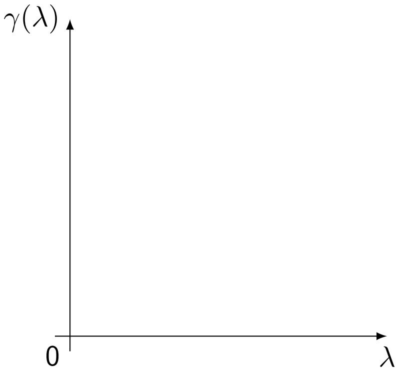
  $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*

  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

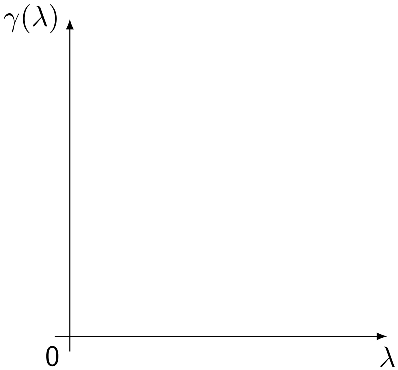
Assumptions

- *Attacker is not naïve*
  $\gamma(\lambda) = 0 \to \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  $\lambda_b > \lambda_a \to \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\to \exists T^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*

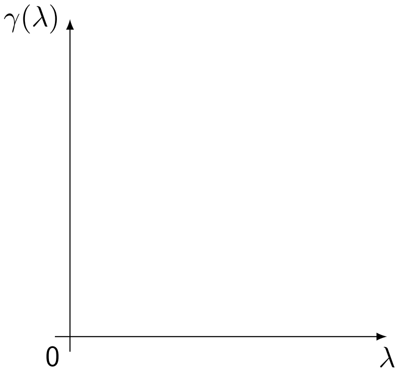
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*

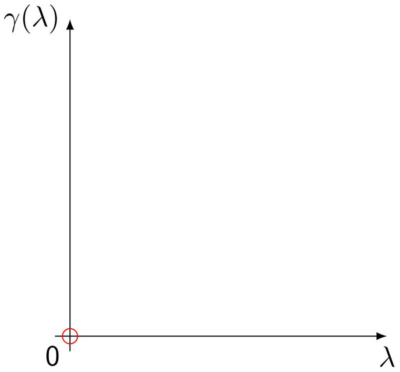
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

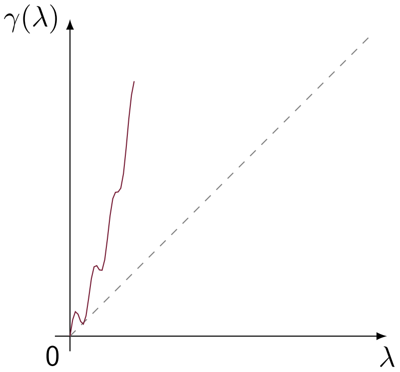
Assumptions

- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  - $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

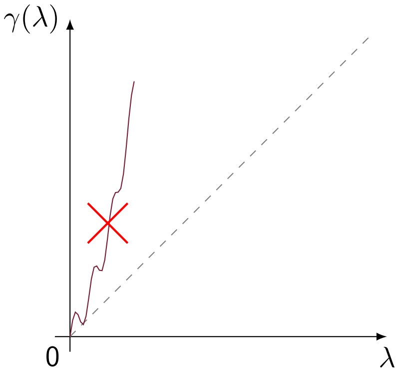
- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

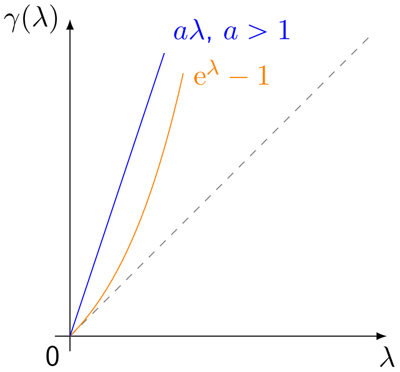
- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

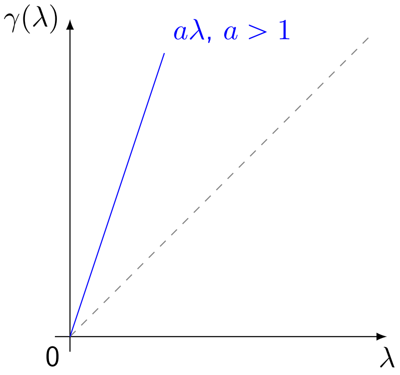
- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\rightarrow \exists T^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

- *Attacker is not naïve*
  - $\gamma(\lambda) = 0 \to \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Really greedy*
  $\lambda_b > \lambda_a \to \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If linear $\to \exists T^{-1}$

Example

4 distinct agents

- Agent 1 is non-cooperative
- It uses $\tilde{\lambda}_1 = \gamma_1(\lambda_1) = \tau_1 I \lambda_1$

- 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

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

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

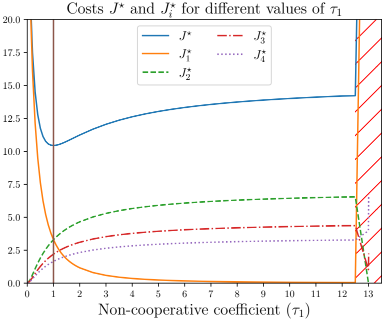
Example

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

Example



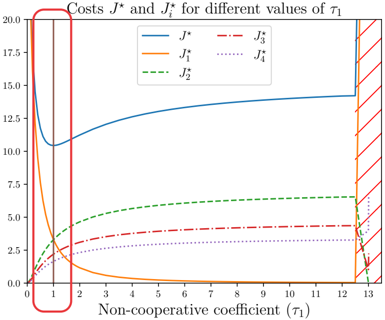
4 distinct agents

- 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

Example

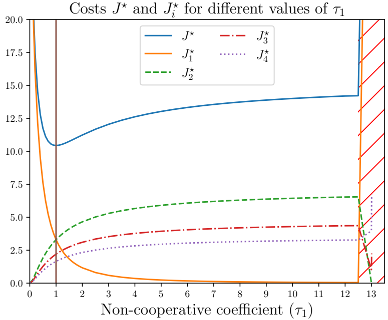


4 distinct agents

- 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

Example

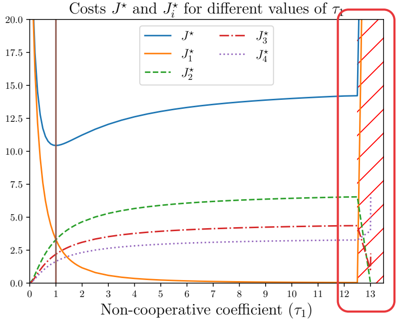


4 distinct agents

- 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

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

- But can we mitigate these effects?
- Yes! (At least in some cases)
- Let's start slowly

CentraleSupélec

- But can we mitigate these effects?
- Yes! (At least in some cases)
- Let's start slowly

- But can we mitigate these effects?
- Yes! (At least in some cases)
- Let's start slowly

- But can we mitigate these effects?
- Yes! (At least in some cases)
- Let's start slowly

- But can we mitigate these effects?
- Yes! (At least in some cases)
- Let's start slowly

Outline

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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

[2] Under some conditions

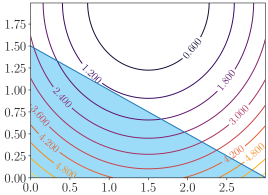
CentraleSupélec

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{U}_i^\star[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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



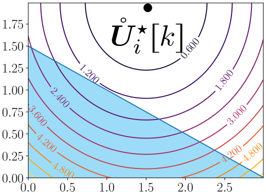
[2] Under some conditions

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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



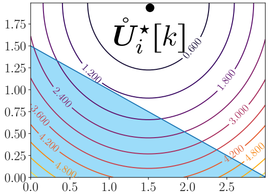
[2] Under some conditions

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^{\star}[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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



[2] Under some conditions

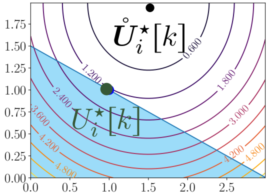
CentraleSupélec

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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



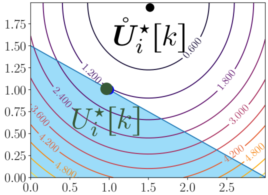
[2] Under some conditions

What are deprived systems?

TL;DR: Systems where all constraints are active

- Unconstrained Solution $\mathring{\boldsymbol{U}}_i^\star[k]$
- All constraints active = Scarcity
  - Solution projected onto boundary
  - Same as with equality constraints[2]

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



[2]Under some conditions ▸ see here

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

CentraleSupélec

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

CentraleSupélec

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat

- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

CentraleSupélec

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

CentraleSupélec

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

Deprived Systems

But why?

- No Scarcity
  - → All constraints satisfied
  - → No coordination needed
  - → No incentive to cheat
- Scarcity
  - → Competition
  - → Consensus/Compromise
  - → Agents may cheat

Deprived Systems

But why?

- No Scarcity
  - $\longrightarrow$ All constraints satisfied
  - $\longrightarrow$ No coordination needed
  - $\longrightarrow$ No incentive to cheat
- Scarcity
  - $\longrightarrow$ Competition
  - $\longrightarrow$ Consensus/Compromise
  - $\longrightarrow$ Agents may cheat

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i \mathring{\boldsymbol{U}}_i^\star[k] \leq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

(local parameters unknown by coordinator) $\left\{\right.$
- P_i is time invariant
- $\boldsymbol{s}_i[k]$ is time variant

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

(local parameters unknown by coordinator)
- P_i is time invariant
- $\boldsymbol{s}_i[k]$ is time variant

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

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

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

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

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

- Transform into equality constraints
- Solution is analytical

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

(local parameters unknown by coordinator) $\Big\{$

- P_i is time invariant
- $s_i[k]$ is time variant

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

- Transform into equality constraints
- Solution is analytical

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

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

$$\bar{\Gamma}_i\mathring{\boldsymbol{U}}_i^\star[k] \preceq \boldsymbol{\theta}_i[k]$$

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

- Transform into equality constraints
- Solution is analytical

(local parameters unknown by coordinator) $\left\{ \right.$
- P_i is time invariant
- $\boldsymbol{s}_i[k]$ is time variant

CentraleSupélec

Deprived Systems

Analysis (Negotiation)

- Update expression also becomes analytical
- If we input $\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$ → DTS with forced input

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

CentraleSupélec

Deprived Systems

Analysis (Negotiation)

- Update expression also becomes analytical
- If we input $\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$ → DTS with forced input

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)} \left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{\lambda}_j^{(p)} \right), \forall i \in \mathcal{M}$$

CentraleSupélec

Deprived Systems

Analysis (Negotiation)

- Update expression also becomes analytical
- If we input $\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k]$ $\rightarrow$ DTS with forced input

$$\boldsymbol{\theta}_i^{(p+1)} = \boldsymbol{\theta}_i^{(p)} + \rho^{(p)}\left(\boldsymbol{\lambda}_i^{(p)} - \frac{1}{M}\sum_{j=1}^{M}\boldsymbol{\lambda}_j^{(p)}\right), \forall i \in \mathcal{M}$$

CentraleSupélec

Deprived Systems

Analysis (Negotiation)

- Update expression also becomes analytical
- If we input $\boldsymbol{\lambda}_i[k] = -P_i\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i[k] \rightarrow$ DTS with forced input

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

see here

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

- New dynamics (Possibly Unstable)

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

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

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

- New dynamics (Possibly Unstable)

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

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

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

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

[1] Using Recursive Least Squares for example

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\widehat{\tilde{s}}_i[k]$ such as:

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

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

[1] Using Recursive Least Squares for example

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\widehat{\tilde{s}}_i[k]$ such as:

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

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

[1] Using Recursive Least Squares for example

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\hat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

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

[1] Using Recursive Least Squares for example

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

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

[1] Using Recursive Least Squares for example

CentraleSupélec

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

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

- Ok, but how can we estimate $\widehat{\tilde{P}}_i[k]$?

[1] Using Recursive Least Squares for example

Detection Mechanism

Assumption

We know nominal $\bar{P}_i$

- If we estimate[1] $\hat{P}_i[k]$ and $\hat{\tilde{\boldsymbol{s}}}_i[k]$ such as:

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

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

[1]Using Recursive Least Squares for example

CentraleSupélec

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

- We need to estimate $\hat{\tilde{P}}_i[k]$ and $\hat{\tilde{s}}_i[k]$ simultaneously
- Challenge: Estimation during negotiation fails
  - Update function couples θ_i^p and λ_i^p → low input excitation
- Solution: Send a random[3] sequence to increase excitation.

[3] A random signal has persistent excitation of any order ( )

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

- We need to estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{s}}_i[k]$ simultaneously
- Challenge: Estimation during negotiation fails
  - Update function couples θ_i^p and λ_i^p → low input excitation
- Solution: Send a random[3] sequence to increase excitation.

[3] A random signal has persistent excitation of any order ( )

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

- We need to estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously
- Challenge: Estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ → low input excitation
- Solution: Send a random[3] sequence to increase excitation.

[3] A random signal has persistent excitation of any order ( )

CentraleSupélec

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

- We need to estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously
- Challenge: Estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation
- Solution: Send a random[3] sequence to increase excitation.

[3] A random signal has persistent excitation of any order ( )

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

- We need to estimate $\widehat{\widetilde{P}}_i[k]$ and $\widehat{\widetilde{\boldsymbol{s}}}_i[k]$ simultaneously
- Challenge: Estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation
- Solution: Send a random[3] sequence to increase excitation.

[3] A random signal has persistent excitation of any order (Adaptive Control)

Classification of mitigation techniques

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

Classification of mitigation techniques

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

CentraleSupélec

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- We now 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

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- We now 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

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- We now 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

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

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- We now 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

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

- We now 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 $\boldsymbol{\lambda}_i$

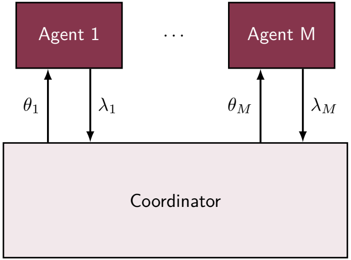
$$\overset{\text{rec}}{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i[k]^{-1}}\widehat{\tilde{\boldsymbol{s}}}_i[k]$$

- 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

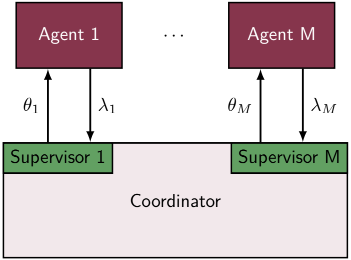


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

Complete Mechanism

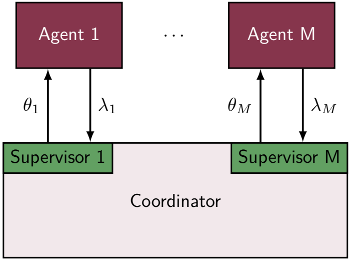


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

Complete Mechanism



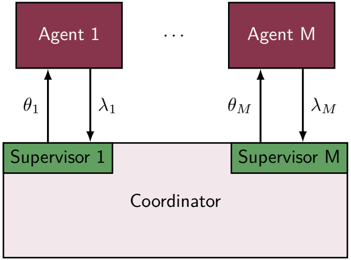
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

CentraleSupélec

Complete Mechanism

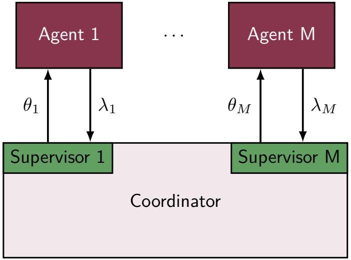


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

Complete Mechanism

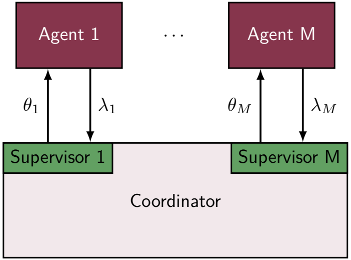


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

1. Detect which agents are non-cooperative
2. Reconstruct λ_i and use in negotiation

Complete Mechanism



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

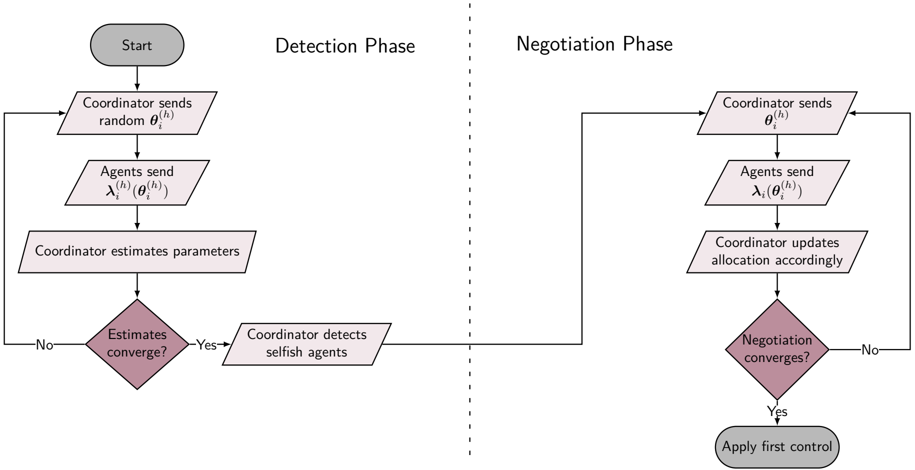
Two Phases

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

CentraleSupélec

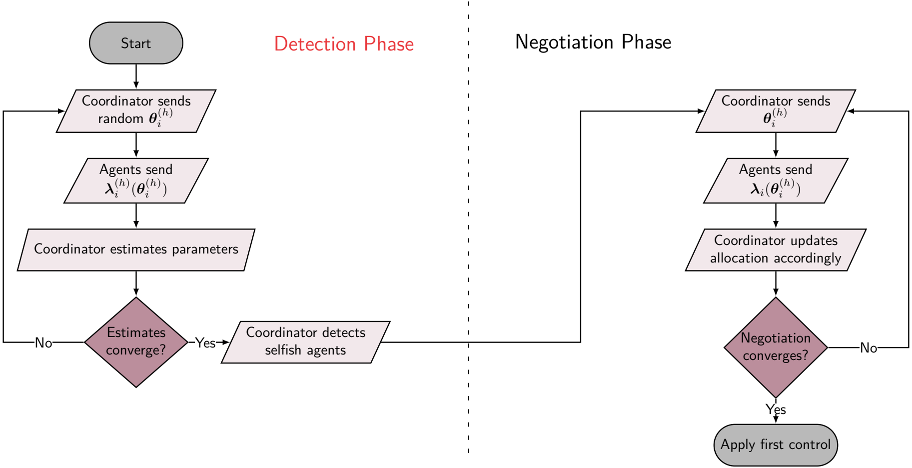
Complete algorithm

RPdMPC-DS



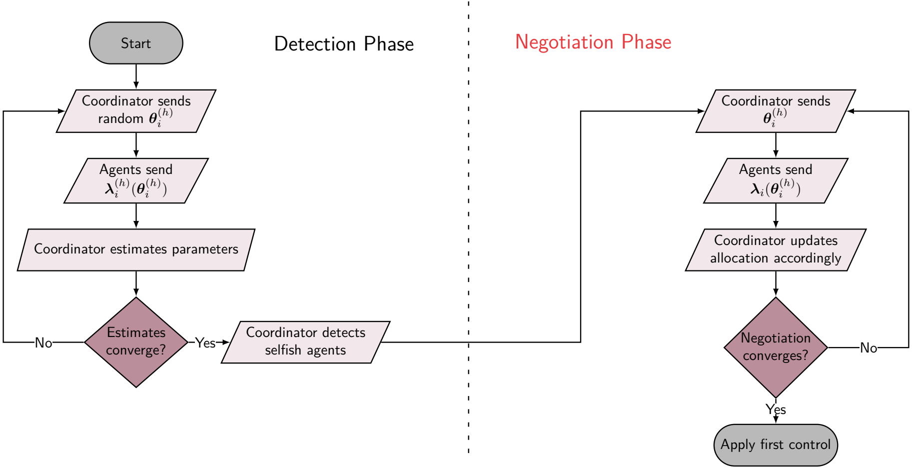
Complete algorithm

RPdMPC-DS



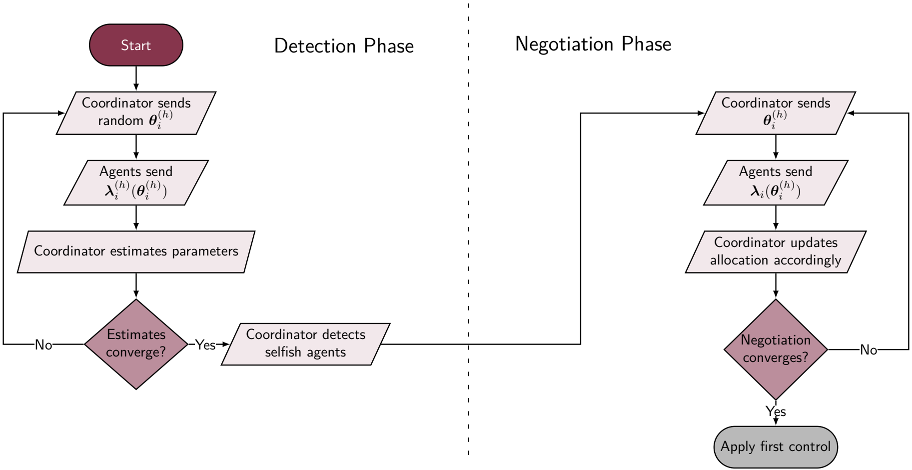
Complete algorithm

RPdMPC-DS



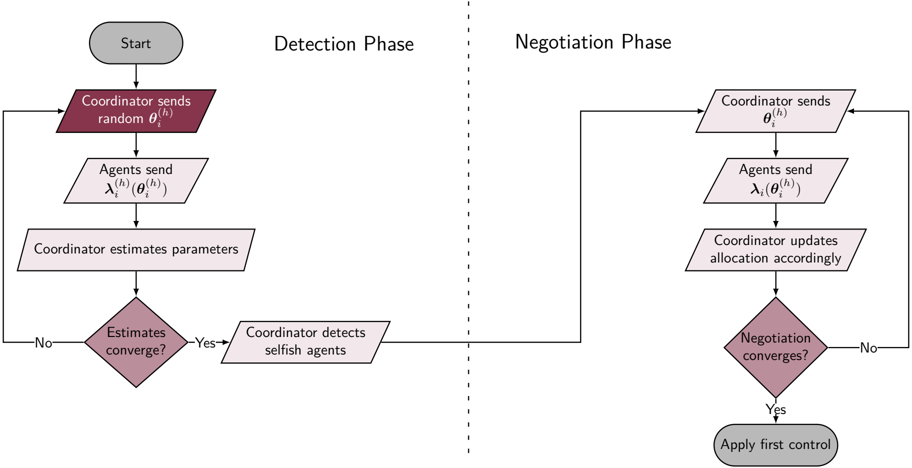
Complete algorithm

RPdMPC-DS



Complete algorithm

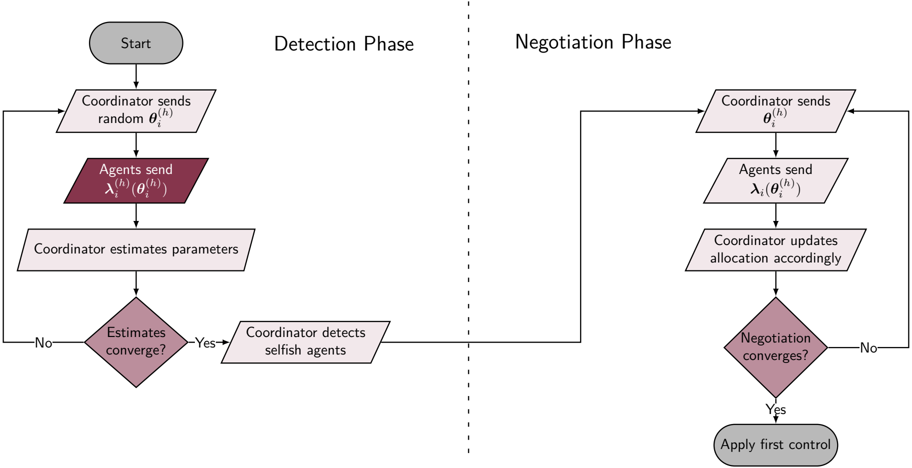
RPdMPC-DS



CentraleSupélec

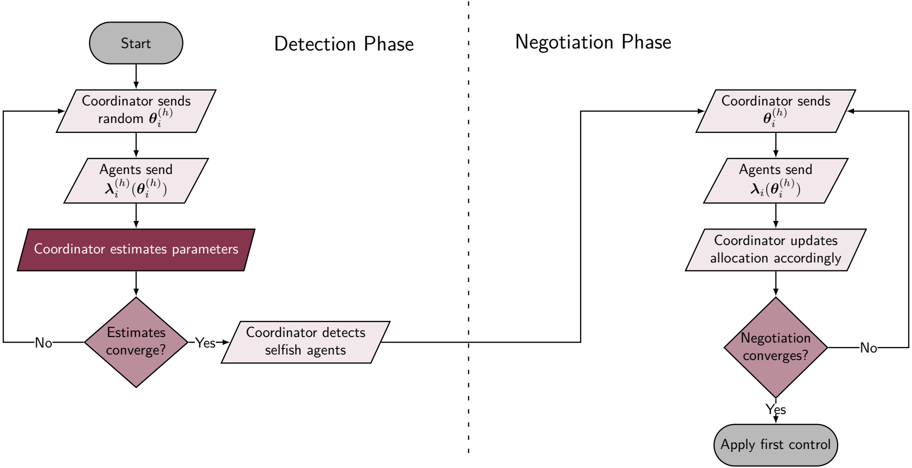
Complete algorithm

RPdMPC-DS



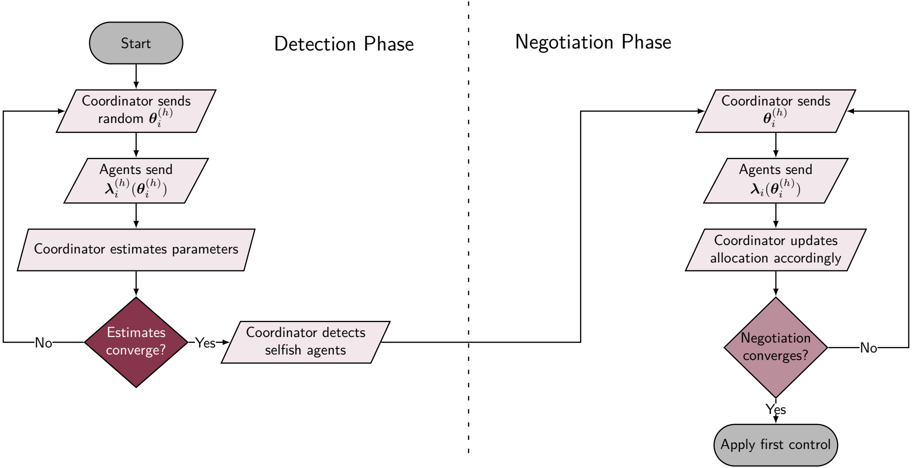
Complete algorithm

RPdMPC-DS



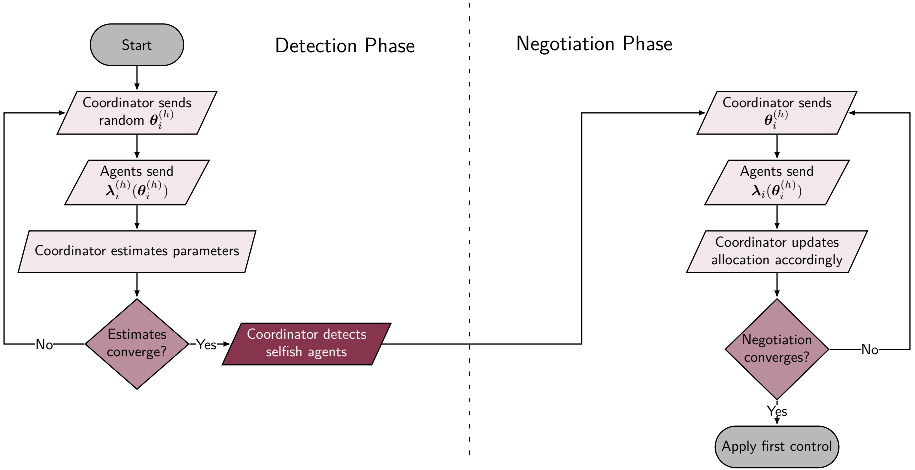
Complete algorithm

RPdMPC-DS



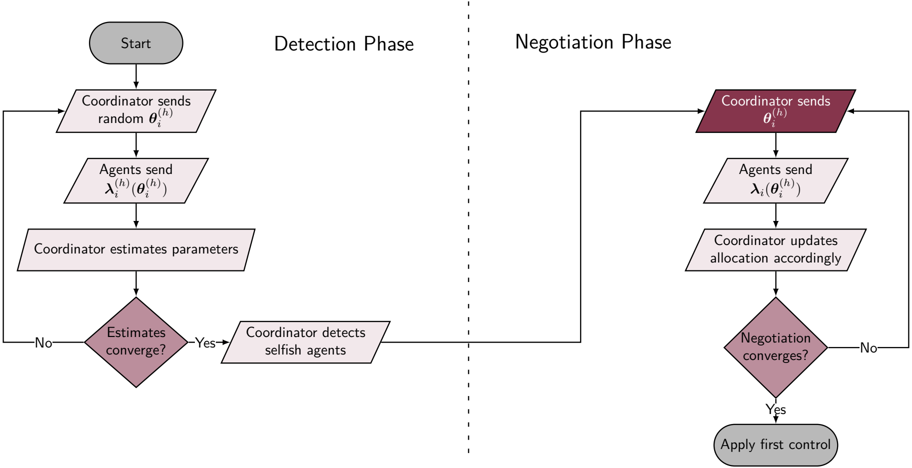
Complete algorithm

RPdMPC-DS



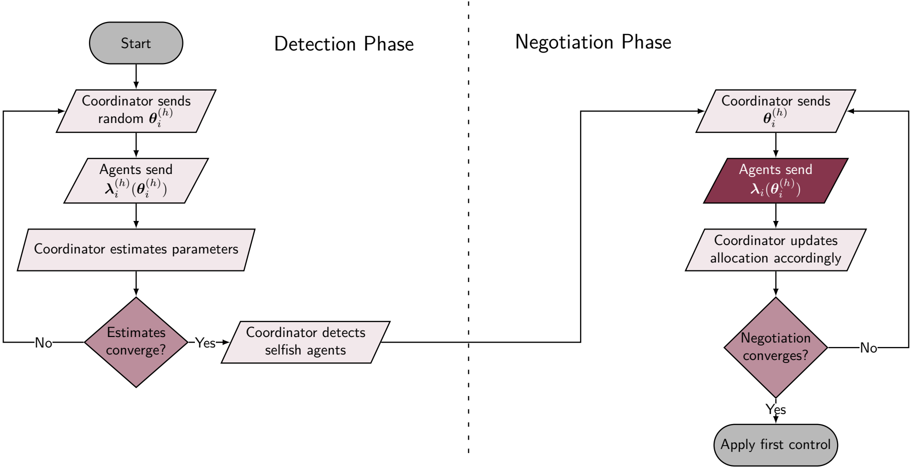
Complete algorithm

RPdMPC-DS



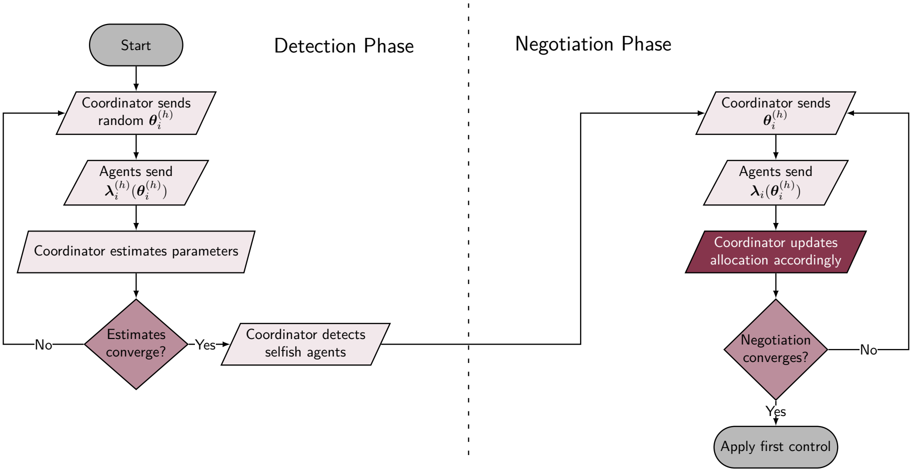
Complete algorithm

RPdMPC-DS



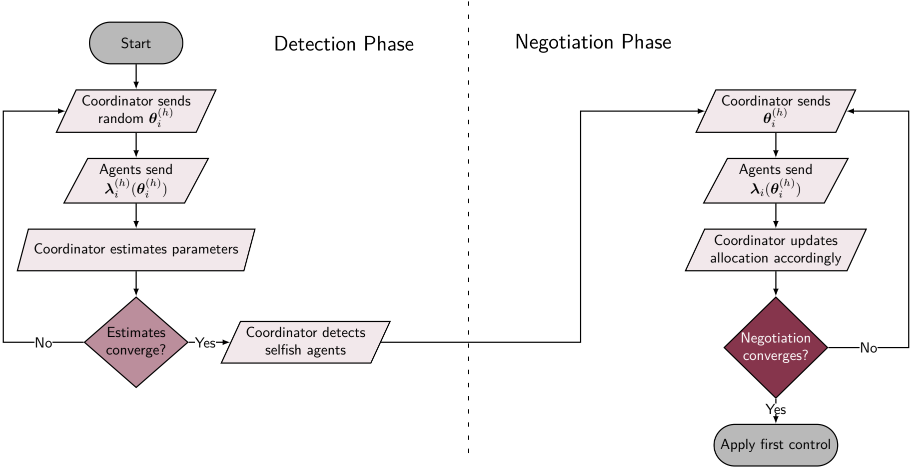
Complete algorithm

RPdMPC-DS



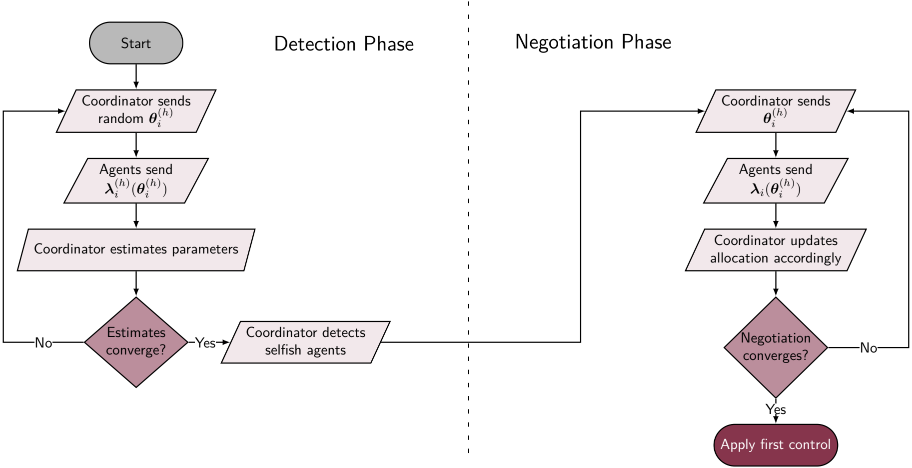
Complete algorithm

RPdMPC-DS

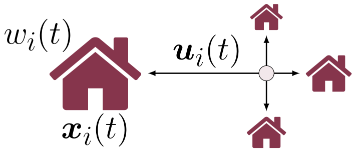


Complete algorithm

RPdMPC-DS



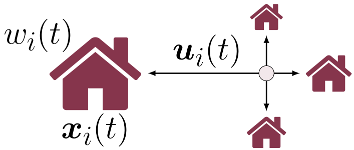
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

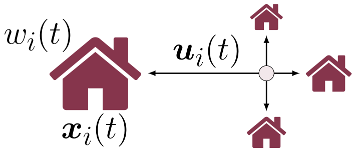
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

Example

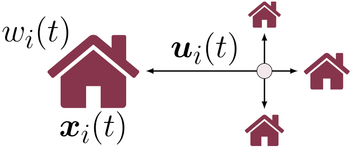


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Example

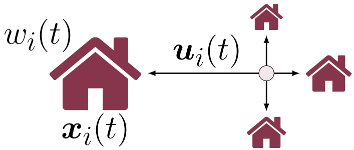


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Example

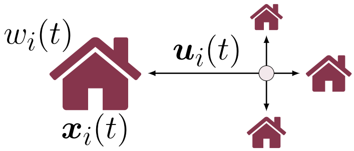


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Example

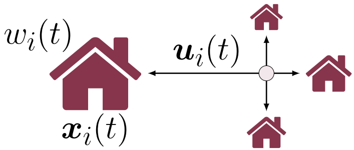


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Example

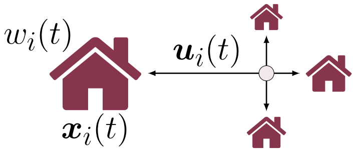


District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of 5h
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Example



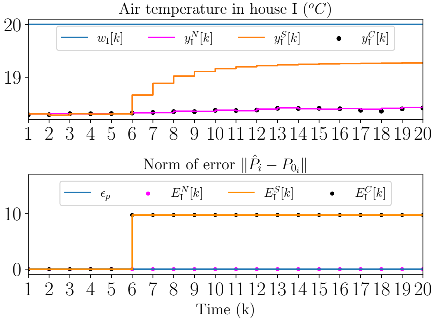
District Heating Network (4 Houses)

- Houses modeled using 3R-2C (monozone)
- Not enough power
- Period of $5\mathrm{h}$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-DS)

CentraleSupélec

Results

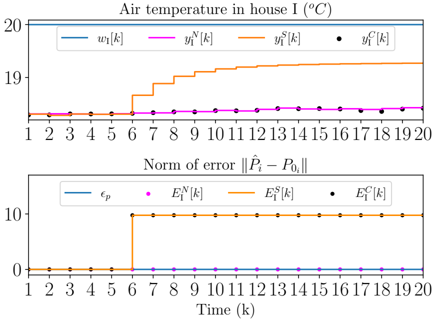
Temporal



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

Results

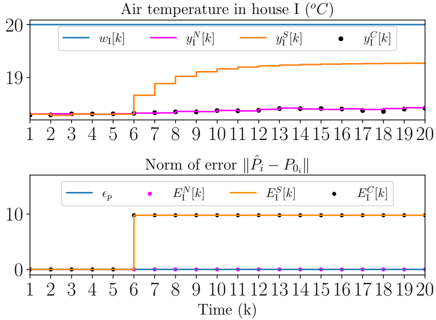
Temporal



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

Results

Temporal



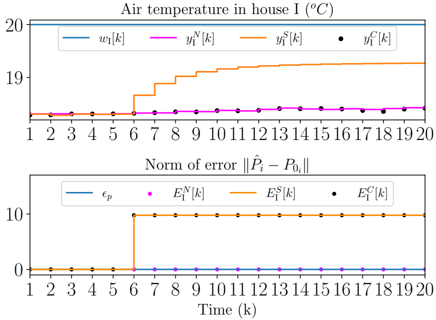
Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- S Temperature close to reference
- C Temperature close to N

CentraleSupélec

Results

Temporal



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

- S Temperature close to reference
- C Temperature close to N

Results

Costs

Objective functions J_i (% error)

Agent	Scenario N	Scenario S	Scenario C
I	299.5	190.8 (−36.3)	301.0 (0.0)
II	192.4	234.1 (21.7)	191.4 (−0.5)
III	305.9	359.1 (17.4)	305.9 (−0.0)
IV	297.5	349.9 (17.6)	297.2 (−0.1)
Global	1095.3	1133.9 (3.5)	1095.5 (0.0)

Results

Costs

Objective functions J_i (% error)

Agent	Scenario N	Scenario S	Scenario C
I	299.5	$190.8\ (-36.3)$	$301.0\ (0.0)$
II	192.4	$234.1\ (21.7)$	$191.4\ (-0.5)$
III	305.9	$359.1\ (17.4)$	$305.9\ (-0.0)$
IV	297.5	$349.9\ (17.6)$	$297.2\ (-0.1)$
Global	1095.3	$1133.9\ (3.5)$	$1095.5\ (0.0)$

Outline

CentraleSupélec

Relaxing scarcity assumption

- Let's relax the scarcity assumption
- And add some local constraints
- Similarly we have the local problems and update

$$\begin{aligned}\underset{\boldsymbol{U}[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}[k]\right\|_H^2 + \boldsymbol{f}[k]^T\boldsymbol{U}[k] \\ \text{subject to} \quad & \bar{\Gamma}\boldsymbol{U}[k] \leq \boldsymbol{U}_{\max} \\ & \boldsymbol{U}[k] \in \mathcal{U}\end{aligned}$$

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

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

CentraleSupélec

Relaxing scarcity assumption

- Let's relax the scarcity assumption
- And add some local constraints
- Similarly we have the local problems and update

$$\begin{aligned} \underset{\boldsymbol{U}[k]}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{U}[k]\|_H^2 + \boldsymbol{f}[k]^T\boldsymbol{U}[k] \\ \text{subject to} \quad & \bar{\Gamma}\boldsymbol{U}[k] \preceq \boldsymbol{U}_{\max} \\ & \boldsymbol{U}[k] \in \mathcal{U} \end{aligned}$$

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

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

CentraleSupélec

Relaxing scarcity assumption

- Let's relax the scarcity assumption
- And add some local constraints
- Similarly we have the local problems and update

$$\begin{aligned} \underset{\boldsymbol{U}[k]}{\text{minimize}} \quad & \tfrac{1}{2}\left\|\boldsymbol{U}[k]\right\|_H^2 + \boldsymbol{f}[k]^T\boldsymbol{U}[k] \\ \text{subject to} \quad & \bar{\Gamma}\boldsymbol{U}[k] \preceq \boldsymbol{U}_{\mathsf{max}} \\ & \boldsymbol{U}[k] \in \mathcal{U} \end{aligned}$$

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

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

CentraleSupélec

Relaxing scarcity assumption

- Let's relax the scarcity assumption
- And add some local constraints
- Similarly we have the local problems and update

$$\begin{array}{ll} \underset{\boldsymbol{U}[k]}{\text{minimize}} & \frac{1}{2}\left\|\boldsymbol{U}[k]\right\|_H^2 + \boldsymbol{f}[k]^T\boldsymbol{U}[k] \\ \text{subject to} & \bar{\Gamma}\boldsymbol{U}[k] \preceq \boldsymbol{U}_{\max} \\ & \boldsymbol{U}[k] \in \mathcal{U} \end{array}$$

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

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

Relaxing scarcity assumption

- Let's relax the scarcity assumption
- And add some local constraints
- Similarly we have the local problems and update

$$\begin{aligned} \underset{\boldsymbol{U}[k]}{\text{minimize}} \quad & \frac{1}{2}\left\|\boldsymbol{U}[k]\right\|_{H}^{2} + \boldsymbol{f}[k]^{T}\boldsymbol{U}[k] \\ \text{subject to} \quad & \bar{\Gamma}\boldsymbol{U}[k] \preceq \boldsymbol{U}_{\max} \\ & \boldsymbol{U}[k] \in \mathcal{U} \end{aligned}$$

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

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

CentraleSupélec

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{n}_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{\left(2^{n_{\text{ineq}}}-1\right)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{\left(2^{n_{\text{ineq}}}-1\right)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{2^{n_{\text{ineq}}}-1}_{\boldsymbol{\lambda}_i} \end{cases}$$

Increasingly Sparse

For n_{ineq} constraints $\longrightarrow 2^{n_{\text{ineq}}}$ permutations

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}_{\boldsymbol{\lambda}_i}^{n} \\ \vdots & \vdots \\ -P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}_{\boldsymbol{\lambda}_i}^{2^{n_{\text{ineq}}}-1} \end{cases}$$

Increasingly Sparse

For n_{ineq} constraints $\longrightarrow$ $2^{n_{\text{ineq}}}$ permutations

CentraleSupélec

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{n}_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{2^{n_{\text{ineq}}}-1}_{\boldsymbol{\lambda}_i} \end{cases}$$

Increasingly Sparse

For n_{ineq} constraints $\longrightarrow$ $2^{n_{\text{ineq}}}$ permutations

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^n_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(2^{n_\text{ineq}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_\text{ineq}}-1)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{2^{n_\text{ineq}}-1}_{\boldsymbol{\lambda}_i} \end{cases}$$

Increasingly Sparse

For n_ineq constraints $\longrightarrow$ 2^{n_ineq} permutations

CentraleSupélec

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^n_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{2^{n_{\text{ineq}}}-1}_{\boldsymbol{\lambda}_i} \end{cases}$$

Increasingly Sparse

For n_{ineq} constraints $\rightarrow 2^{n_{\text{ineq}}}$ permutations

Analyzing System

Solution for $\lambda_i[k]$

Instead of having

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

Now we have

$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{n}_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(2^{n_{\mathsf{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\mathsf{ineq}}}-1)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^{2^{n_{\mathsf{ineq}}}-1}_{\boldsymbol{\lambda}_i} \end{cases} \Bigg\downarrow$$

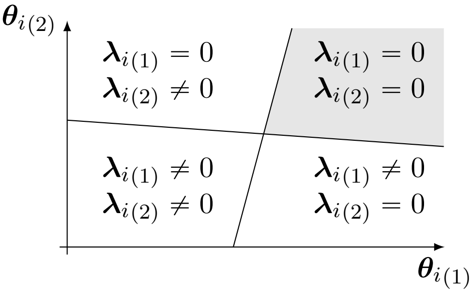
Increasingly Sparse

For n_{ineq} constraints $\rightarrow$ $2^{n_{\mathsf{ineq}}}$ permutations

CentraleSupélec

Analyzing System

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



Two constraints partitioning $\boldsymbol{\theta}_i$ solution space.

CentraleSupélec

Analyzing System

Negotiation

$$\mathrm{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)}+\rho^{(p)}\boldsymbol{\lambda}[k]^{(p)}) = \underset{\boldsymbol{x}}{\operatorname{argmin}} \left\{ \begin{array}{ll} \underset{\boldsymbol{U}[k]}{\text{minimize}} & \left\|\boldsymbol{x} - \boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)}\right\| \\ \text{subject to} & I_c^M \boldsymbol{x} \preceq \boldsymbol{U}_{\max} : \boldsymbol{\mu} \end{array} \right\}$$

CentraleSupélec

Analyzing System

Negotiation (Continued)

$$\boldsymbol{\theta}[k]^{(p+1)} = \begin{cases} \boldsymbol{x}_0 + I_c^{M(0)}\left[-P_{\boldsymbol{\mu}}^{(0)}\boldsymbol{U}_{max} + \boldsymbol{s}_{\boldsymbol{\mu}}^{(0)}[k]\right], & \text{if } \boldsymbol{x}_0 \in \mathcal{R}_{\boldsymbol{\mu}}^{0} \\ \vdots & \vdots \\ \boldsymbol{x}_0, & \text{if } \boldsymbol{x}_0 \in \mathcal{R}_{\boldsymbol{\mu}}^{2^c-1} \end{cases}$$

$$\text{where } \boldsymbol{x}_0 = \boldsymbol{\theta}[k]^{(p)} + \rho^{(p)}\boldsymbol{\lambda}[k]^{(p)}$$

Analyzing System

Conclusion

$$\bar{\epsilon} = \overbrace{2^{c}}^{\text{regions in projection}} \times \underbrace{2^{n_{\text{ineq}}} \times \cdots \times 2^{n_{\text{ineq}}}}_{M\times\text{regions in each }\boldsymbol{\lambda}_i} = 2^{c+Mn_{\text{ineq}}}$$

Mitigation

- Ideal

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

- $P_i^{(0)}$ only invertible

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

  - But how to force scarcity? Artificial Scarcity

Mitigation

- Ideal

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

- $P_i^{(0)}$ only invertible

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

  - But how to force scarcity? Artificial Scarcity

Mitigation

- Ideal

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

- $P_i^{(0)}$ only invertible

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

- But how to force scarcity? Artificial Scarcity

CentraleSupélec

Mitigation

- Ideal

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

- $P_i^{(0)}$ only invertible

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

  - But how to force scarcity? Artificial Scarcity

CentraleSupélec

Mitigation

- Ideal

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

- $P_i^{(0)}$ only invertible

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

  - But how to force scarcity? Artificial Scarcity

CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active

Artificial Scarcity

Who is it? Who is it?

- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active

Artificial Scarcity

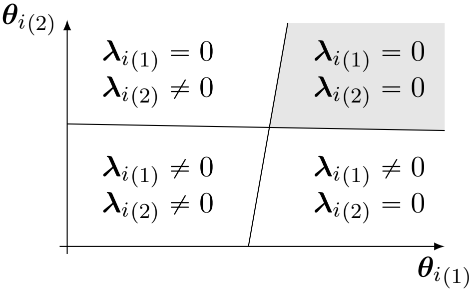
Who is it? Who is it?

- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active

Artificial Scarcity

Who is it? Who is it?

- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active

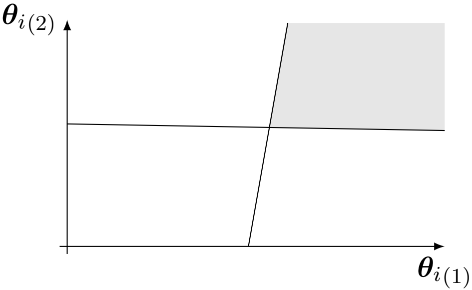


CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

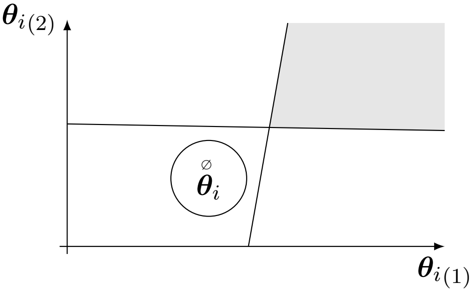
- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active



Artificial Scarcity

Who is it? Who is it?

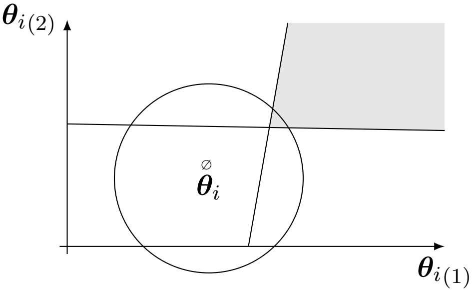
- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active



Artificial Scarcity

Who is it? Who is it?

- Since we control $\boldsymbol{\theta}_i$ let's choose one where all constraints are active



Expectation Maximization

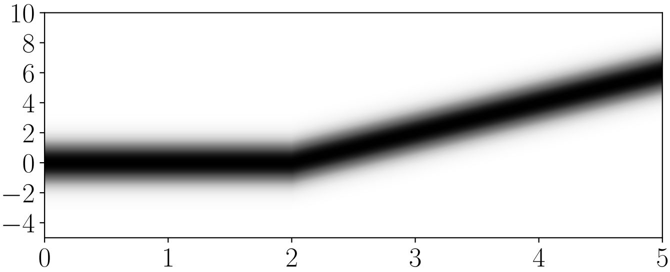


Figure 1: Gaussian Mixture for a 1D PWA function with 2 modes.

Expectation Maximization

Algorithm

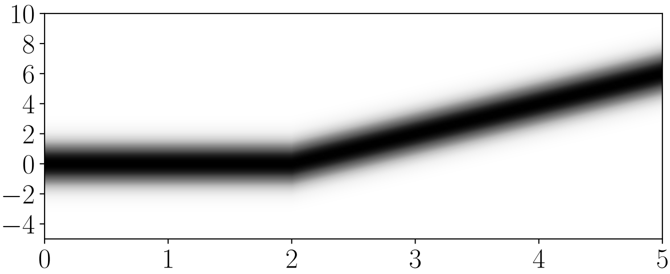


Figure 2: Gaussian Mixture for a 1D PWA function with 2 modes.

Detection

Same same

- Error $E_i^{(0)}[k] = \left\| \widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F$
- Create threshold $\epsilon_{P_i^{(0)}}$
- Indicator $\mathfrak{d}_i \in \{0, 1\}$ detects the attack in agent i.
- $\mathfrak{d}_i^{(0)} = \mathbb{1}_{\{E_i^{(0)}[k] \geqslant \epsilon_{P_i^{(0)}}\}}$

Detection

Same same

- Error $E_i^{(0)}[k] = \left\| \widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F$
- Create threshold $\epsilon_{P_i^{(0)}}$
- Indicator $\eth_i \in \{0, 1\}$ detects the attack in agent i.
- $\eth_i^{(0)} = \mathbb{1}_{\{E_i^{(0)}[k] \geqslant \epsilon_{P_i^{(0)}}\}}$

Detection

Same same

- Error $E_i^{(0)}[k] = \left\|\widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)}\right\|_F$
- Create threshold $\epsilon_{P_i^{(0)}}$
- Indicator $\mathfrak{d}_i \in \{0, 1\}$ detects the attack in agent i.
- $\mathfrak{d}_i^{(0)} = \mathbb{1}_{\{E_i^{(0)}[k] \geqslant \epsilon_{P_i^{(0)}}\}}$

Detection

Same same

- Error $E_i^{(0)}[k] = \left\| \widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i{}^{(0)} \right\|_F$
- Create threshold $\epsilon_{P_i^{(0)}}$
- Indicator $\mathfrak{d}_i \in \{0, 1\}$ detects the attack in agent i.
- $\mathfrak{d}_i^{(0)} = \mathbb{1}_{\{E_i^{(0)}[k] \geqslant \epsilon_{P_i^{(0)}}\}}$

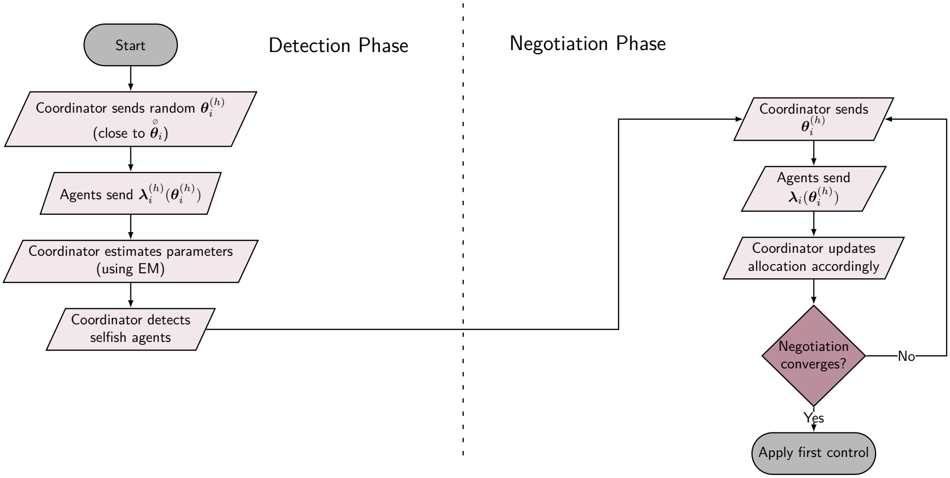
Detection

Same same

- Error $E_i^{(0)}[k] = \left\| \widehat{\widetilde{P}}_i^{(0)}[k] - \bar{P}_i^{(0)} \right\|_F$
- Create threshold $\epsilon_{P_i^{(0)}}$
- Indicator $\mathfrak{d}_i \in \{0,1\}$ detects the attack in agent i.
- $\mathfrak{d}_i^{(0)} = \mathbb{1}_{\{E_i^{(0)}[k] \geqslant \epsilon_{P_i^{(0)}}\}}$

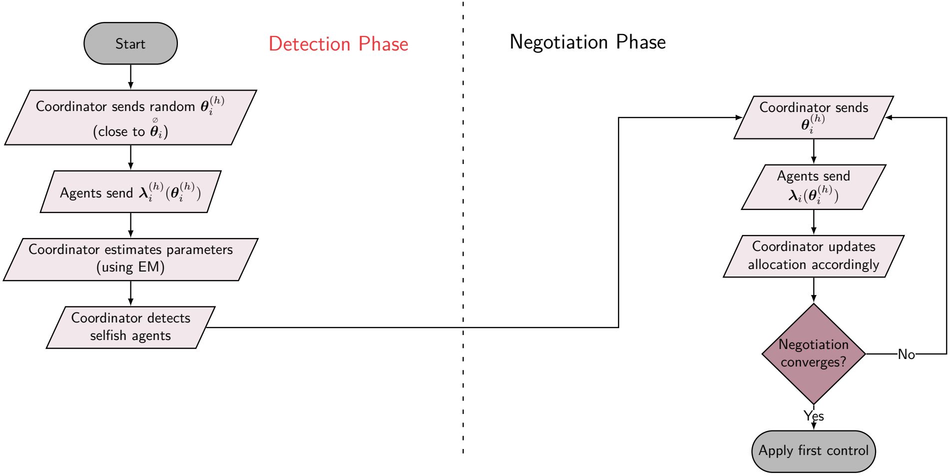
Complete algorithm

RPdMPC-AS Refaire



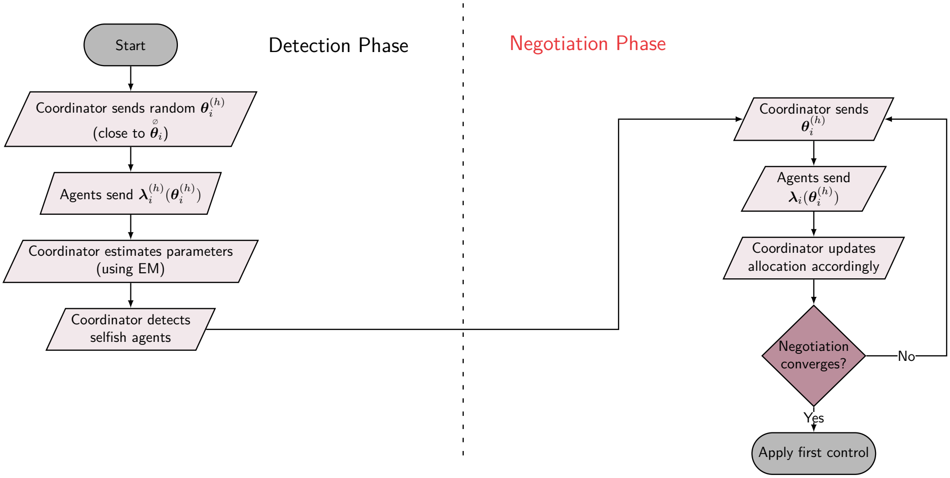
Complete algorithm

RPdMPC-AS Refaire



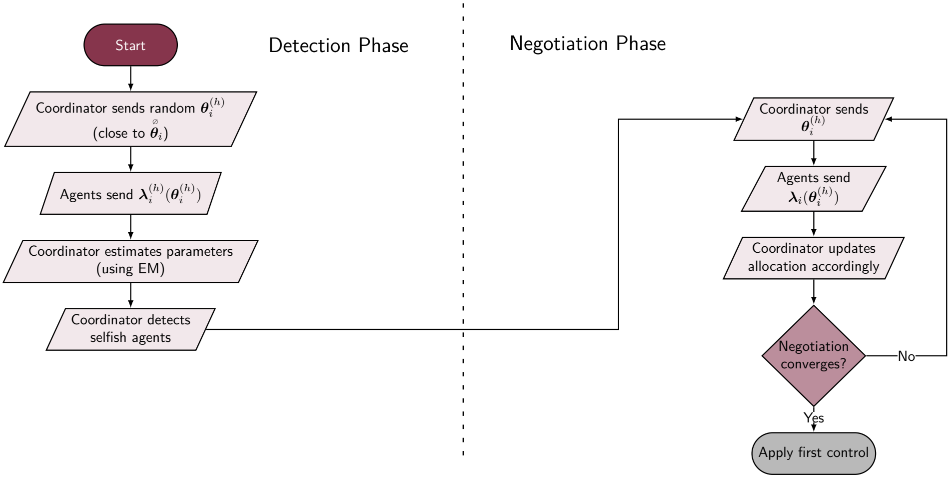
Complete algorithm

RPdMPC-AS Refaire



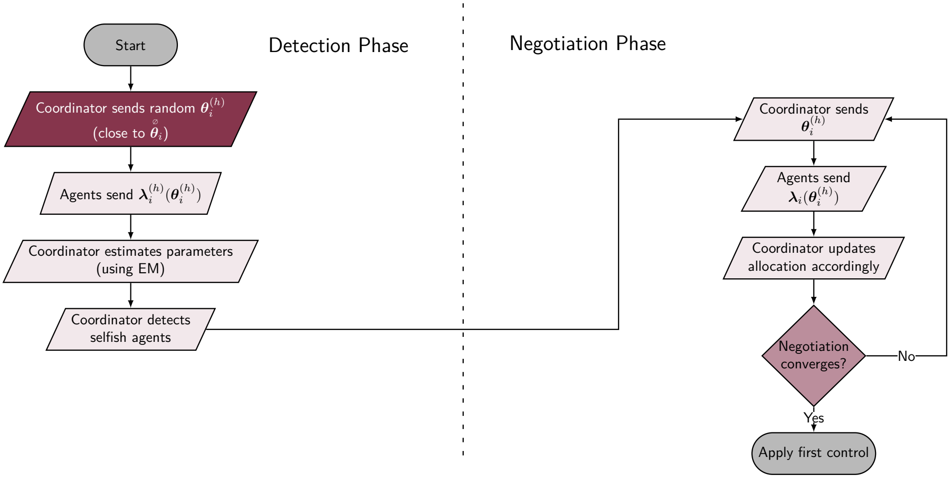
Complete algorithm

RPdMPC-AS Refaire



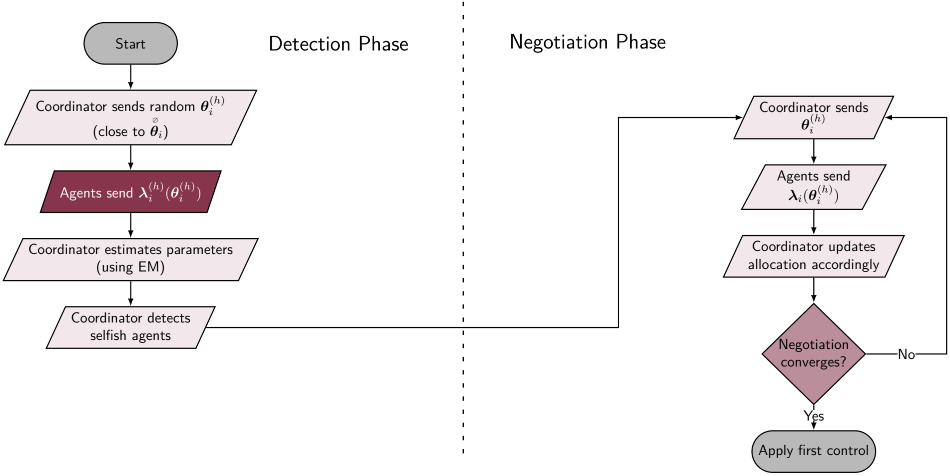
Complete algorithm

RPdMPC-AS Refaire



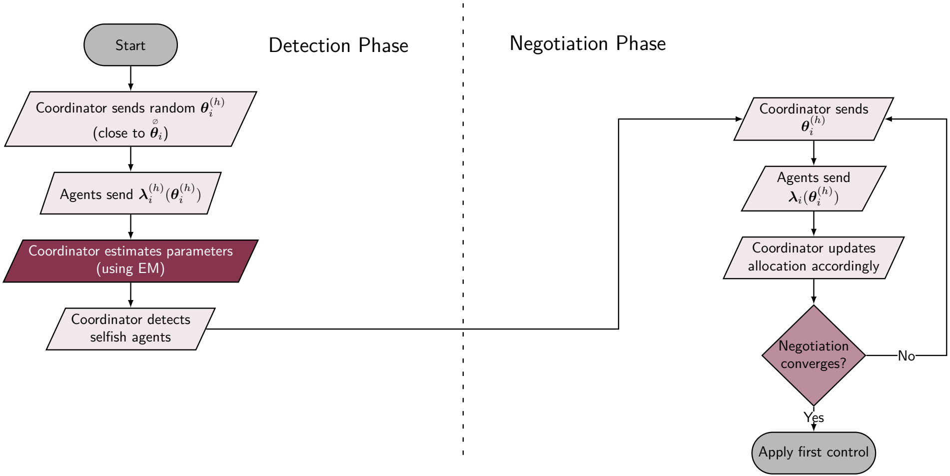
Complete algorithm

RPdMPC-AS Refaire



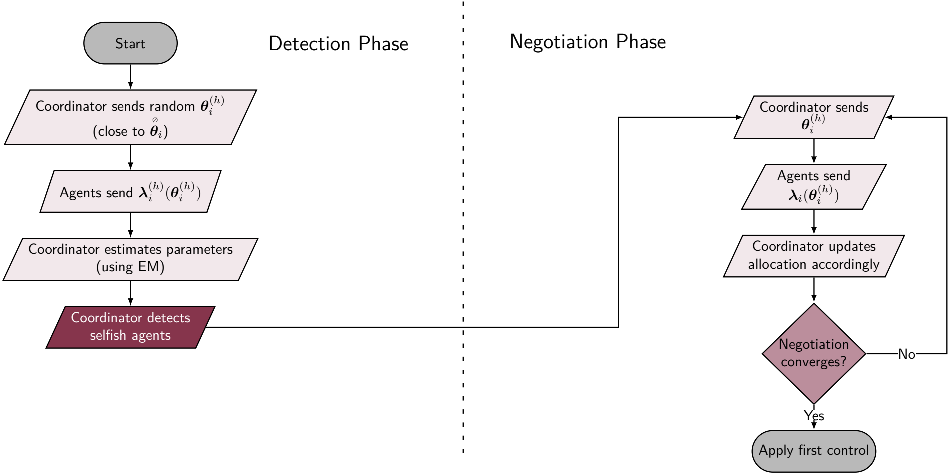
Complete algorithm

RPdMPC-AS Refaire



Complete algorithm

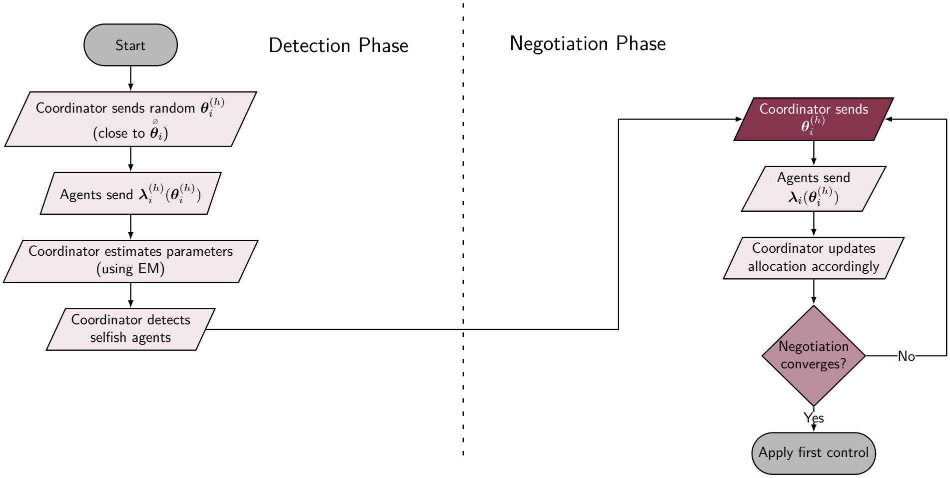
RPdMPC-AS Refaire



CentraleSupélec

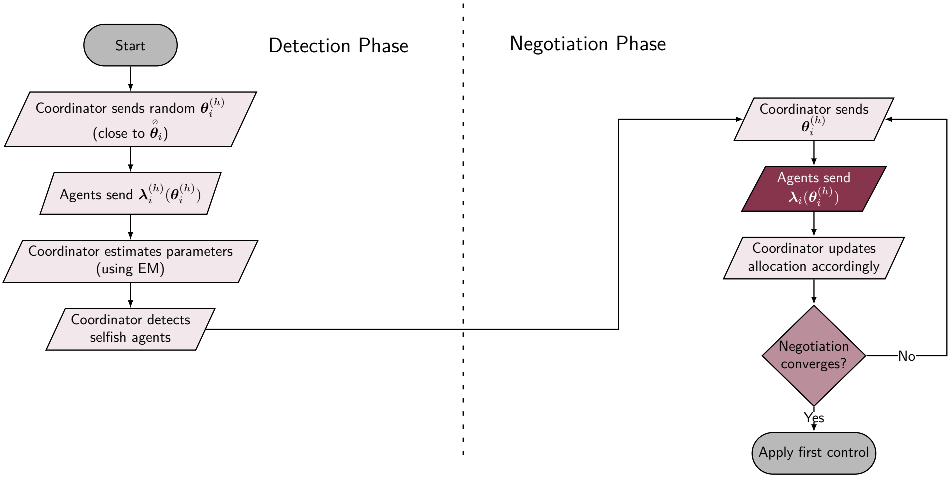
Complete algorithm

RPdMPC-AS Refaire



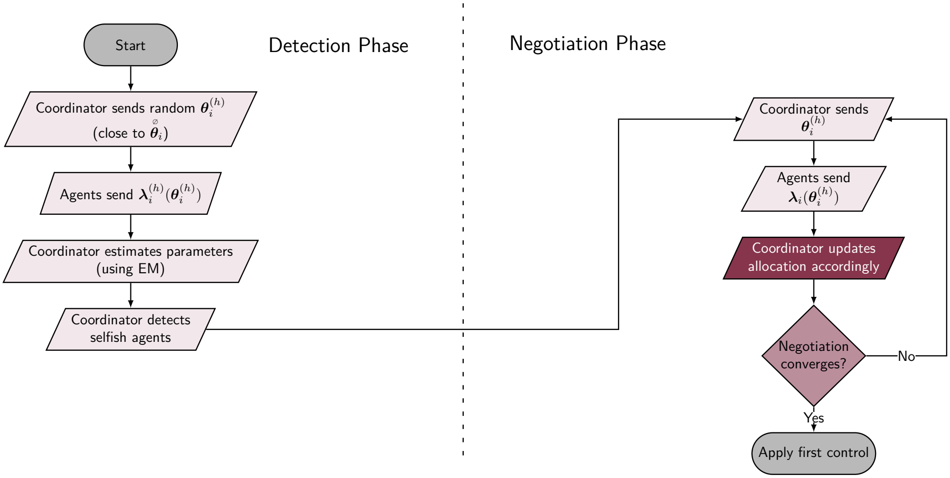
Complete algorithm

RPdMPC-AS Refaire



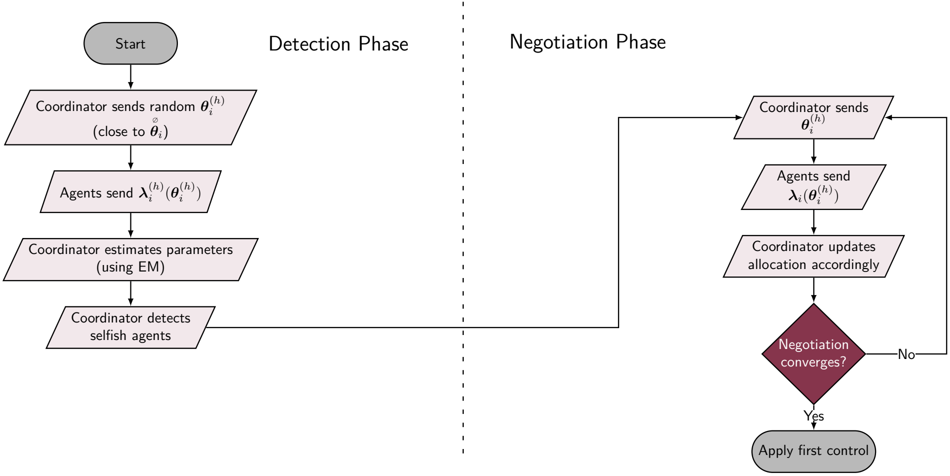
Complete algorithm

RPdMPC-AS Refaire



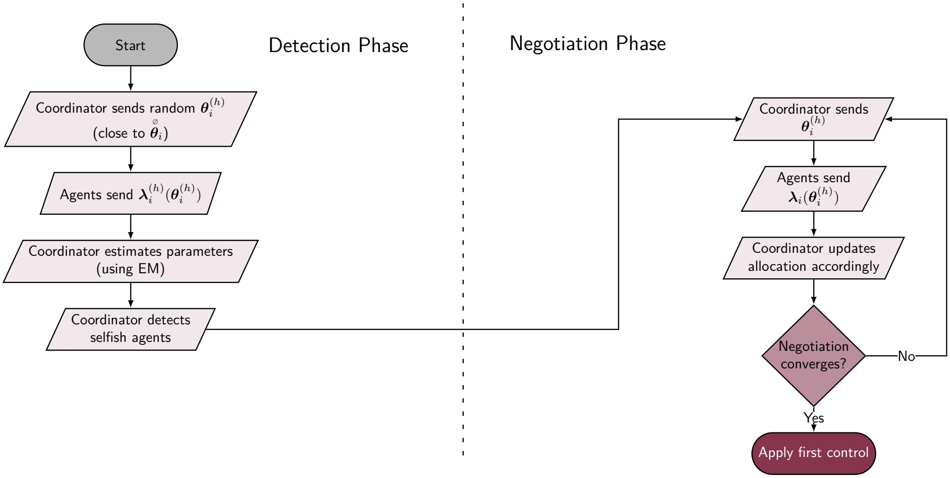
Complete algorithm

RPdMPC-AS Refaire



Complete algorithm

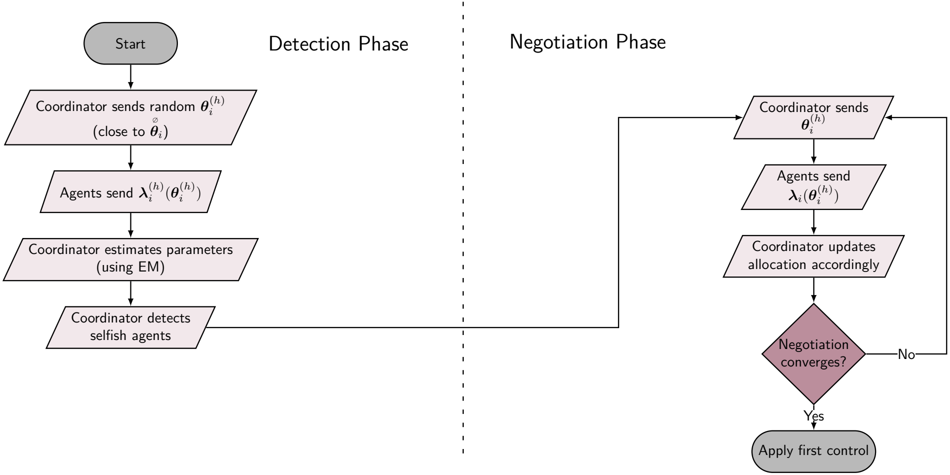
RPdMPC-AS Refaire



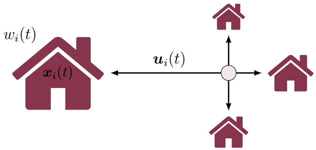
CentraleSupélec

Complete algorithm

RPdMPC-AS Refaire



Example

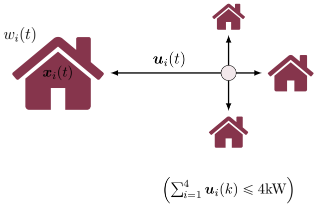


$$\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\text{kW}\right)$$

District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- Not enough power
- Period of 5h ($T_s = 0.25\text{h}$)
- 3 scenarios
  - (N) Nominal
  - (C) Agent I cheats (dMPC)
  - (S) Agent I cheats (RPdMPC-AS)
- $T_I = \begin{bmatrix} 14.43288267 & 0. & 0. & 0. \\ 0. & 13.4590903 & 0. & 0. \\ 0. & 0. & 6.93065061 & 0. \\ 0. & 0. & 0. & 3.4447393 \end{bmatrix}$

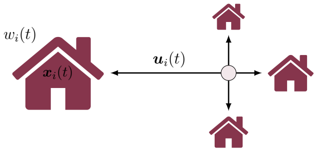
Example



District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- ~~Not enough power~~ (Change $(\boldsymbol{x}_0, \boldsymbol{w}_0)$)
- Period of 5h $(T_s = 0.25\text{h})$
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-AS)
- $T_I = \begin{bmatrix} 14.43288267 & 0. & 0. & 0. \\ 0. & 13.4590903 & 0. & 0. \\ 0. & 0. & 6.93065061 & 0. \\ 0. & 0. & 0. & 3.4447393 \end{bmatrix}$

Example



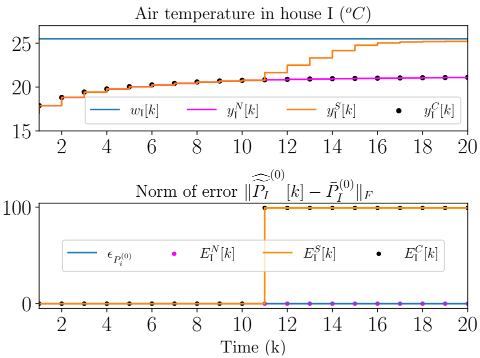
$\left(\sum_{i=1}^{4} \boldsymbol{u}_i(k) \leqslant 4\mathrm{kW}\right)$

District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- ~~Not enough power~~ (Change $(\boldsymbol{x}_0, \boldsymbol{w}_0)$)
- Period of $5\mathrm{h}$ ($T_s = 0.25\mathrm{h}$)
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-AS)
- $T_I = \begin{bmatrix} 14.43288267 & 0. & 0. & 0. \\ 0. & 13.4590903 & 0. & 0. \\ 0. & 0. & 6.93065061 & 0. \\ 0. & 0. & 0. & 3.4447393 \end{bmatrix}$

Results

Temporal

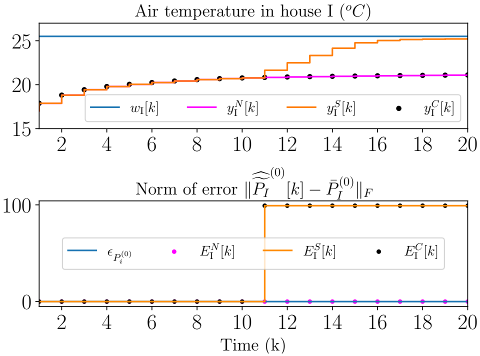


Temperature in house I and the variable $E_I(k)$ for different scenarios.
N Nominal, S Selfish behavior, C Selfish + Correction

CentraleSupélec

Results

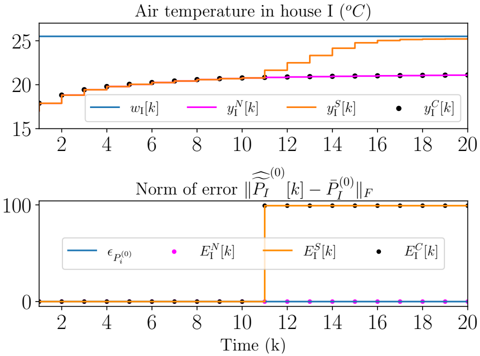
Temporal



Temperature in house I and the variable $E_I(k)$ for different scenarios.
N Nominal, S Selfish behavior, C Selfish + Correction

Results

Temporal

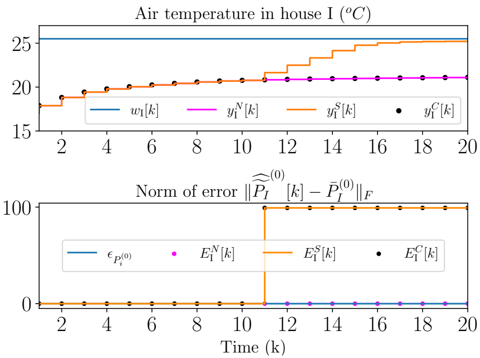


Temperature in house I and the variable $E_I(k)$ for different scenarios.
N Nominal, S Selfish behavior, C Selfish + Correction

CentraleSupélec

Results

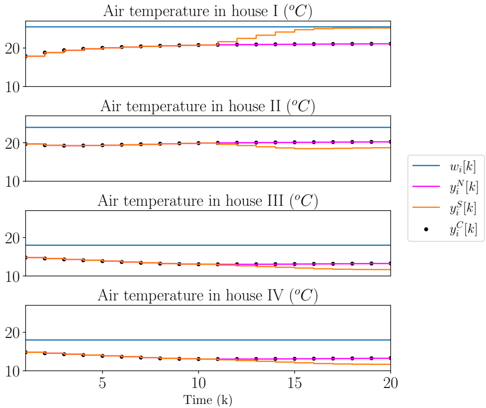
Temporal



Temperature in house I and the variable $E_I(k)$ for different scenarios.
N Nominal, S Selfish behavior, C Selfish + Correction

Results

Temporal (Continued)

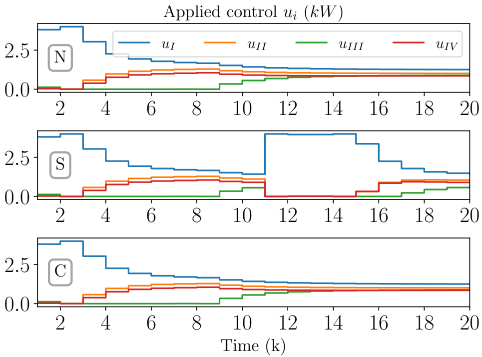


Air temperature in all houses for different scenarios.

N Nominal, S Selfish behavior, C Selfish + Correction

Results

Control



Control applied in all houses for different scenarios.
N Nominal, S Selflish behavior, C Selfish + Correction

Results

Costs

Objective functions J_i (% error).

Agent	Scenario N	Scenario S	Scenario C
I	19868.2	12618.5 (−36.5)	19868.2 (−0.0)
II	13784.5	18721.1 (35.8)	13784.5 (0.0)
III	17276.0	22324.9 (29.2)	17276.1 (0.0)
IV	10086.0	13872.4 (37.5)	10086.0 (0.0)
Global	61014.7	67536.9 (10.7)	61014.7 (−0.0)

Summary

- Vulnerabilities of Primal decomposition dMPC
- Resilient strategy for 2 kinds of systems
  - Deprived systems (where demand is greater than total resources)
  - Systems with possible artificial scarcity (sensible optimal demand)

CentraleSupélec

Summary

- Vulnerabilities of Primal decomposition dMPC
- Resilient strategy for 2 kinds of systems
  - Deprived systems (where demand is greater than total resources)
  - Systems with possible artificial scarcity (sensible optimal demand)

CentraleSupélec

Summary

- Vulnerabilities of Primal decomposition dMPC
- Resilient strategy for 2 kinds of systems
  - Deprived systems (where demand is greater than total resources)
  - Systems with possible artificial scarcity (sensible optimal demand)

Summary

- Vulnerabilities of Primal decomposition dMPC
- Resilient strategy for 2 kinds of systems
  - Deprived systems (where demand is greater than total resources)
  - Systems with possible artificial scarcity (sensible optimal demand)

CentraleSupélec

Summary

- Vulnerabilities of Primal decomposition dMPC
- Resilient strategy for 2 kinds of systems
  - Deprived systems (where demand is greater than total resources)
  - Systems with possible artificial scarcity (sensible optimal demand)

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

CentraleSupélec

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

CentraleSupélec

Future Directions

- Study of robustness + noise
- Partial reconstruction of cheating matrix
- Resilient strategy with soft constraints
- Recursive EM (or alternative)
- ...

Thank you!

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

Contact
rafael.accacio.nogueira@gmail.com

Temporary page!

LaTeX was unable to guess the total number of pages correctly. As there was some unprocessed data that should have been added to the final page this extra page has been added to receive it.
If you rerun the document (without altering it) this surplus page will go away, because LaTeX now knows how many pages to expect for this document.