Security of distributed Model Predictive Control under False Data injection

Rafael Accácio NOGUEIRA

2022-12-08

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)

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)

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

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 $\rightarrow$ 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

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

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 (QoS, technical restrictions, ...)

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 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, ...)

CentraleSupélec

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, ...)

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

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, ...)

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

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, ...)

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

CentraleSupélec

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, ...)

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

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, ...)

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

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, ...)

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

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, ...)

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

CentraleSupélec

Model-based Predictive Control

Find optimal control sequence using predictions based on a model.

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

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

Model-based Predictive Control

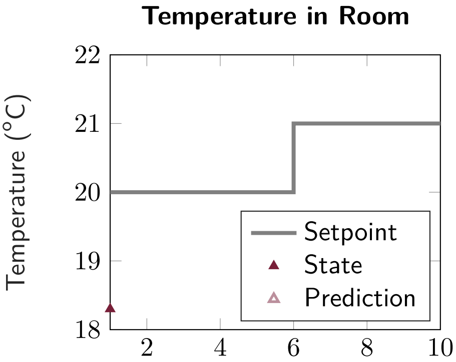
Find optimal control sequence using predictions based on a model.

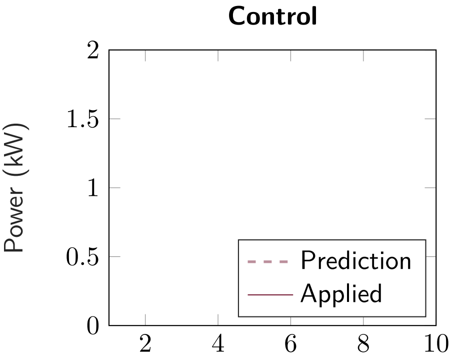
- We need an optimization problem
  - Decision variable is the control sequence (Over horizon N)
  - Objective function to optimize
  - System's Model (states and inputs)
  - Other constraints to respect (QoS, technical restrictions, ...)

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

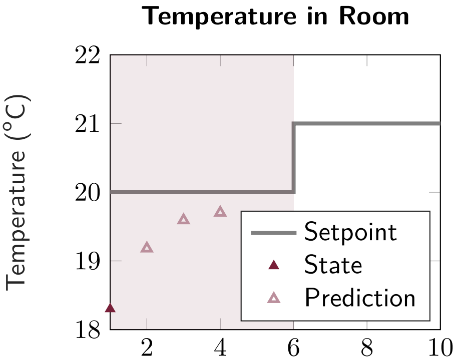
In a nutshell

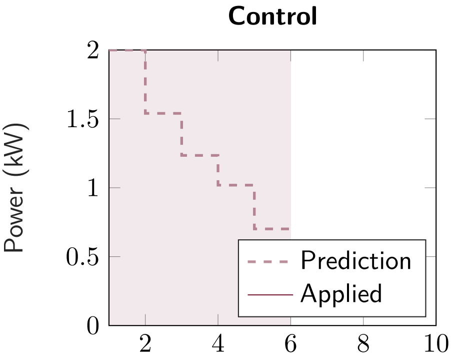
Find optimal control sequence





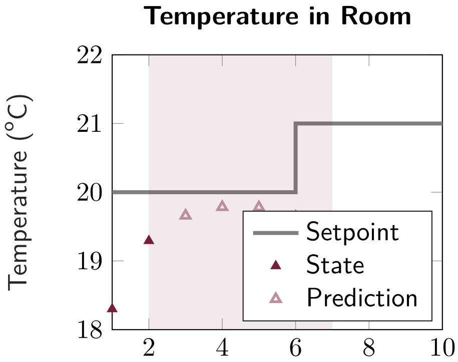
Find optimal control sequence

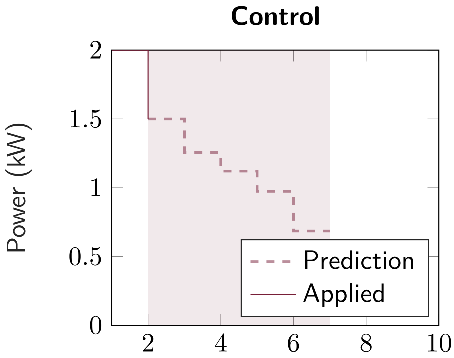




In a nutshell

Find optimal control sequence, apply first element

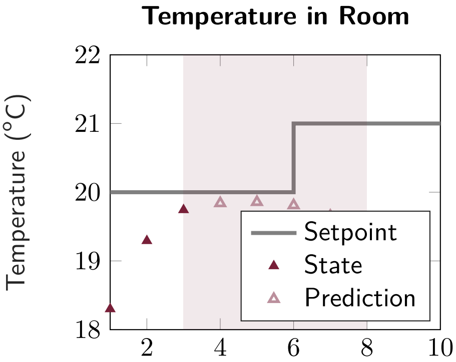


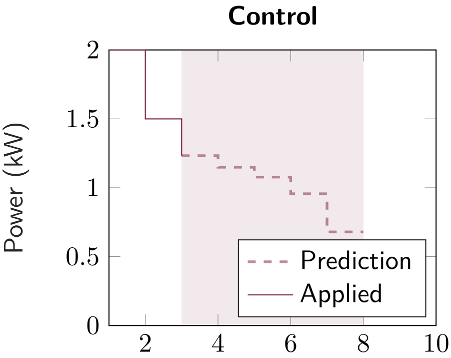


Model Predictive Control

In a nutshell

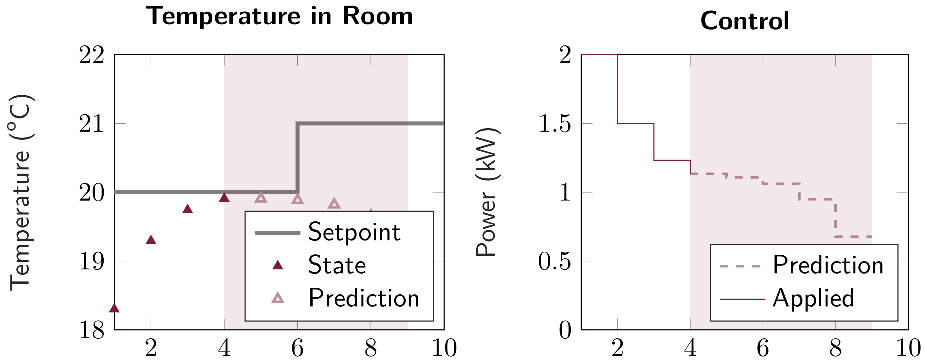
Find optimal control sequence, apply first element, rinse repeat





In a nutshell

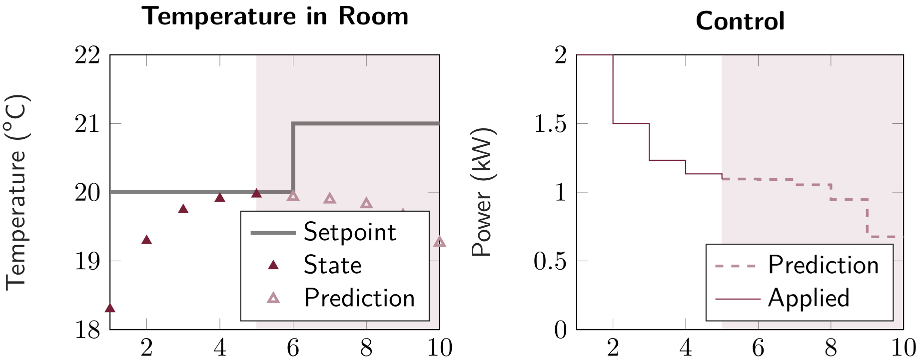
Find optimal control sequence, apply first element, rinse repeat → Receding Horizon



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

CentraleSupélec

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

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

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

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

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
    - 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

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

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

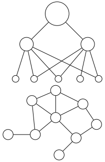
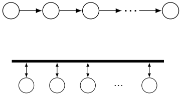
It is about communication

- 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

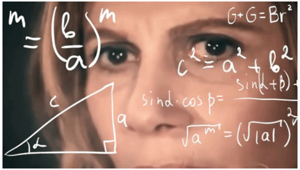
- We break the MPC into multiple
- Make them Communicate , But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...

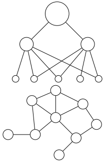



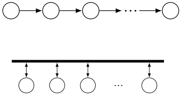
1

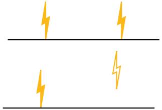
It is about communication

- 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
    - ...







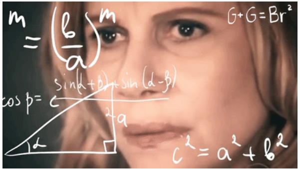


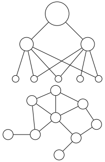
1

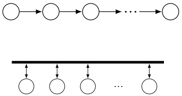
CentraleSupélec

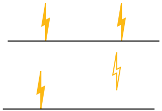
It is about communication

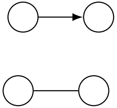
- We break the MPC into multiple
- Make them Communicate , But how?
  - Many flavors to choose from[1]
    - Hierarchical/Anarchical
    - Sequential/Parallel
    - Synchronous/Asynchronous
    - Bidirectional/Unidirectional
    - ...











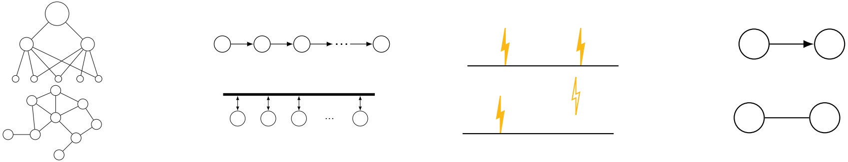
1

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
    - ...

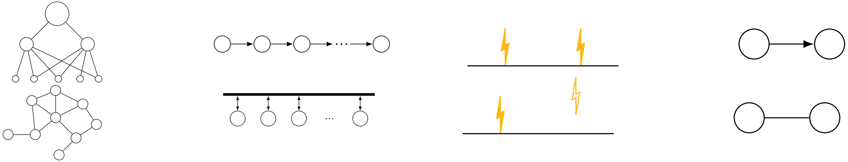


[1] Distributed Model Predictive Control made easy

Distributed Model Predictive Control

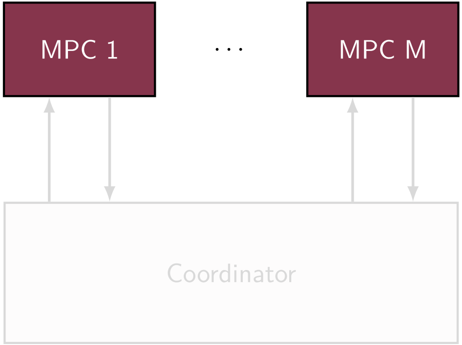
It is about communication

- 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

Optimization Frameworks

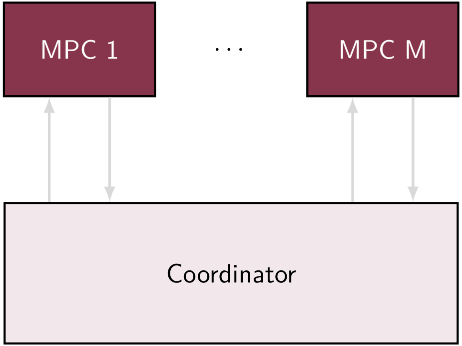


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

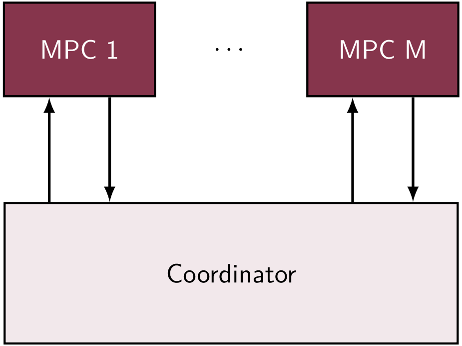


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

} Until Convergence

CentraleSupélec

Optimization Frameworks

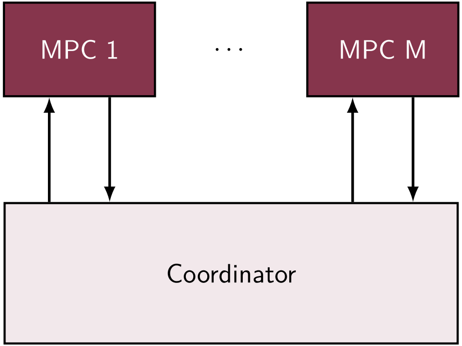


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

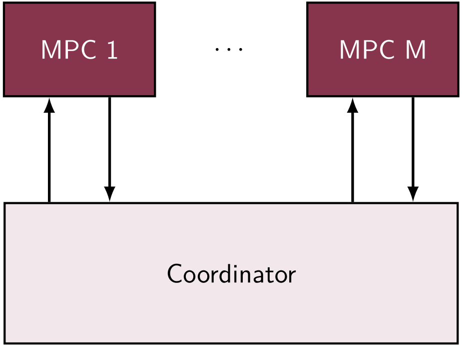


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

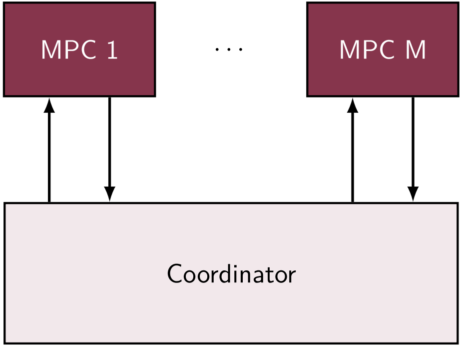


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

Until Convergence

Distributed Model Predictive Control

Optimization Frameworks

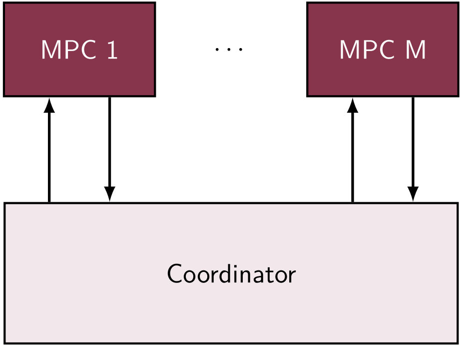


- Coordinator → Hierarchical
- Bidirectional
- No delay → Synchronous
- Agents solve local problems
- Variables are updated

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

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!

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!

CentraleSupélec

Distributed Model Predictive Control

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

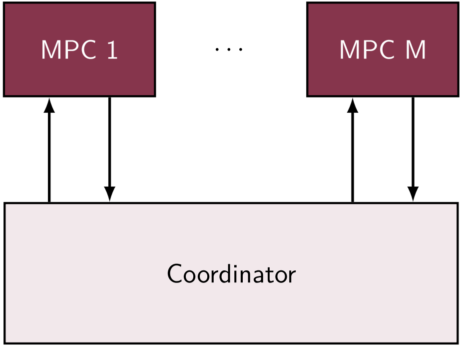
- How can an agent attack?
- What are the consequences of an attack?
- Can we mitigate the effects?

Let's have a preview!

CentraleSupélec

How can a non-cooperative agent attack?

Literature

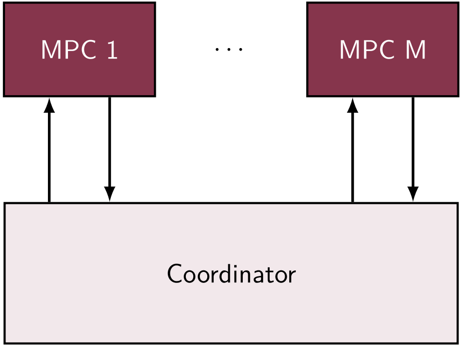


- [Vel+17a; CMI18] present attacks
  - Objective function
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

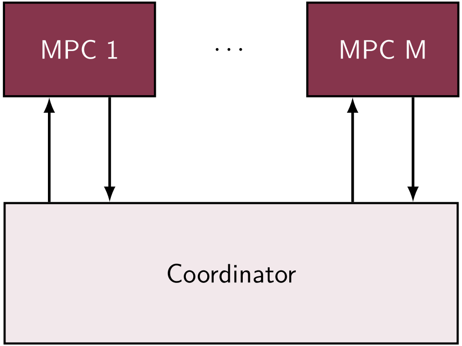


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

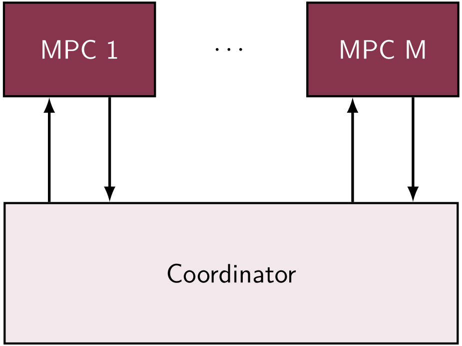


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

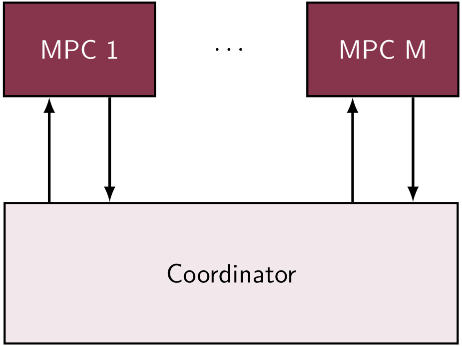


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

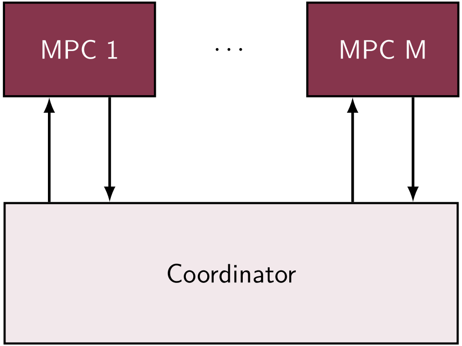


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

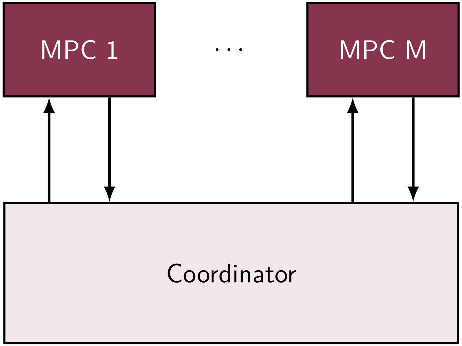


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

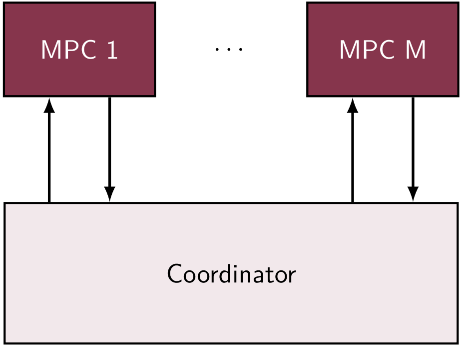
How can a non-cooperative agent attack?

Literature



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

Deception Attacks

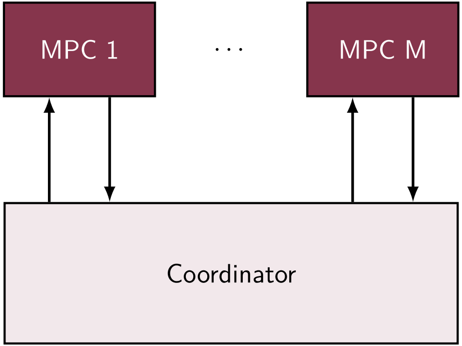


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature

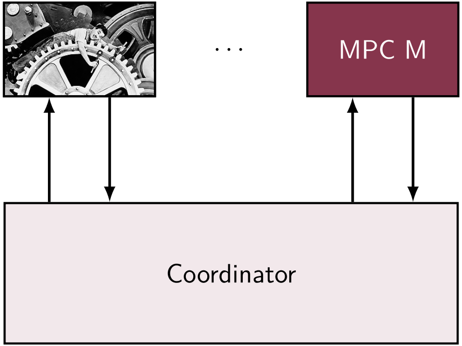


- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks

How can a non-cooperative agent attack?

Literature



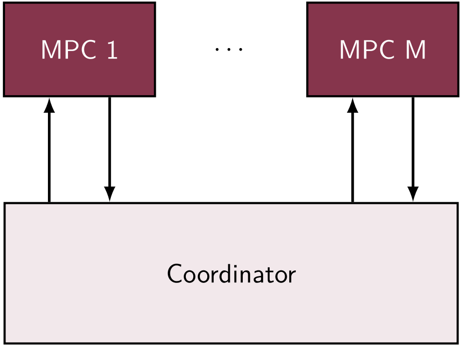
- [Vel+17a; CMI18] present attacks
  - Objective function
    - Selfish Attack
    - Fake weights
    - Fake reference
  - Fake constraints
  - Liar agent

Deception Attacks (Internal change)

CentraleSupélec

How can a non-cooperative agent attack?

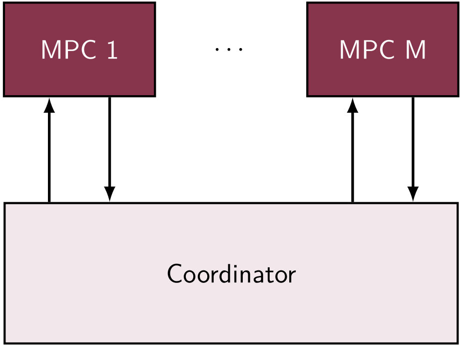
Our approach



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

How can a non-cooperative agent attack?

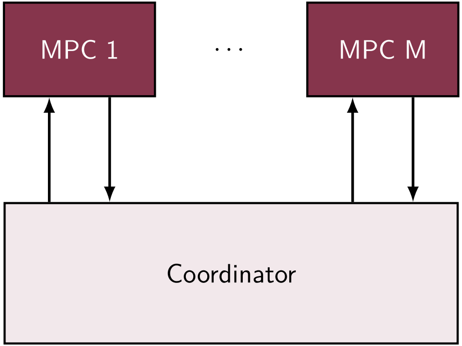
Our approach



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

How can a non-cooperative agent attack?

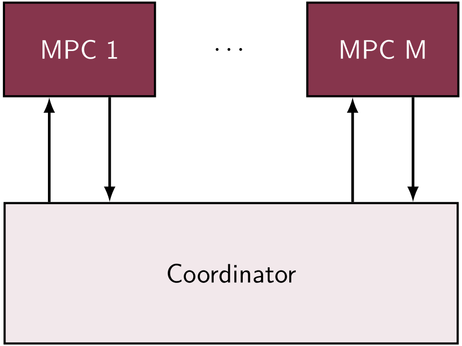
Our approach



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

How can a non-cooperative agent attack?

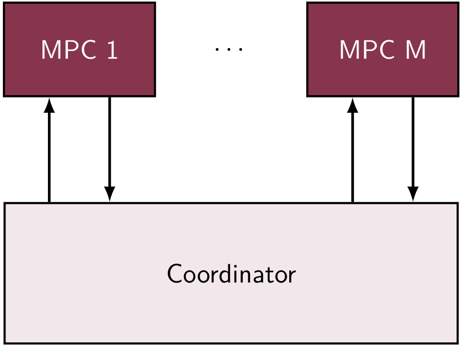
Our approach



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

How can a non-cooperative agent attack?

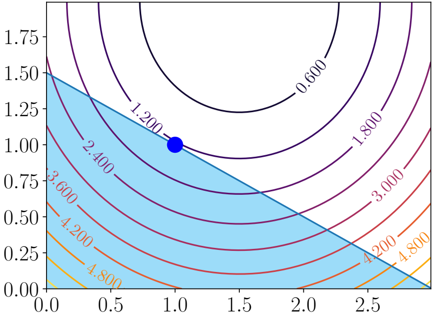
Our approach



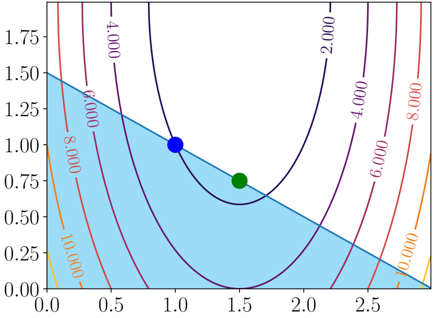
- We are in coordinator's shoes
- What matters is the interface
  - Attacker changes communication
    - False Data Injection

Consequence of an attack

- Attack modifies optimization problem
    - Optimum value is shifted



Original minimum.

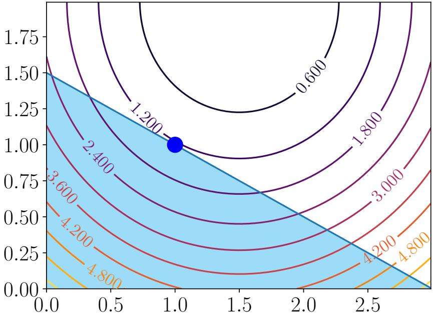


Minimum after attack.

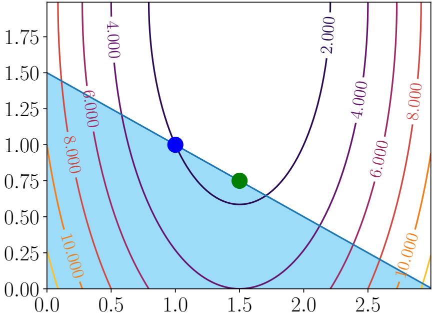
CentraleSupélec

Consequence of an attack

- Attack modifies optimization problem
  - Optimum value is shifted



Original minimum.

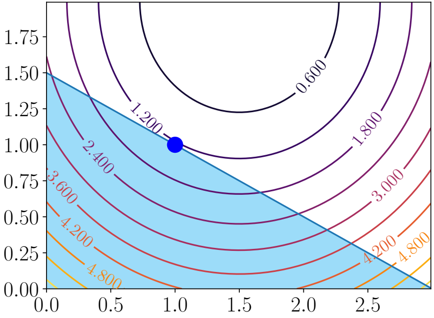


Minimum after attack.

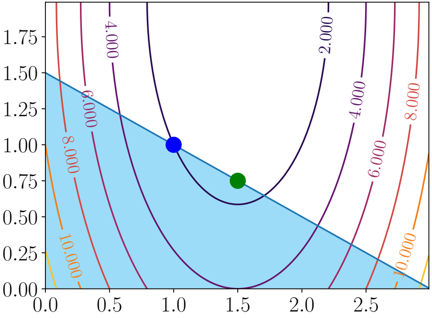
CentraleSupélec

Consequence of an attack

- Attack modifies optimization problem
  - Optimum value is shifted



Original minimum.



Minimum after attack.

CentraleSupélec

Mitigating the effects

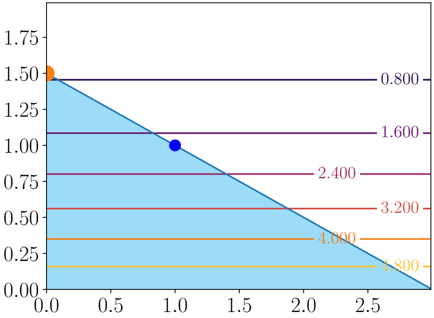
- We can recover by
  - Ignoring attacker
  - Recuperating original behavior (at least trying)

Mitigating the effects

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

Mitigating the effects

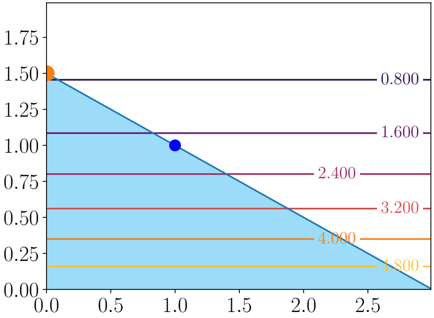
- We can recover by
  - Ignoring attacker
  - Recuperating original behavior (at least trying)



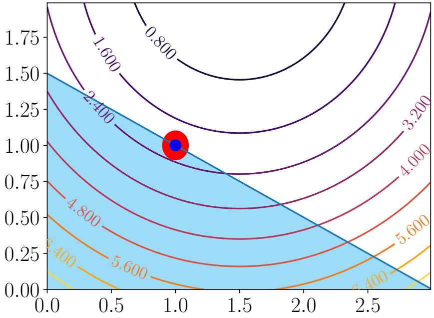
Ignore attacker.

Mitigating the effects

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



Ignore attacker.

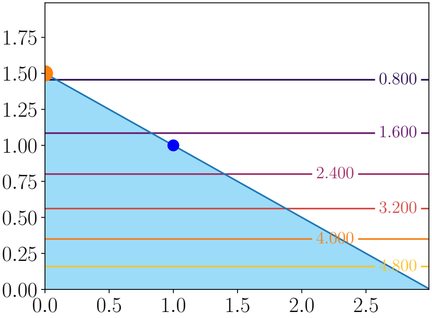


Recover original behavior.

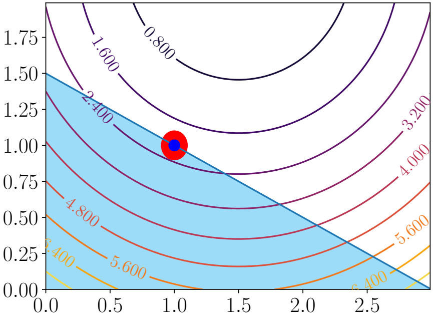
CentraleSupélec

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

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

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

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

$\left\{ \text{Attack free} \right.$

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 $\begin{cases} \text{Attack free} \\ \text{When attack detected} \end{cases}$
  1. Detection/Isolation
  2. Mitigation

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

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

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

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

CentraleSupélec

Outline

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

Outline

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

CentraleSupélec

Outline

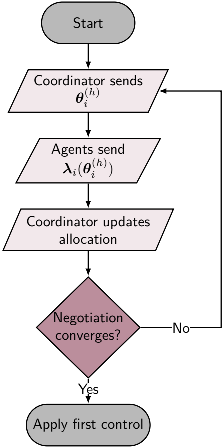
Outline

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

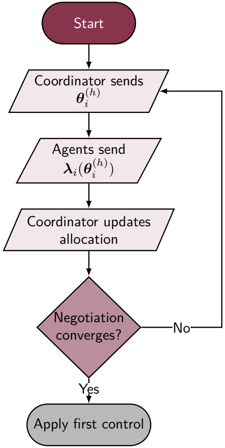
CentraleSupélec

Outline

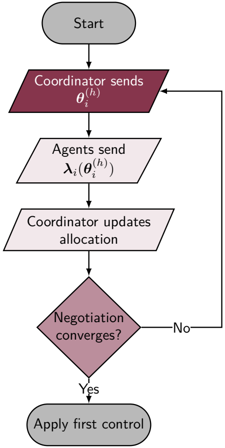
Quantity Decomposition | Resource Allocation



Quantity Decomposition | Resource Allocation

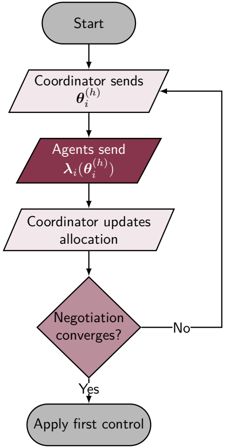


Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

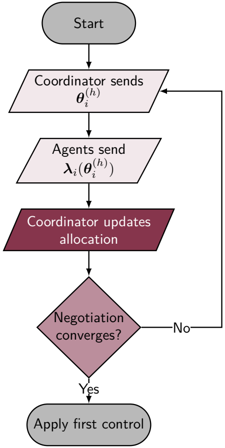
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

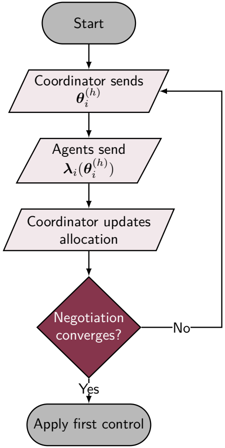
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

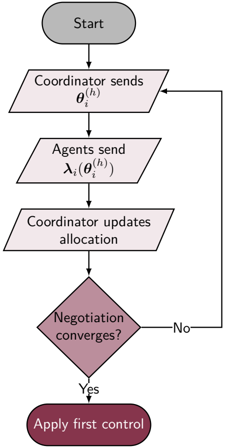
Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

Quantity Decomposition | Resource Allocation



Allocation $\boldsymbol{\theta}_i$

Dissatisfaction $\boldsymbol{\lambda}_i$

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

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

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

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

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

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

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

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

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

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

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

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

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

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

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

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

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

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

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

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

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

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

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

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

CentraleSupélec

Primal Decomposition

or Quantity Decomposition | or Resource Allocation

- Objective is sum of local ones
- Constraints couple variables

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

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

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

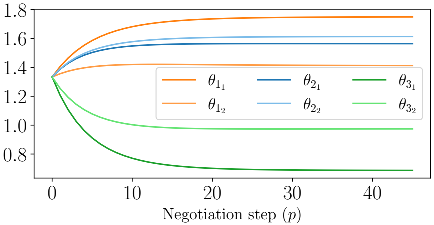
$$\begin{aligned} \underset{\boldsymbol{u}_i}{\text{minimize}} \quad & J_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \\ \text{s. t.} \quad & \boldsymbol{h}_i(\boldsymbol{x}_i, \boldsymbol{u}_i) \preceq \boldsymbol{\theta}_i : \boldsymbol{\lambda}_i \end{aligned}$$

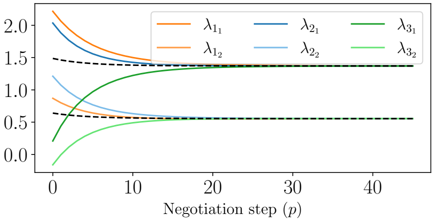
$$\boldsymbol{\theta}[k]^{(p+1)} = \text{Proj}^{\mathcal{S}}(\boldsymbol{\theta}[k]^{(p)} + \rho^{(p)} \boldsymbol{\lambda}[k]^{(p)})$$

CentraleSupélec

Quantity Decomposition | Resource Allocation

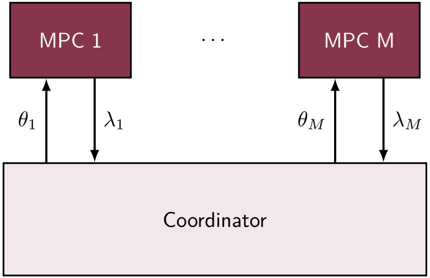
Until everybody is equally dissatisfied





How can a non-cooperative agent attack?

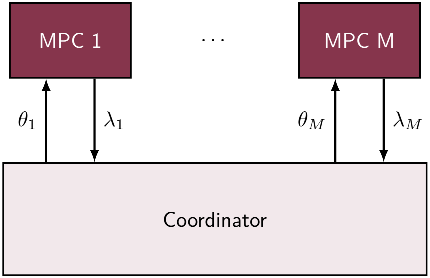
Our approach



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

How can a non-cooperative agent attack?

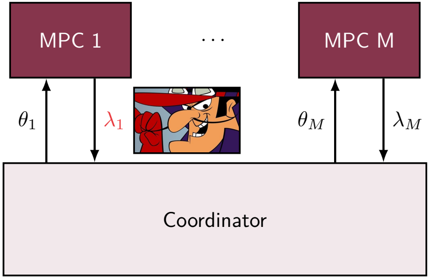
Our approach



- $\boldsymbol{\lambda}_i$ 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

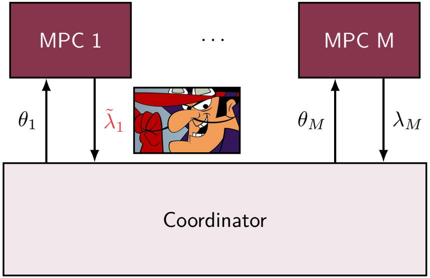


- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters
- Malicious agent modifies $\boldsymbol{\lambda}_i$

CentraleSupélec

How can a non-cooperative agent attack?

Our approach



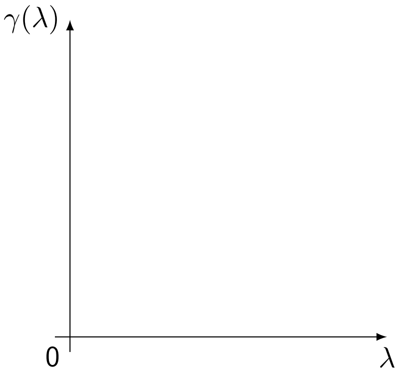
- $\boldsymbol{\lambda}_i$ is the only interface
- $\boldsymbol{\lambda}_i$ depends on local parameters
- Malicious agent modifies $\boldsymbol{\lambda}_i$

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

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

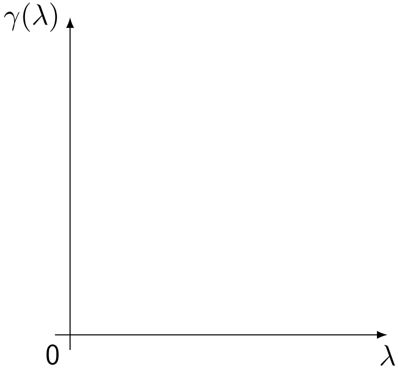
Assumptions

- *Attacker satisfied only if it really is* $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\lambda}_i = T_i[k]\lambda_i \rightarrow \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

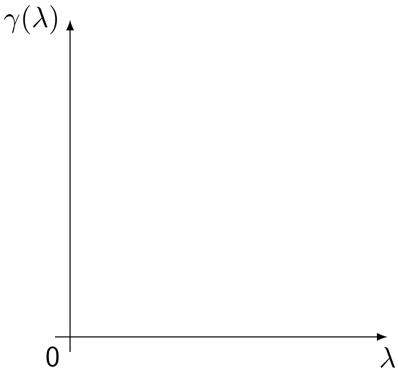
Assumptions

- *Attacker satisfied only if it really is* $\gamma(\lambda) = 0 \to \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \to \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\lambda}_i = T_i[k]\lambda_i \to \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

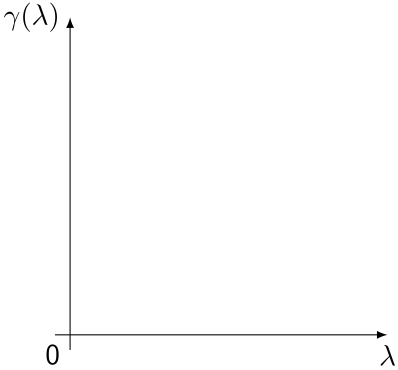
Assumptions

- *Attacker satisfied only if it really is*
  $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\lambda}_i = T_i[k]\lambda_i \rightarrow \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

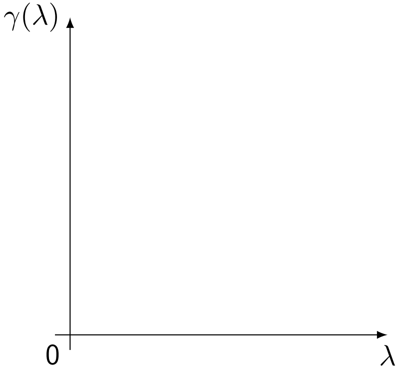
Assumptions

- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

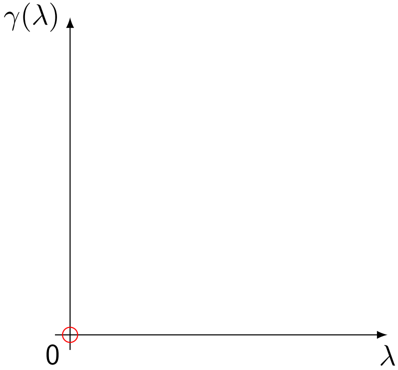
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

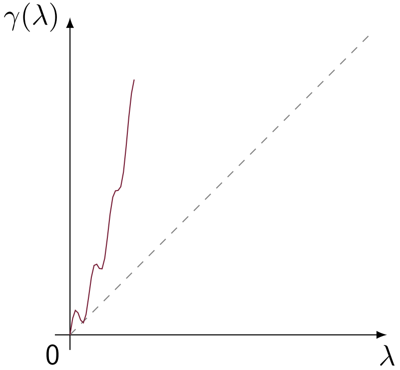
Assumptions

- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy $\gamma(\lambda) > \lambda$*
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

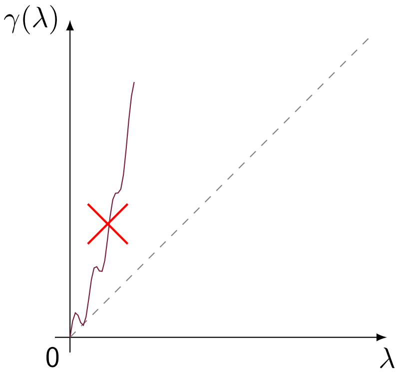
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \to \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing*
  $\lambda_b > \lambda_a \to \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \to \exists T_i[k]^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

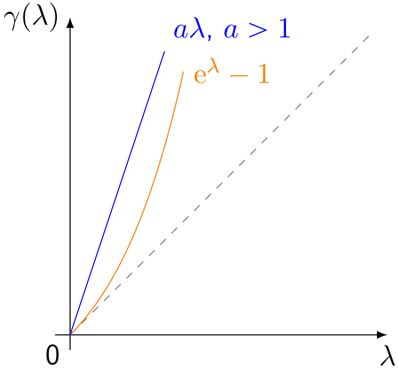
Assumptions

- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing*
  $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

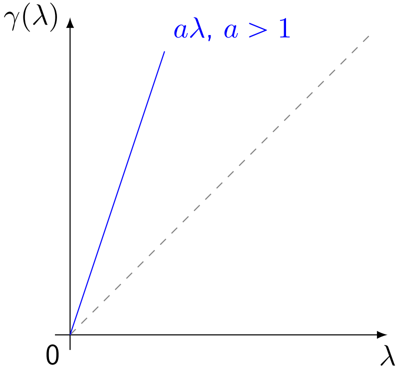
- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

CentraleSupélec

How does an agent lie?

Liar, Liar, Pants of fire



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

Assumptions

- *Attacker satisfied only if it really is*
  - $\gamma(\lambda) = 0 \rightarrow \lambda = 0$
- *Attacker is greedy* $\gamma(\lambda) > \lambda$
- *Attack is monotonically increasing* $\lambda_b > \lambda_a \rightarrow \gamma(\lambda_b) > \gamma(\lambda_a)$

- Invertible
- If $\tilde{\boldsymbol{\lambda}}_i = T_i[k]\boldsymbol{\lambda}_i \rightarrow \exists T_i[k]^{-1}$

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

CentraleSupélec

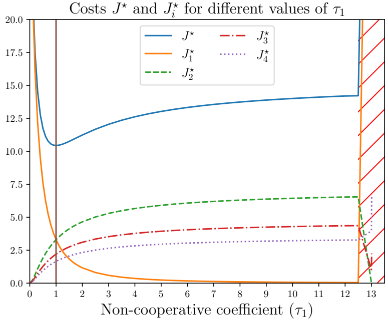
Example

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



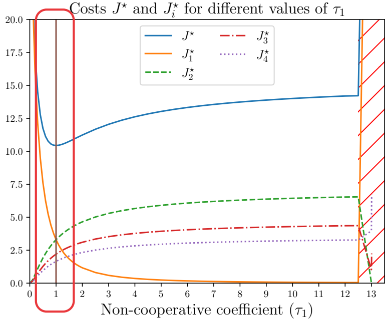
4 distinct agents

- 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

CentraleSupélec

Example



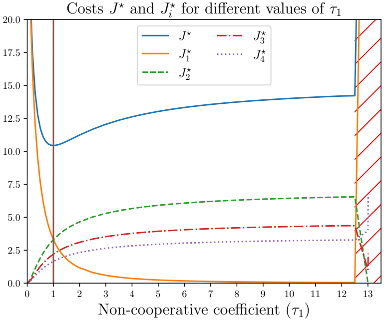
4 distinct agents

- 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

CentraleSupélec

Example

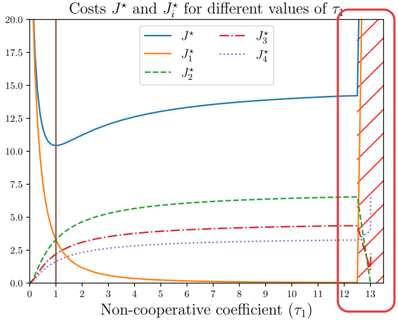


4 distinct agents

- 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

CentraleSupélec

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

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

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

CentraleSupélec

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

Outline

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

CentraleSupélec

What are deprived systems?

Systems whose optimal solution has all constraints active

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

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

[2] Under some conditions

CentraleSupélec

What are deprived systems?

Systems whose optimal solution has all constraints active

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

$$\begin{array}{ll} \underset{U_i[k]}{\text{minimize}} & \frac{1}{2}\|U_i[k]\|_{H_i}^2 + f_i[k]^T U_i[k] \\ \text{subject to} & \bar{\Gamma}_i U_i[k] \leq \theta_i[k] : \lambda_i[k] \end{array}$$

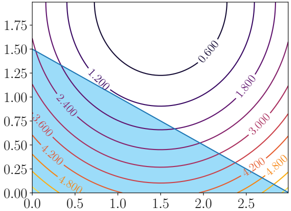
[2] Under some conditions

What are deprived systems?

Systems whose optimal solution has all constraints active

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

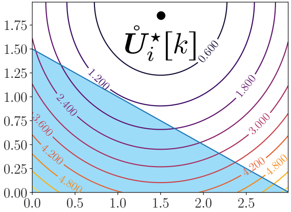
CentraleSupélec

What are deprived systems?

Systems whose optimal solution has all constraints active

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

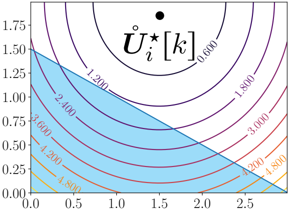
CentraleSupélec

What are deprived systems?

Systems whose optimal solution has all constraints active

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

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



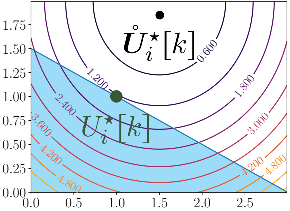
[2] Under some conditions

What are deprived systems?

Systems whose optimal solution has all constraints active

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

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



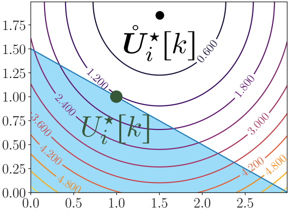
[2]Under some conditions see here

What are deprived systems?

Systems whose optimal solution has all constraints active

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

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



[2] Under some conditions see here

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
  - → 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
  - → 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

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

(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*

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

- Solution is analytical and affine

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

(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*

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

- Solution is analytical and affine

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

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

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

(local parameters unknown by coordinator)

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

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

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

CentraleSupélec

Deprived Systems

Analysis

Assumptions

- *Quadratic local problems*
- *Scarcity*

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

- Solution is analytical and affine

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

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

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

CentraleSupélec

Deprived Systems

Under attack!

- Normal behavior
  - 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

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

- But wait! P_i is not supposed to change!
- Change → 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

- 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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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

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

- 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 $\widehat{\tilde{s}}_i[k]$ 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 $\hat{\tilde{s}}_i[k]$ such as:

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

- 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 $\widehat{\tilde{s}}_i[k]$ 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

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

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

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

- We estimate $\widehat{\widetilde{P}}_i[k]$ and $\widehat{\widetilde{s}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples θ_i^p and λ_i^p $\rightarrow$ low input excitation
- Solution: Send a random[3] sequence to increase excitation until convergence.

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

CentraleSupélec

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

- We estimate $\widehat{\tilde{P}}_i[k]$ and $\widehat{\tilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples θ_i^p and λ_i^p $\rightarrow$ low input excitation
- Solution: Send a random[3] sequence to increase excitation until convergence.

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

CentraleSupélec

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

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

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

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

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

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

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

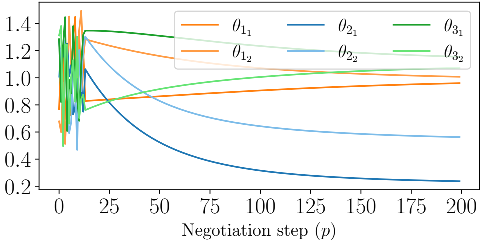
- We estimate $\widehat{\widetilde{P}}_i[k]$ and $\widehat{\widetilde{\boldsymbol{s}}}_i[k]$ simultaneously using RLS
- Challenge: Online estimation during negotiation fails
  - Update function couples $\boldsymbol{\theta}_i^p$ and $\boldsymbol{\lambda}_i^p$ $\rightarrow$ low input excitation
- Solution: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

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

CentraleSupélec

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

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



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

CentraleSupélec

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 $\boldsymbol{\lambda}_i$

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

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

- Reconstruct $\boldsymbol{\lambda}_i$

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

- Choose adequate version for coordination

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

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

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

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

- Reconstruct $\boldsymbol{\lambda}_i$

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

- Choose adequate version for coordination

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

CentraleSupélec

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

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

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

- Reconstruct $\boldsymbol{\lambda}_i$

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

- Choose adequate version for coordination

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

CentraleSupélec

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

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

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

- Reconstruct $\boldsymbol{\lambda}_i$

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

- Choose adequate version for coordination

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

CentraleSupélec

Mitigation mechanism

Reconstructing $\boldsymbol{\lambda}_i$

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

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

- Reconstruct $\boldsymbol{\lambda}_i$

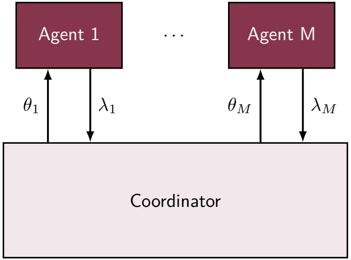
$$\overset{\text{rec}}{\boldsymbol{\lambda}}_i = -\bar{P}_i\boldsymbol{\theta}_i - \widehat{T_i[k]^{-1}}\widehat{\tilde{\boldsymbol{s}}}_i[k]$$

- Choose adequate version for coordination

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

CentraleSupélec

Complete Mechanism

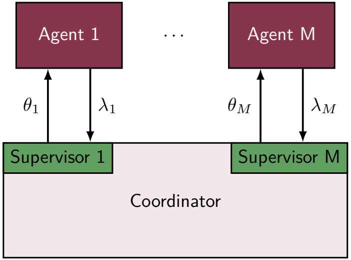


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

Complete Mechanism

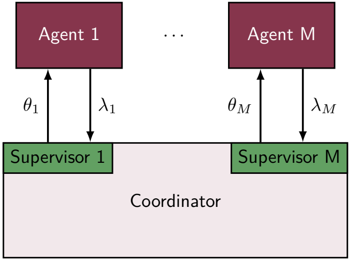


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

Complete Mechanism



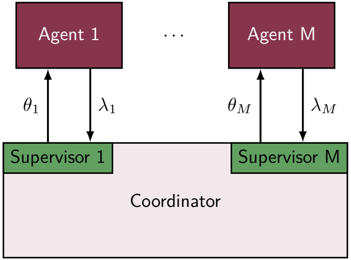
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

CentraleSupélec

Complete Mechanism

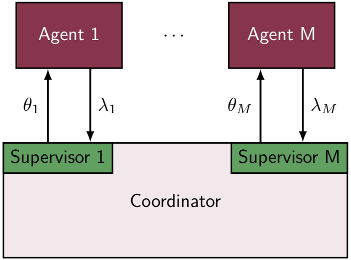


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

Complete Mechanism

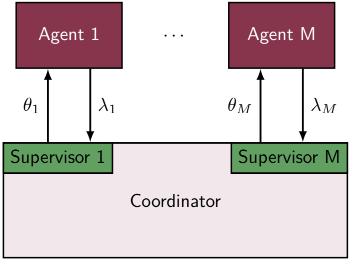


- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

Complete Mechanism



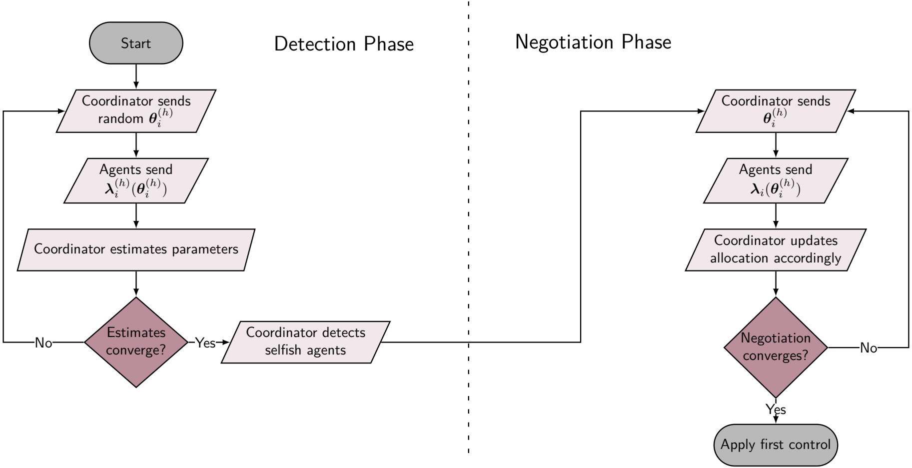
- Supervise exchanges by inquiring the agents
- Estimate how they will behave

Two Phases

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

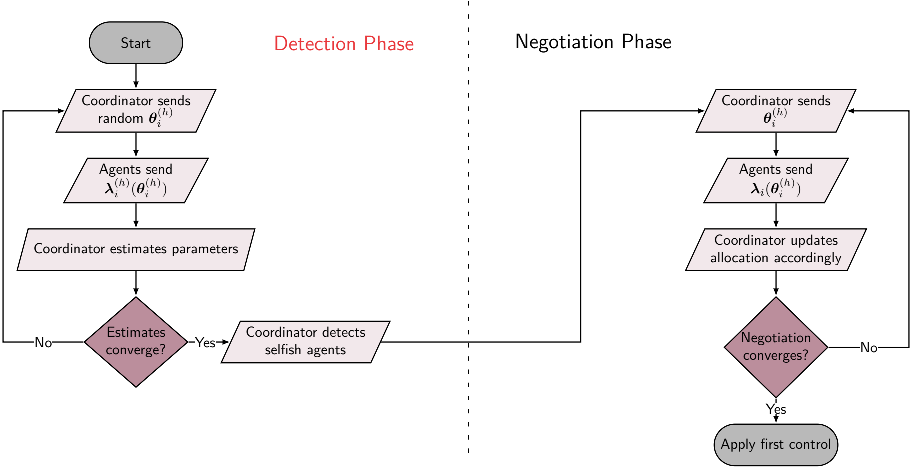
Complete algorithm

RPdMPC-DS



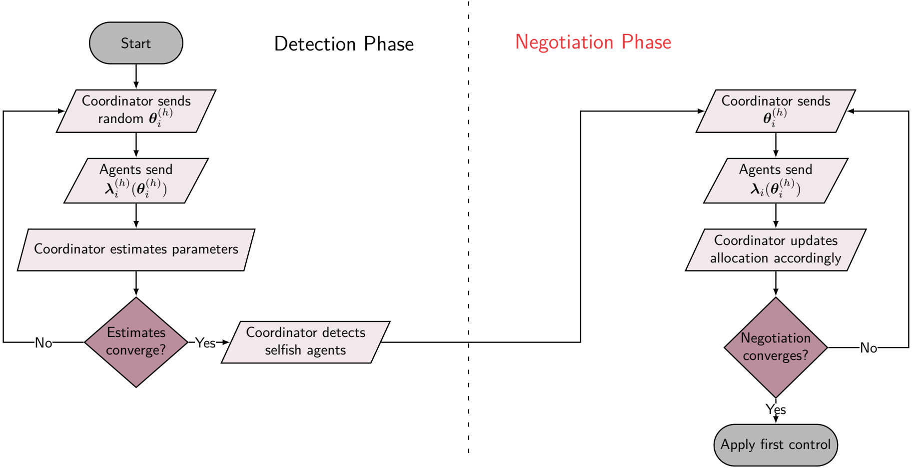
Complete algorithm

RPdMPC-DS



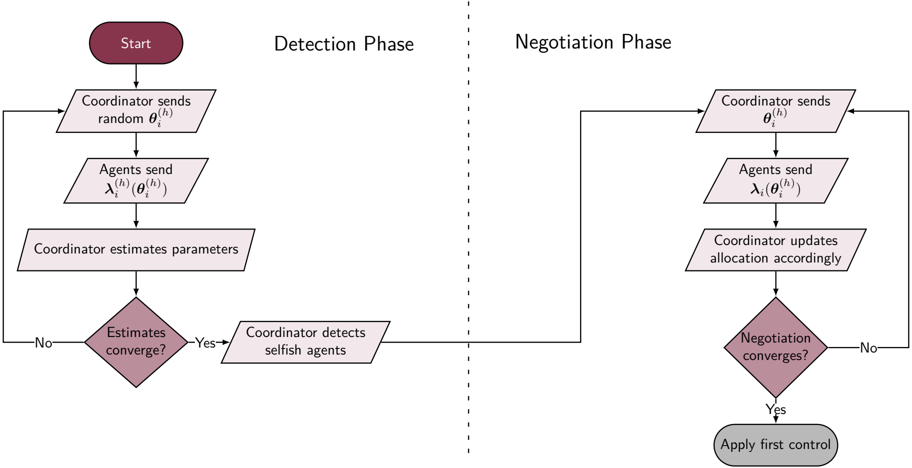
Complete algorithm

RPdMPC-DS



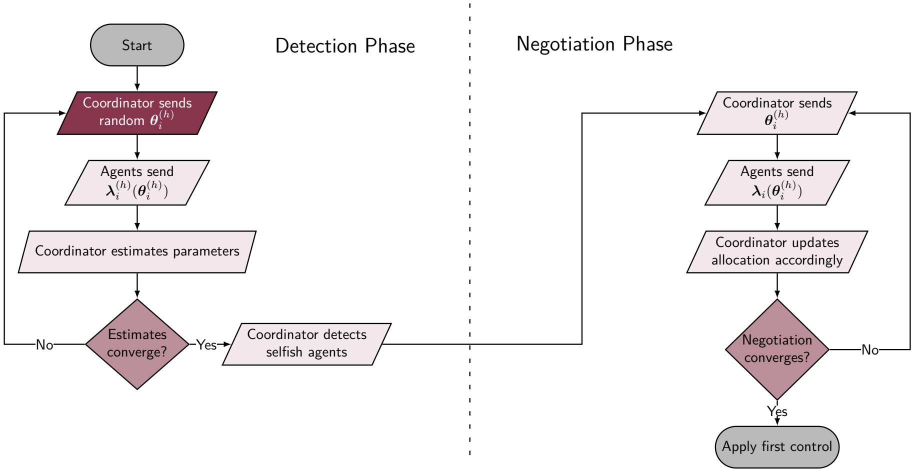
Complete algorithm

RPdMPC-DS



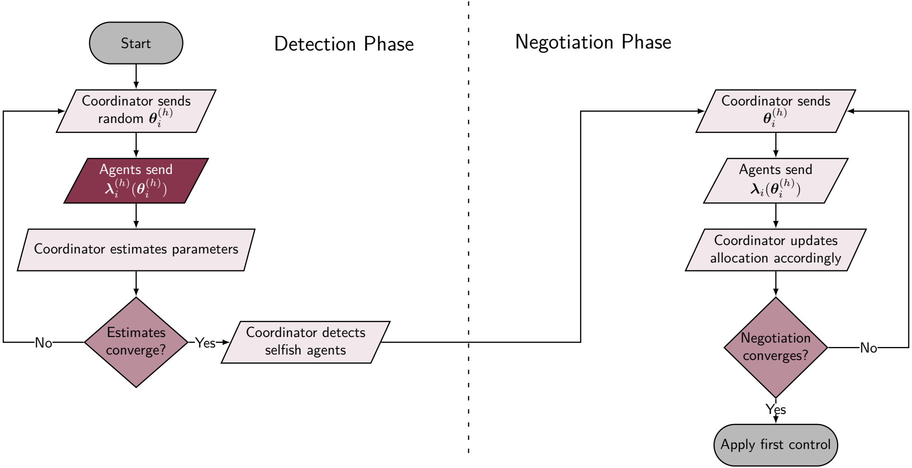
Complete algorithm

RPdMPC-DS



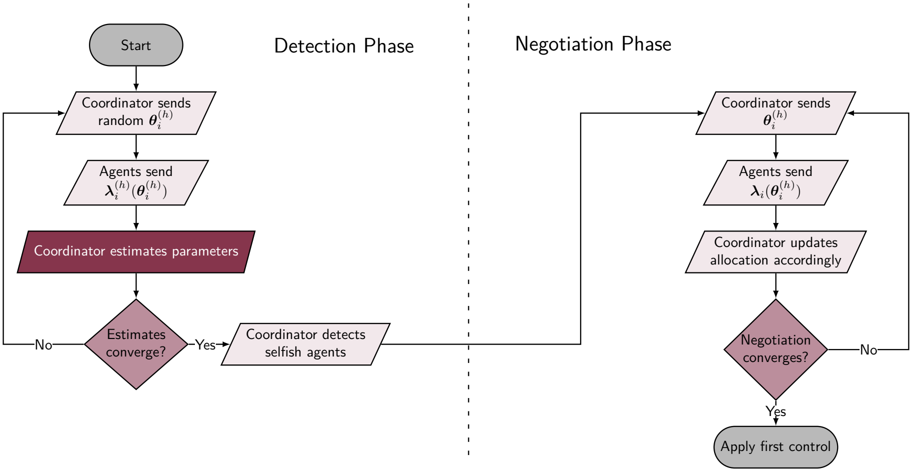
Complete algorithm

RPdMPC-DS



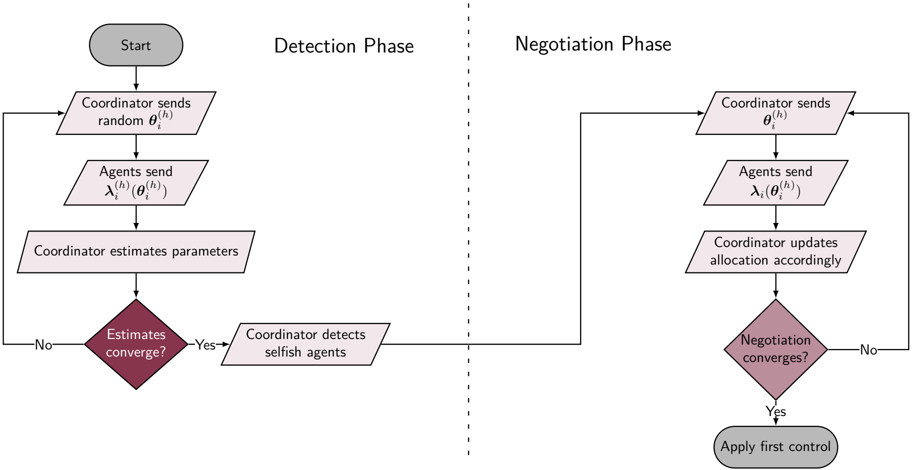
Complete algorithm

RPdMPC-DS



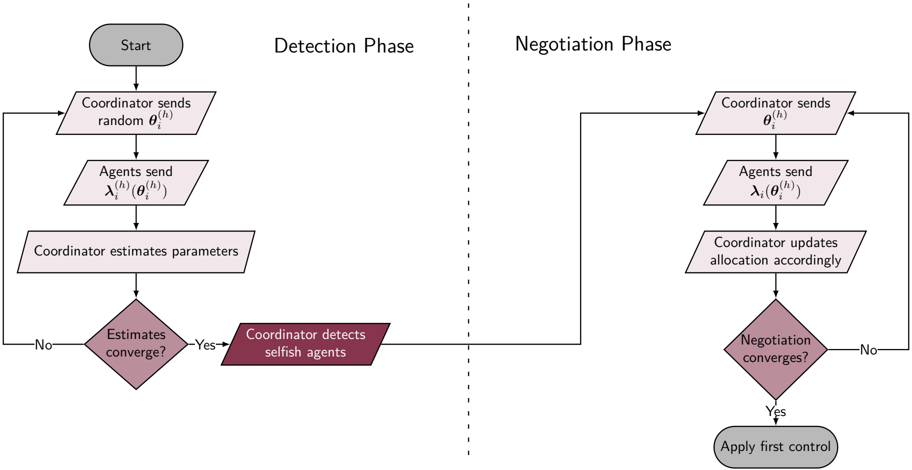
Complete algorithm

RPdMPC-DS



Complete algorithm

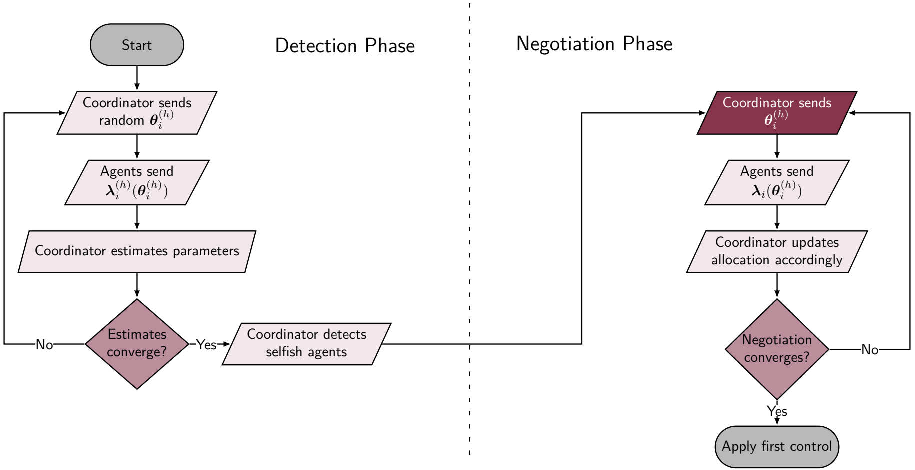
RPdMPC-DS



CentraleSupélec

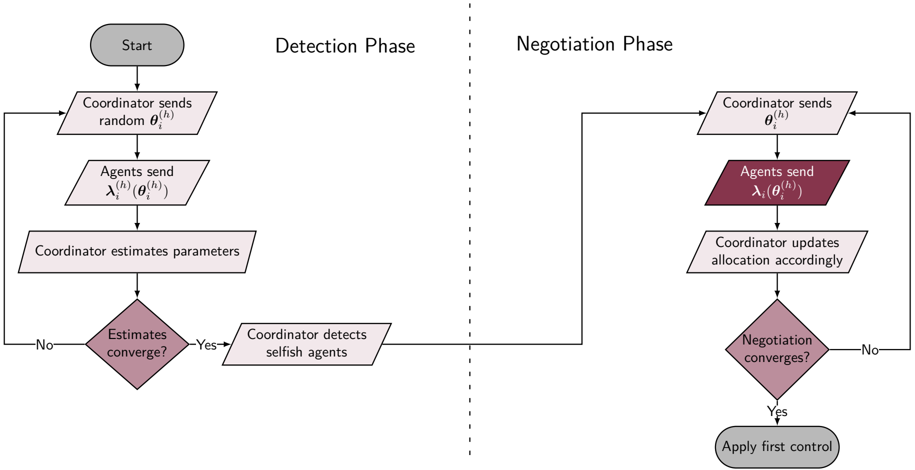
Complete algorithm

RPdMPC-DS



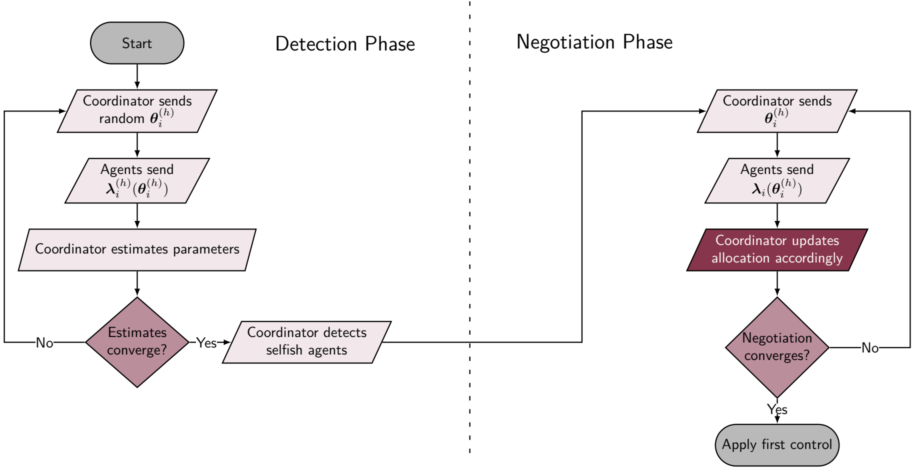
Complete algorithm

RPdMPC-DS



Complete algorithm

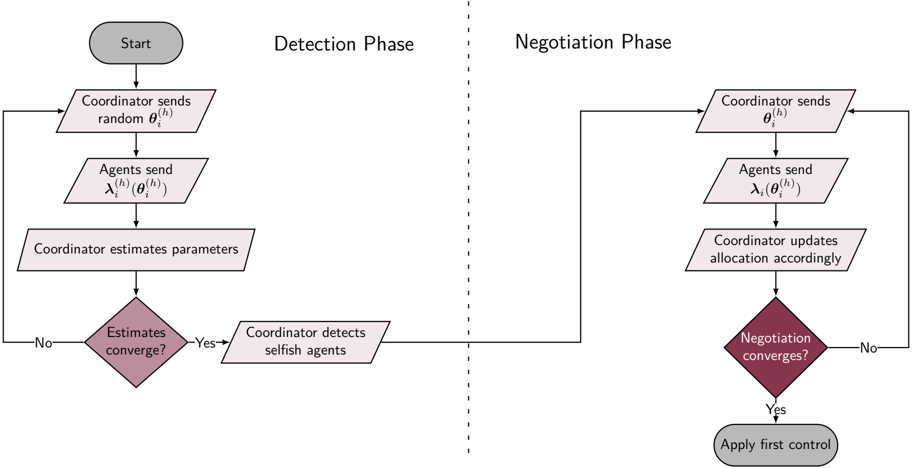
RPdMPC-DS



CentraleSupélec

Complete algorithm

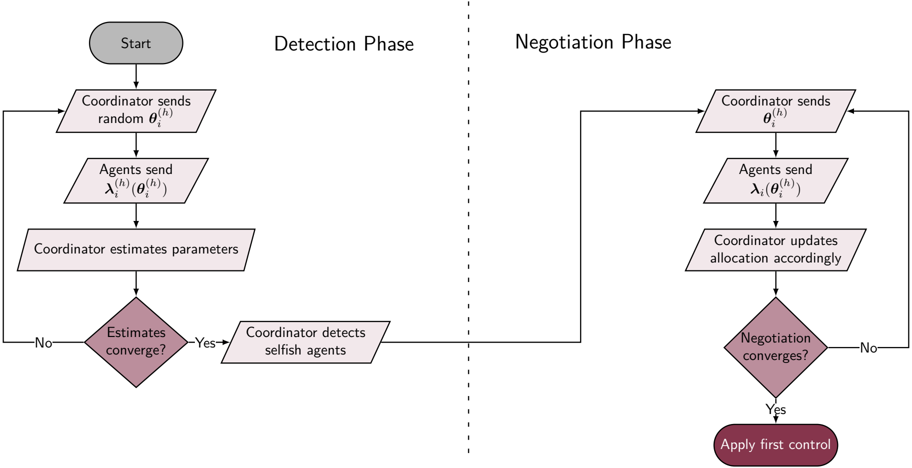
RPdMPC-DS



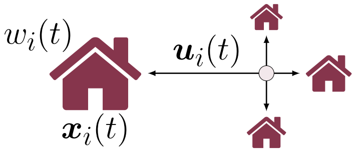
CentraleSupélec

Complete algorithm

RPdMPC-DS



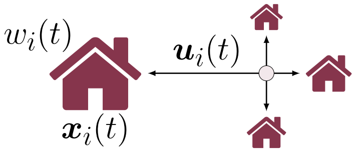
Example



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)

Example

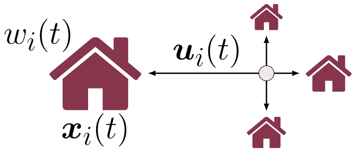


District Heating Network (4 Houses)

- 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

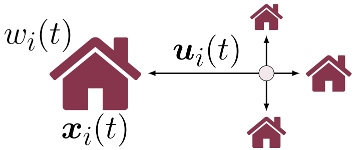
Example



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)

Example

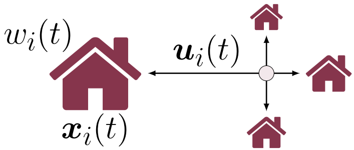


District Heating Network (4 Houses)

- 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

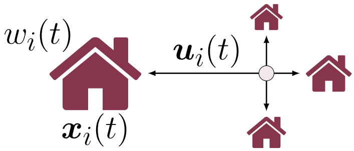
Example



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)

Example

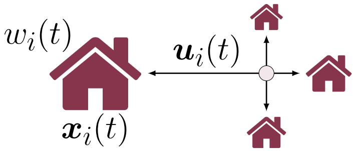


District Heating Network (4 Houses)

- 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

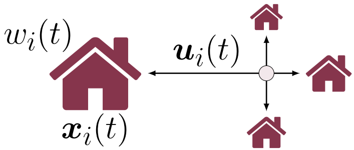


District Heating Network (4 Houses)

- 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



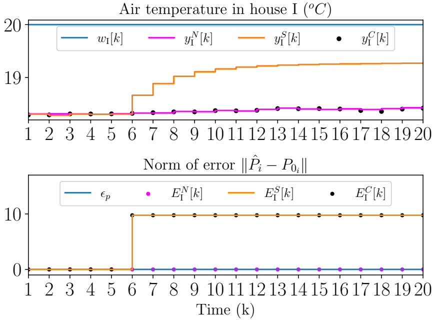
District Heating Network (4 Houses)

- 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

Results

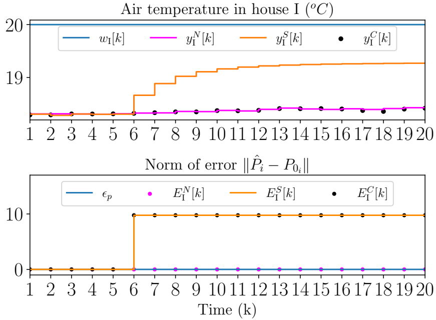
Temporal



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

Results

Temporal

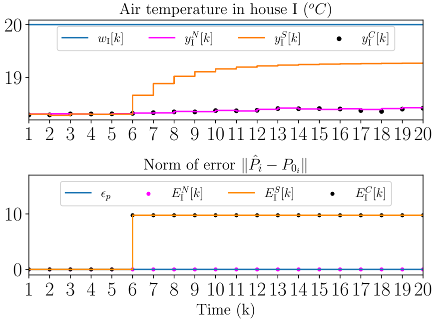


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$
- S Agent increases its comfort
- C Restablish behavior close to N

Results

Temporal



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

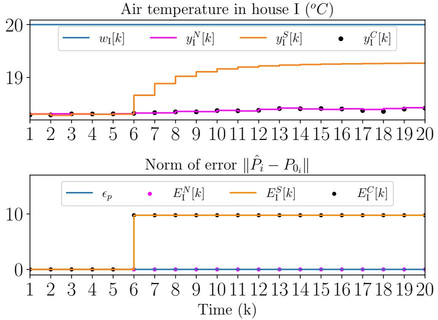
- Agent starts cheating in $k = 6$

S Agent increases its comfort

C Restablish behavior close to N

Results

Temporal

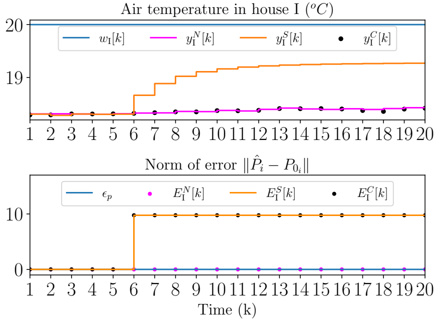


Temperature in house I.
Error $E_I(k)$.
N Nominal, S Selfish, C Corrected

- Agent starts cheating in $k = 6$
- S Agent increases its comfort
- C Restablish behavior close to N

Results

Temporal



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

- Agent starts cheating in $k = 6$
- S Agent increases its comfort
- C Restablish behavior close to N

CentraleSupélec

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.3	0.503
II	21.671	-0.547
III	17.387	-0.004
IV	17.626	-0.09
Global	3.526	0.016

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.3	0.503
II	21.671	-0.547
III	17.387	-0.004
IV	17.626	-0.09
Global	3.526	0.016

Outline

3. Resilient Primal Decomposition-based dMPC using Artificial Scarcity
   - Relaxing some assumptions
   - Adapting the algorithm
   - Applying mechanism

Relaxing scarcity assumption

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

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

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

CentraleSupélec

Relaxing scarcity assumption

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

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

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

CentraleSupélec

Relaxing scarcity assumption

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

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

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

Relaxing scarcity assumption

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

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

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

CentraleSupélec

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

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

Now, we may have multiple (Piecewise affine function)

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

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

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

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

Now, we may have multiple (Piecewise affine function)

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

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

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

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

Now, we may have multiple (Piecewise affine function)

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

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

Analyzing System

Solution for $\lambda_i[k]$

Instead of having one single affine solution

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

Now, we may have multiple (Piecewise affine function)

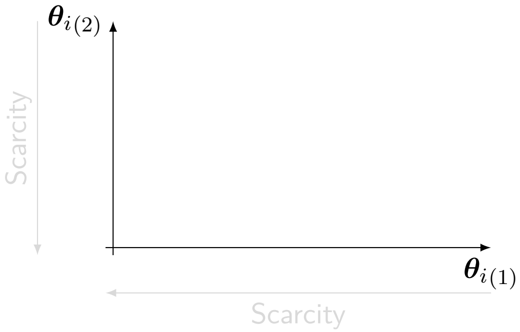
$$\boldsymbol{\lambda}_i[k] = \begin{cases} -P_i^{(0)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(0)}[k], & \text{if } \boldsymbol{\theta}_i[k] \in \mathcal{R}^0_{\boldsymbol{\lambda}_i} \\ \vdots & \vdots \\ -P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k],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

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

CentraleSupélec

Analyzing System

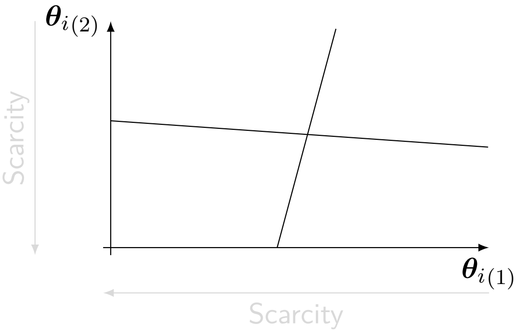
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

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

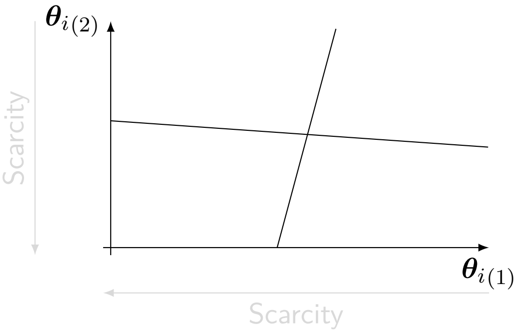


Separation surfaces depend on state and local parameters.

Unknown by the coordinator.

Analyzing System

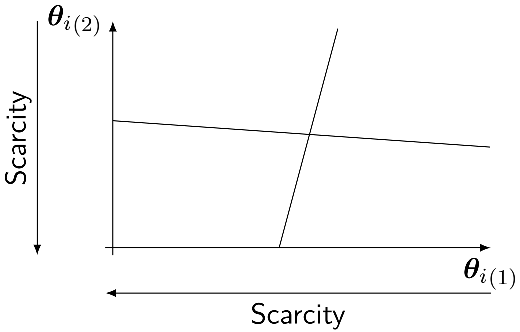
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

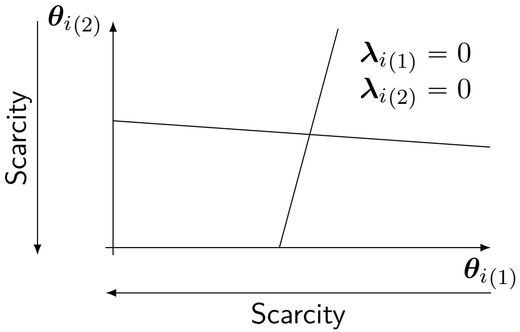
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

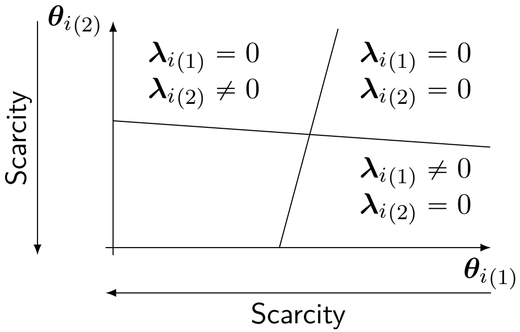
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

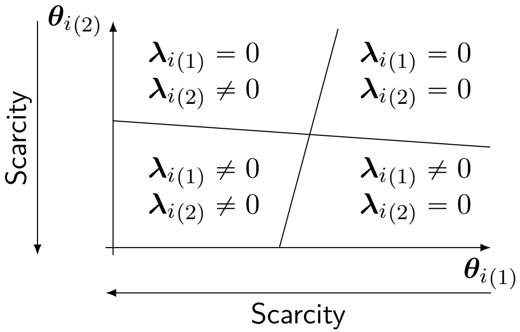
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

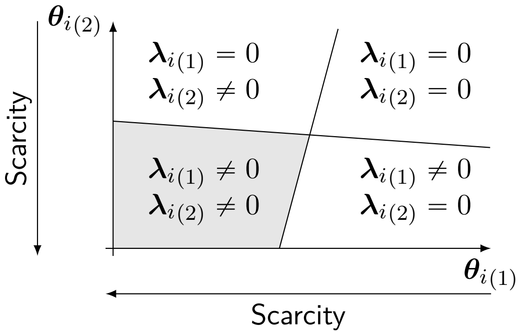
Solution for $\lambda_i[k]$ (Continued)



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

Analyzing System

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



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

Analyzing System

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

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

$\uparrow$ Scarcity $\downarrow$ Sparsity

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

None constraints active $\quad -P_i^{(2^{n_\text{ineq}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_\text{ineq}}-1)}[k] \rightarrow \mathbf{0}$

CentraleSupélec

Analyzing System

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

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

Scarcity ↑ Sparsity ↓

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

None constraints active $-P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k] \rightarrow \mathbf{0}$

CentraleSupélec

Analyzing System

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

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

$\uparrow$ Scarcity $\downarrow$ Sparsity

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

None constraints active $-P_i^{(2^{n_\text{ineq}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_\text{ineq}}-1)}[k] \rightarrow \mathbf{0}$

Analyzing System

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

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

$\uparrow$ Scarcity $\downarrow$ Sparsity

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

None constraints active $\quad -P_i^{(2^{n_\text{ineq}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_\text{ineq}}-1)}[k] \rightarrow \mathbf{0}$

CentraleSupélec

Analyzing System

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

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

Scarcity ↑ Sparsity ↓

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

None constraints active $\quad -P_i^{(2^{n_{\text{ineq}}}-1)}\boldsymbol{\theta}_i[k] - \boldsymbol{s}_i^{(2^{n_{\text{ineq}}}-1)}[k] \rightarrow \mathbf{0}$

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

CentraleSupélec

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

CentraleSupélec

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

CentraleSupélec

Analyzing System

Under attack!

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

Parameters are modified. But not the regions' limits

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

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

CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\theta}_i$ which activates all constraints[4]

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

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ *which activates all constraints*[4]

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

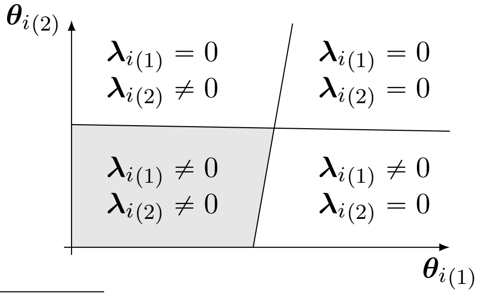
CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ *which activates all constraints*[4]



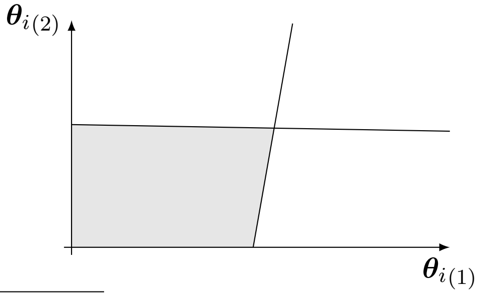
[4] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



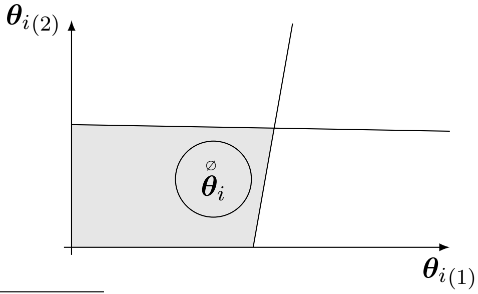
[4] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



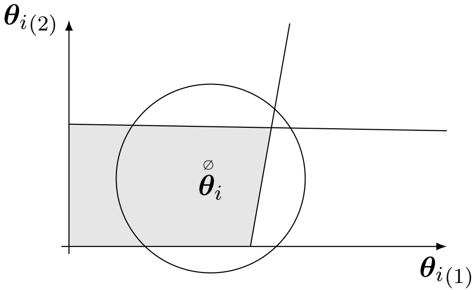
[4] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



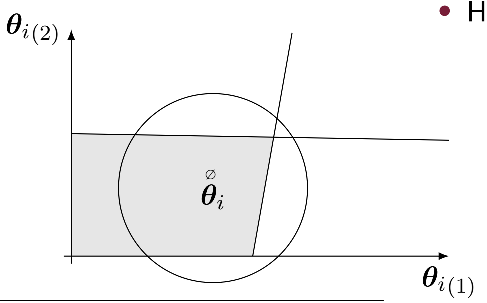
[4] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



- How to known the radius?
  - We don't.
  - Let's estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$ nonetheless

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

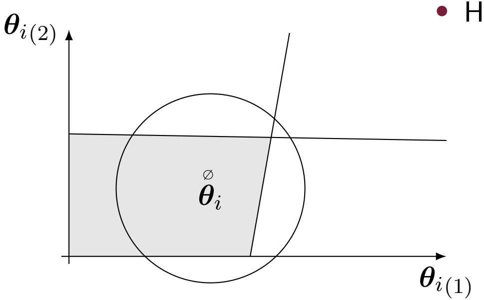
CentraleSupélec

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



- How to known the radius?
  - We don't.
  - Let's estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$ nonetheless

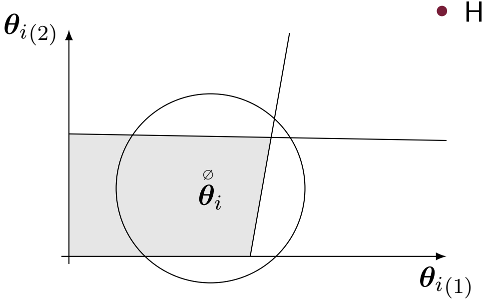
[4] If we have local constraints, we suppose this point respects them.

Artificial Scarcity

Who is it? Who is it?

Assumption

We known a point $\overset{\varnothing}{\boldsymbol{\theta}}_i$ which activates all constraints[4]



- How to known the radius?
  - We don't.
  - Let's estimate $\widehat{\widetilde{P}}_i^{(0)}[k]$ nonetheless

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

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\tilde{P}}_i^{(n)}[k], \widehat{\tilde{s}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\tilde{P}}_i^{(n)}[k], \widehat{\tilde{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\tilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{P}_i^{(n)}[k], \widehat{s}_i^{(n)}[k])$ having generated each $\tilde{\lambda}_i^o[k]$
  - M new estimates $(\widehat{\widehat{P}}_i^{(n)}[k], \widehat{\widehat{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widehat{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{s}}_i^{(n)}[k])$ having generated each $\tilde{\lambda}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{P}_i^{(n)}[k], \widehat{s}_i^{(n)}[k])$ having generated each $\tilde{\lambda}_i^o[k]$
  - M new estimates $(\widehat{\widehat{P}}_i^{(n)}[k], \widehat{\widehat{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widehat{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widehat{P}}_i^{(n)}[k], \widehat{\boldsymbol{s}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widehat{P}}_i^{(n)}[k], \widehat{\boldsymbol{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widehat{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\tilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

CentraleSupélec

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - E the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - M new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\widetilde{\boldsymbol{s}}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Enter Expectation Maximization

- Iterative method to estimate parameters of multimodal models[5]
- We give multiple observations $\boldsymbol{\theta}_i^o[k]$ and $\tilde{\boldsymbol{\lambda}}_i^o[k]$
- At each step we calculate
  - (E) the probability of each $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\boldsymbol{s}}_i^{(n)}[k])$ having generated each $\tilde{\boldsymbol{\lambda}}_i^o[k]$
  - (M) new estimates $(\widehat{\widetilde{P}}_i^{(n)}[k], \widehat{\boldsymbol{s}}_i^{(n)}[k])$ based on the probabilities
- At the end we have
  1. Parameters with associated region index
  2. Observations with associated region index
- We consult the index associated to $\overset{\varnothing}{\boldsymbol{\theta}}_i$
- We recover the associated parameter, i.e., $\widehat{\widetilde{P}}_i^{(0)}[k]$

[5] Such as our PWA function using some tricks

Detection and Mitigation

Same same, but different

Assumption

We know nominal $\bar{P}_i^{(0)}$

- Detection

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

- Mitigation

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

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

CentraleSupélec

Detection and Mitigation

Same same, but different

Assumption

We know nominal $\bar{P}_i^{(0)}$

- Detection

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

- Mitigation

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

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

CentraleSupélec

Detection and Mitigation

Same same, but different

Assumption

We know nominal $\bar{P}_i^{(0)}$

- Detection

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

- Mitigation

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

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

CentraleSupélec

Detection and Mitigation

Same same, but different

Assumption

We know nominal $\bar{P}_i^{(0)}$

- Detection

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

- Mitigation

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

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

CentraleSupélec

Detection and Mitigation

Same same, but different

Assumption

We know nominal $\bar{P}_i^{(0)}$

- Detection

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

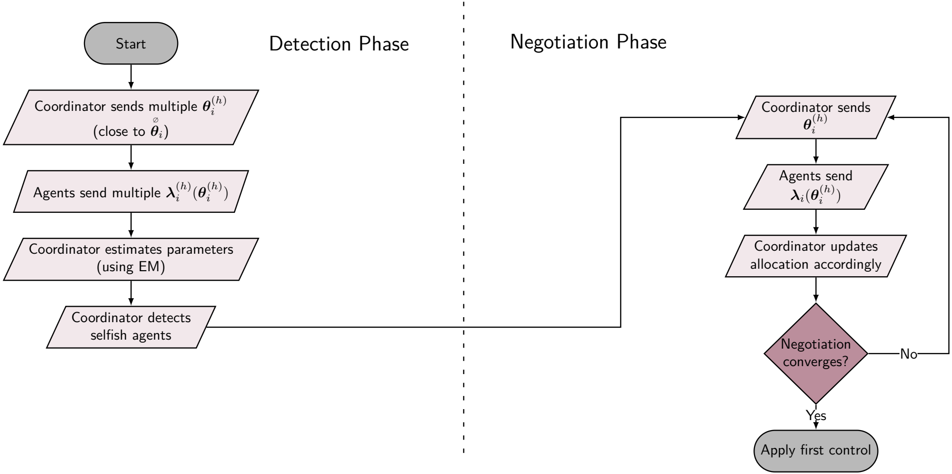
- Mitigation

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

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

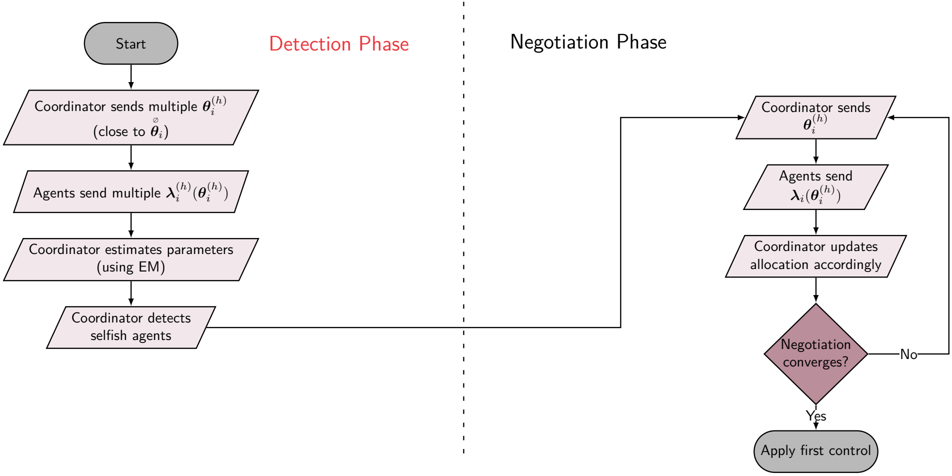
Complete algorithm

RPdMPC-AS



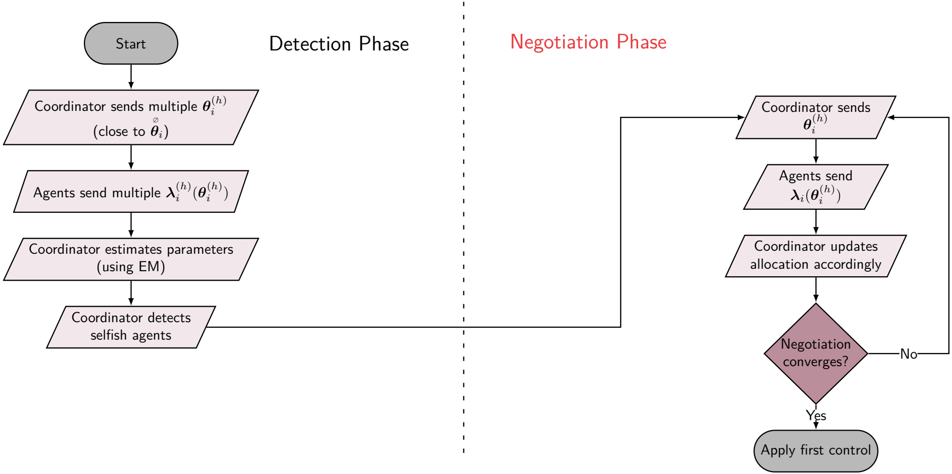
Complete algorithm

RPdMPC-AS



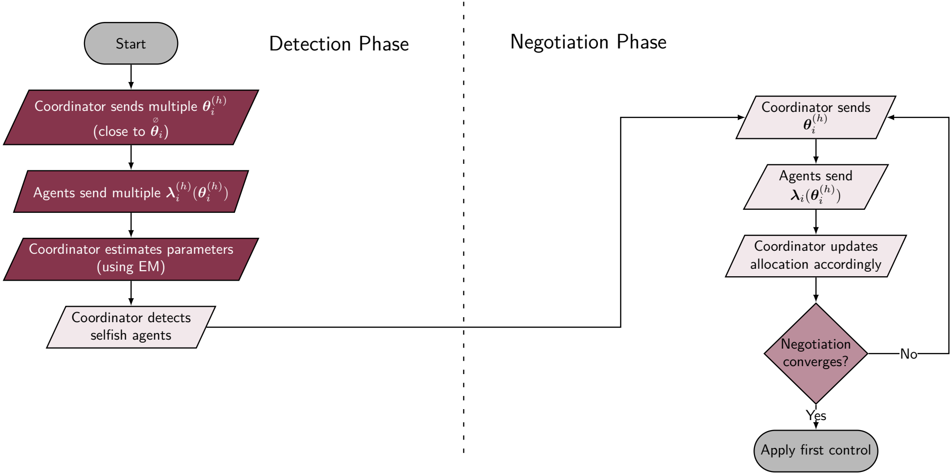
Complete algorithm

RPdMPC-AS



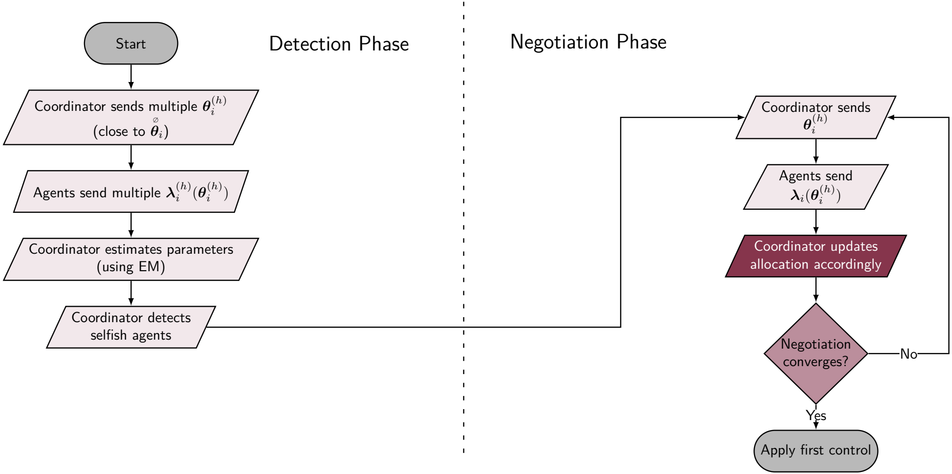
Complete algorithm

RPdMPC-AS

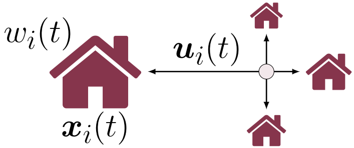


Complete algorithm

RPdMPC-AS



Example

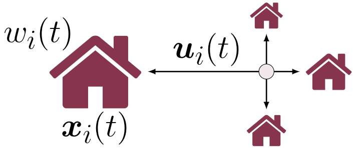


District Heating Network (4 Houses)

- Houses modeled using 3R-2C
- Not enough power
- Period of $5\mathrm{h}$ ($T_s = 0.25\mathrm{h}$)
- 3 scenarios
  - N Nominal
  - C Agent I cheats (dMPC)
  - S Agent I cheats (RPdMPC-AS)

CentraleSupélec

Example

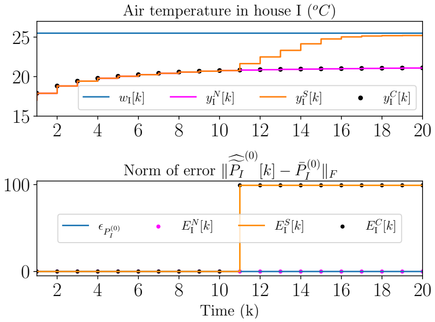


District Heating Network (4 Houses)

- 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)

Results

Temporal



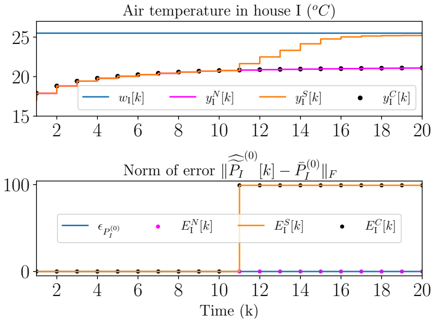
Temperature in house I.

Error $E_I(k)$.

N Nominal, S Selflish C Corrected

Results

Temporal



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

Results

Costs

Objective functions J_i (Normalized error %)

Agent	Selfish	Corrected
I	-36.489	-0.0
II	35.813	0.0
III	29.225	0.0
IV	37.541	0.0
Global	10.689	-0.0

Too good to be true!

It's a kind of magic!

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

Too good to be true!

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

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

CentraleSupélec

Too good to be true!

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

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

CentraleSupélec

Too good to be true!

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

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

Too good to be true!

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

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

Too good to be true!

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

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

Too good to be true!

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

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

CentraleSupélec

Too good to be true!

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

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

CentraleSupélec

Too good to be true!

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

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

CentraleSupélec

Outline

4 Conclusion

Conclusion

Main takeaways

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

Conclusion

Main takeaways

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

Conclusion

Main takeaways

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

Conclusion

Main takeaways

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

CentraleSupélec

Conclusion

Main takeaways

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

CentraleSupélec

Conclusion

Main takeaways

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

Conclusion

Main takeaways

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

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Conclusion

- Insights from the analysis of the solutions of the optimization problems
  - We found some parameters that are constant when there is no cheating
  - The same parameters change when system is attacked
- Exploiting the solution, we find how to invert the cheating function
  - Straightforward if system is scarce
  - If not scarce we try to force it artifically
- But what if scarcity information is not available even artificially?

CentraleSupélec

Open question Future Directions

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

CentraleSupélec

Open question Future Directions

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

CentraleSupélec

Open question Future Directions

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

CentraleSupélec

Open question Future Directions

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

CentraleSupélec

Open question Future Directions

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

Open question Future Directions

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

CentraleSupélec

Thank you!

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

Contact
rafael.accacio.nogueira@gmail.com