NTNU – Trondheim
Norwegian University of Science and Technology

SUBPRO
SUBSEA PRODUCTION AND PROCESSING

A Distributed Algorithm for Scenario-based Model Predictive Control using Primal Decomposition

Dinesh Krishnamoorthy*, Bjarne Foss, Sigurd Skogestad

Department of Chemical Engineering
Norwegian University of Science and Technology (NTNU)
**dinesh.krishnamoorthy@ntnu.no*

26 July 2018

IFAC ADCHEM 2018, Shenyang, China

Outline

Model Predictive Control under Uncertainty

$$\min_{\mathbf{x}_k,\mathbf{u}_k} \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_k, \mathbf{u}_k)$$

s.t

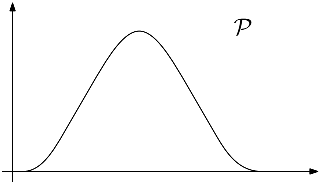
$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{p})$$

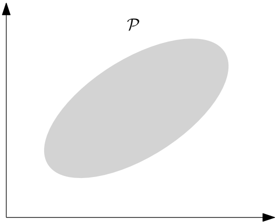
$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{p}) \leq 0$$

$$\mathbf{x}_0 = \hat{\boldsymbol{x}}$$

$$\color{red}{\boldsymbol{p} \in \mathcal{P}}$$

$$\forall k \in \{1, \cdots, N\}$$





Model Predictive Control under Uncertainty

$$\min_{\mathbf{x}_k, \mathbf{u}_k} \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_k, \mathbf{u}_k)$$

s.t

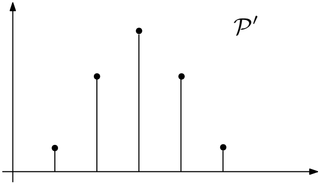
$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{p})$$

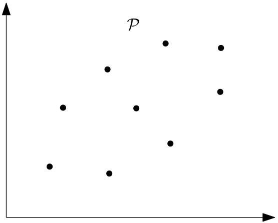
$$\mathbf{g}(\mathbf{x}_k, \mathbf{u}_k, \boldsymbol{p}) \leq 0$$

$$\mathbf{x}_0 = \hat{\boldsymbol{x}}$$

$$\boldsymbol{p} \in \mathcal{P}$$

$$\forall k \in \{1, \cdots, N\}$$





Multistage Scenario-based MPC

- Propagates future evolution of uncertainty using a discrete scenario tree.

[1] P.O.M. Scokaert and D.Q. Mayne, Min-max feedback model predictive control, IEEE Transactions on Automatic Control 43:8 (1998) 11361142.

Multistage Scenario-based MPC

- Propagates future evolution of uncertainty using a discrete scenario tree.
- Optimize over different control trajectories[1]

[1] P.O.M. Scokaert and D.Q. Mayne, Min-max feedback model predictive control, IEEE Transactions on Automatic Control 43:8 (1998) 11361142.

Multistage Scenario-based MPC

- Propagates future evolution of uncertainty using a discrete scenario tree.
- Optimize over different control trajectories[1]
- Closed-loop optimization - introduces recourse action.

[1] P.O.M. Scokaert and D.Q. Mayne, Min-max feedback model predictive control, IEEE Transactions on Automatic Control 43:8 (1998) 11361142.

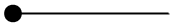
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory

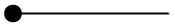


State trajectory

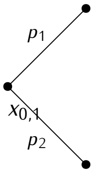
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory



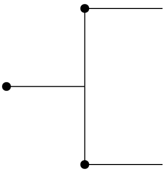
State trajectory



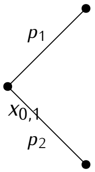
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory



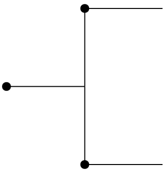
State trajectory



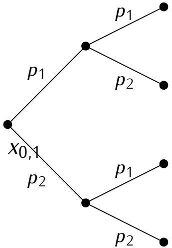
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory



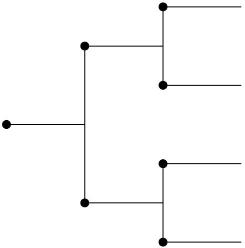
State trajectory



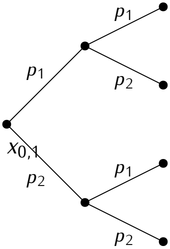
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory



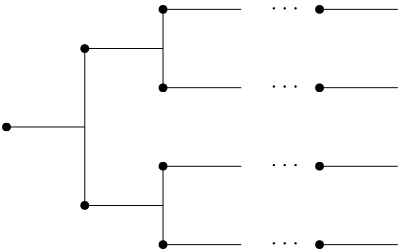
State trajectory



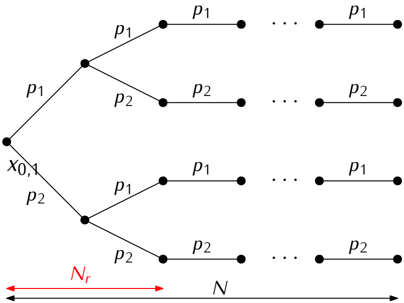
Scenario Tree Generation

$p \in \{p_1, p_2\}$

Control trajectory

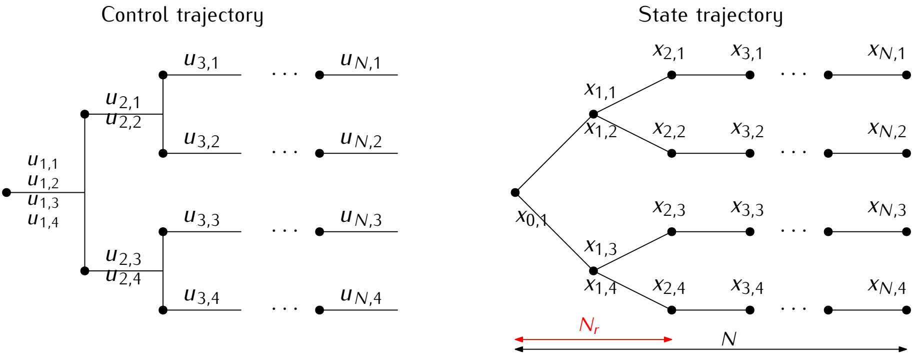


State trajectory

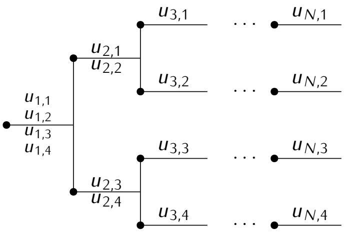


Scenario Tree Generation

$p \in \{p_1, p_2\}$

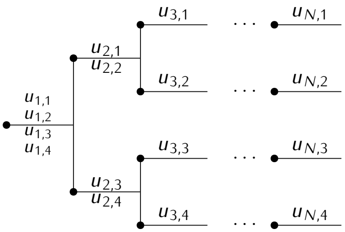


Non-anticipativity constraints



$$u_{1,1} = u_{1,2} = u_{1,3} = u_{1,4}$$
$$u_{2,1} = u_{2,2}$$
$$u_{2,3} = u_{2,4}$$

Non-anticipativity constraints



$$u_{1,1} = u_{1,2} = u_{1,3} = u_{1,4}$$
$$u_{2,1} = u_{2,2}$$
$$u_{2,3} = u_{2,4}$$

Non-anticipativity constraints enable closed-loop implementation!

Multistage Scenario-based MPC

Mathematical formulation

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{j=1}^{S} \omega_j \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j})$$

s.t

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j})$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = 0$$

$$\forall j \in \{1, \cdots, S\}$$

$$\forall k \in \{1, \cdots, N\}$$

Multistage Scenario-based MPC

Mathematical formulation

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{j=1}^{S} \omega_j \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j})$$

s.t

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j})$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = 0$$

$$\forall j \in \{1, \cdots, S\}$$
$$\forall k \in \{1, \cdots, N\}$$

$$\bar{\mathbf{E}} = \left[\begin{array}{c|c|c|c|c} E_{1,2} & -E_{1,2} & & & \\ & E_{2,3} & -E_{2,3} & & \\ & & \ddots & \ddots & \\ & & & E_{S-1,S} & -E_{S-1,S} \end{array}\right]$$

$$E_{j,j+1} = \left[\begin{array}{ccc|ccc} I_{n_u} & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & I_{n_u} & 0 & \cdots & 0 \end{array}\right]$$

$$E_{j,j+1} \in \mathbb{R}^{n_u n_{o,(j,j+1)} \times n_u N}$$

Addressing problem size

Mathematical formulation

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{j=1}^{S} \omega_j \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j})$$

s.t

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\boldsymbol{p}_{k,j})$$

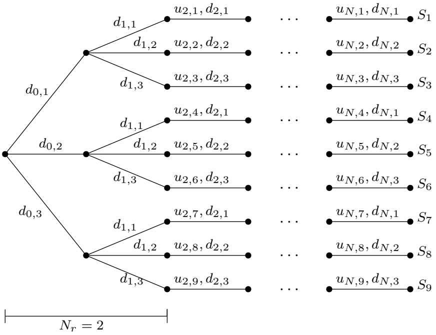
$$\mathbf{g}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\boldsymbol{p}_{k,j}) \leq 0$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = 0$$

$$\forall j \in \{1,\cdots,S\}$$

$$\forall k \in \{1,\cdots,N\}$$



Scenario Decomposition

Scenario decomposition using Dual decomposition[2]

Scenario Subproblem

$$\Gamma_j(\lambda) := \min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \omega_j \sum_{k=1}^{N} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) + \lambda^T \bar{\mathbf{E}}_j \mathbf{u}_j$$

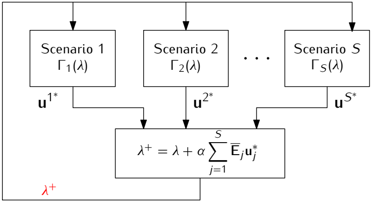
s.t.

$$\mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j})$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}}$$

$$\forall k \in \{1, \cdots, N\}$$

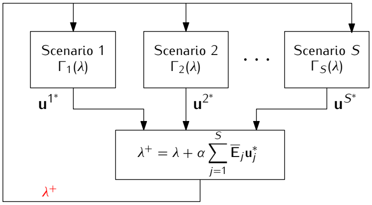


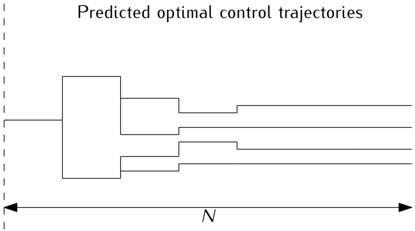
Stopping criteria: $\Delta\lambda = |\lambda^+ - \lambda| \leq \epsilon$

[2] R. Marti, S. Lucia, D. Sarabia, R. Paulen, S. Engell, C. de Prada (2015) Improving scenario decomposition algorithms for robust nonlinear model predictive control, Comput. Chem. Eng. Vol 79 p.30-45

Scenario Decomposition

Non-anticipativity constraints are feasible only upon convergence !

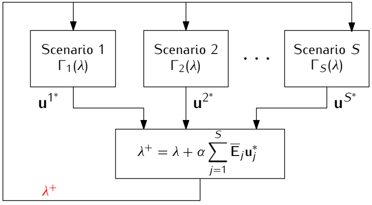


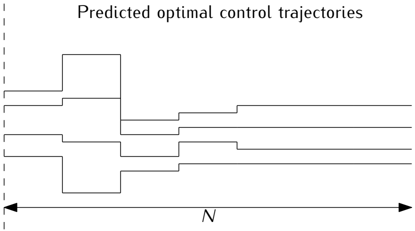


Closed-loop implementation OK

Scenario Decomposition

Non-anticipativity constraints are feasible only upon convergence !

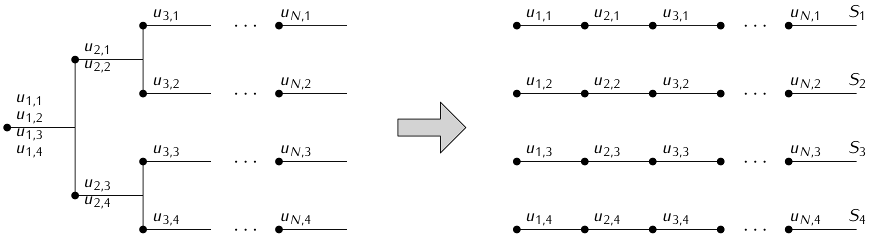




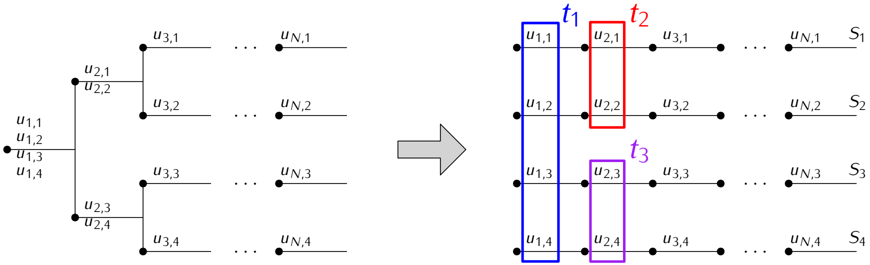
Closed-loop implementation fails!!

"Correctness of computation is a function of time!"

Scenario decomposition using Primal decomposition



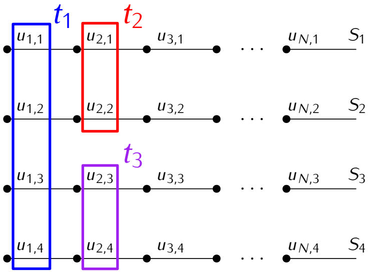
Scenario decomposition using Primal decomposition



Scenario decomposition using Primal decomposition

Non-anticipativity constraint

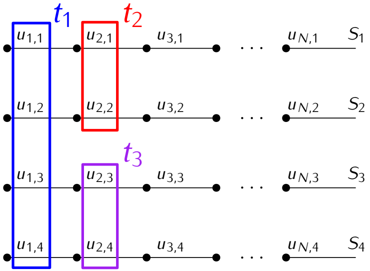
$$u_{1,1} = u_{1,2} = u_{1,3} = u_{1,4} = t_1$$
$$u_{2,1} = u_{2,2} = t_2$$
$$u_{2,3} = u_{2,4} = t_3$$



Scenario decomposition using Primal decomposition

Non-anticipativity constraint

$$u_{1,1} = u_{1,2} = u_{1,3} = u_{1,4} = t_1$$
$$u_{2,1} = u_{2,2} = t_2$$
$$u_{2,3} = u_{2,4} = t_3$$



- Solve each scenario subproblem by fixing $\mathbf{t}$
- Update $\mathbf{t}$ iteratively in the master problem

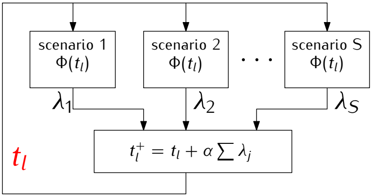
Non-anticipativity constraints always feasible !

Scenario decomposition using Primal decomposition

Scenario Subproblem

$$\boldsymbol{\Phi}(\color{red}{\boldsymbol{t}_l}) = \min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j})$$

$$\begin{aligned}
\text{s.t} \qquad & \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \\
& \mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \le 0 \\
& \mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \\
& \color{red}{\bar{\mathbf{E}}_j \mathbf{u}_j = \bar{\mathbf{t}}} \\
& \quad \forall k \in \{1, \cdots, N\}
\end{aligned}$$



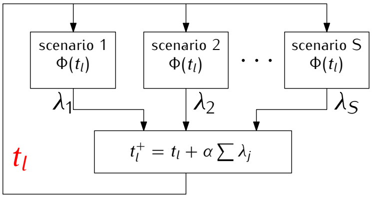
$$t_l \in \mathbb{R}^{n_u}, \ \forall l \in \{1, \ldots, \textstyle\sum_{m=1}^{N_r} M^{m-1}\}$$

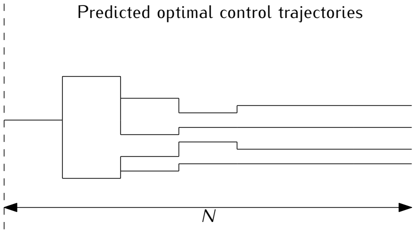
$$\bar{\mathbf{t}} = \left[\begin{array}{c|c|c|c|c} t_{1,2} & -t_{1,2} & & & \\ & t_{2,3} & -t_{2,3} & & \\ & & \ddots & \ddots & \\ & & & t_{S-1,S} & -t_{S-1,S} \end{array} \right] \qquad t_{j,j+1} \in \mathbb{R}^{n_u n_{c,(j,j+1)}}$$

Stopping criteria: $\Delta \boldsymbol{t}_l = |\boldsymbol{t}_l^+ - \boldsymbol{t}_l| \le \epsilon$

Scenario Decomposition using Primal Decomposition

Non-anticipativity constraints are always feasible !

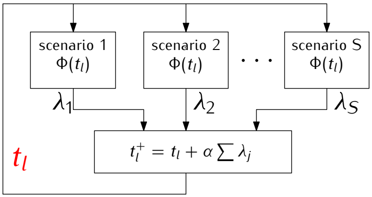


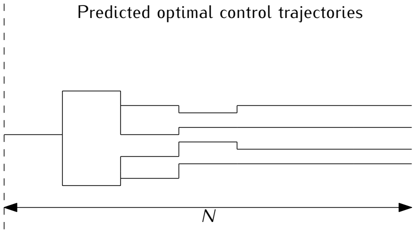


Closed-loop implementation OK

Scenario Decomposition using Primal Decomposition

Non-anticipativity constraints are always feasible !





Closed-loop implementation *still* OK!!

"Approximate solution now is better than accurate solution tomorrow!"

Scenario Decomposition using Primal Decomposition

- Warm-start $\boldsymbol{t}$

[3] Krishnamoorthy, D., Suwartadi, E., Foss, B., Skogestad, S. and Jschke, J., 2018. Improving Scenario Decomposition for Multistage MPC using a Sensitivity-based Path-following Algorithm. IEEE Control Systems Letters, Vol. 2(4).

Scenario Decomposition using Primal Decomposition

- Warm-start $\boldsymbol{t}$
- Suitable step length α to update $\boldsymbol{t}$

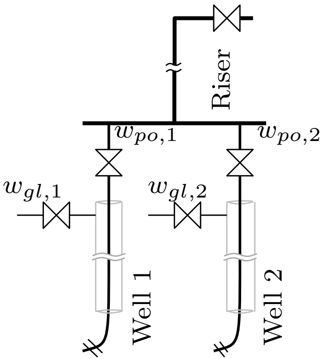
[3] Krishnamoorthy, D., Suwartadi, E., Foss, B., Skogestad, S. and Jschke, J., 2018. Improving Scenario Decomposition for Multistage MPC using a Sensitivity-based Path-following Algorithm. IEEE Control Systems Letters, Vol. 2(4).

Scenario Decomposition using Primal Decomposition

- Warm-start $\boldsymbol{t}$
- Suitable step length α to update $\boldsymbol{t}$
- Solve scenario subproblems using sensitivity-based path-following algorithm [3]

[3] Krishnamoorthy, D., Suwartadi, E., Foss, B., Skogestad, S. and Jschke, J., 2018. Improving Scenario Decomposition for Multistage MPC using a Sensitivity-based Path-following Algorithm. IEEE Control Systems Letters, Vol. 2(4).

Case-study: Gas-lift optimization



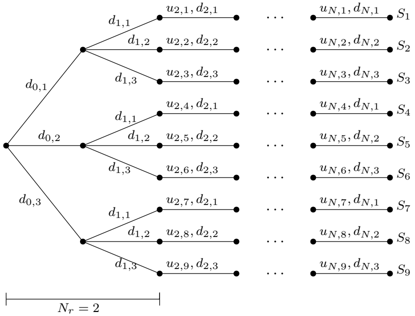
$$\min_{w_{gl}} \sum_{k=1}^{N} \left[-\$_o \sum_{i=1}^{2} w_{po,i_k} + \$_{gl} \sum_{i=1}^{2} w_{gl,i_k} \right]$$

s.t.

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, GOR_{i_k})$$
$$GOR_i \in \{GOR_{0_i} \pm \sigma_i\}$$
$$\forall i \in \{1,2\}, k \in \{1, \cdots, N\}$$

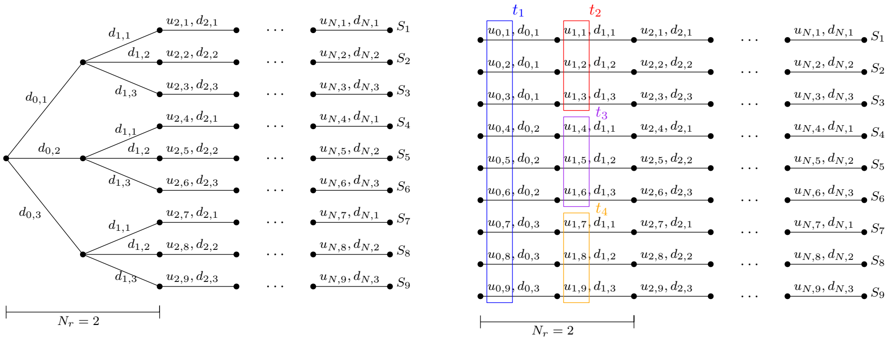
Illustrative Example

$M = 3,\ N_r = 2 \Rightarrow S = M^{N_r} = 9$ scenarios



Illustrative Example

$M = 3,\ N_r = 2 \Rightarrow S = M^{N_r} = 9$ scenarios



Illustrative Example

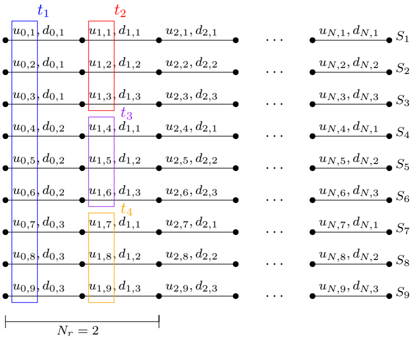
$$\boldsymbol{t}_l = \{\boldsymbol{t}_1^T, \ldots, \boldsymbol{t}_4^T\}$$

$$\boldsymbol{t}_1^+ = \boldsymbol{t}_1 + \sum_{j=1}^{9} \lambda_{1,j}$$

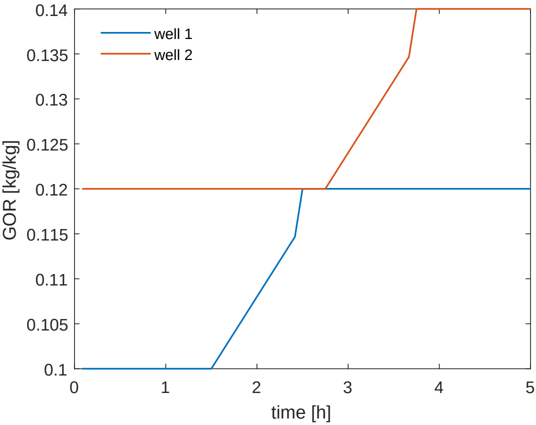
$$\boldsymbol{t}_2^+ = \boldsymbol{t}_2 + \sum_{j=1}^{3} \lambda_{2,j}$$

$$\boldsymbol{t}_3^+ = \boldsymbol{t}_3 + \sum_{j=4}^{6} \lambda_{2,j}$$

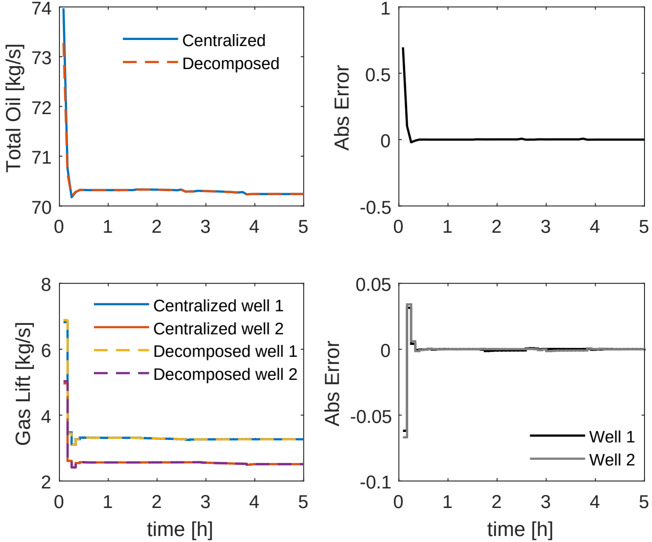
$$\boldsymbol{t}_4^+ = \boldsymbol{t}_4 + \sum_{j=7}^{9} \lambda_{2,j}$$



Simulation Results - Primal Decomposition

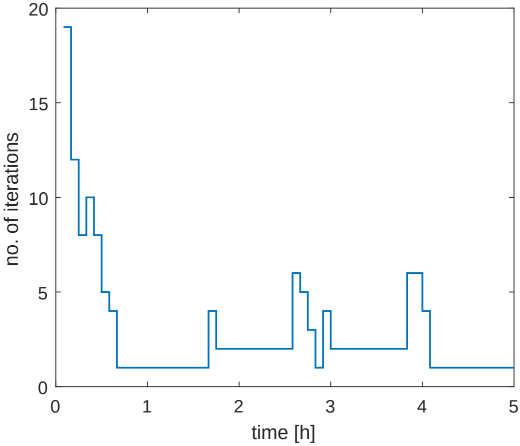


Simulation Results - Primal Decomposition



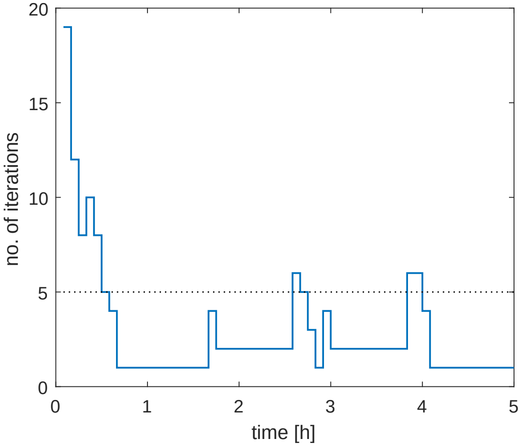
Simulation Results - Primal Decomposition

Number of iterations between the master and subproblems to update $\boldsymbol{t}_l$



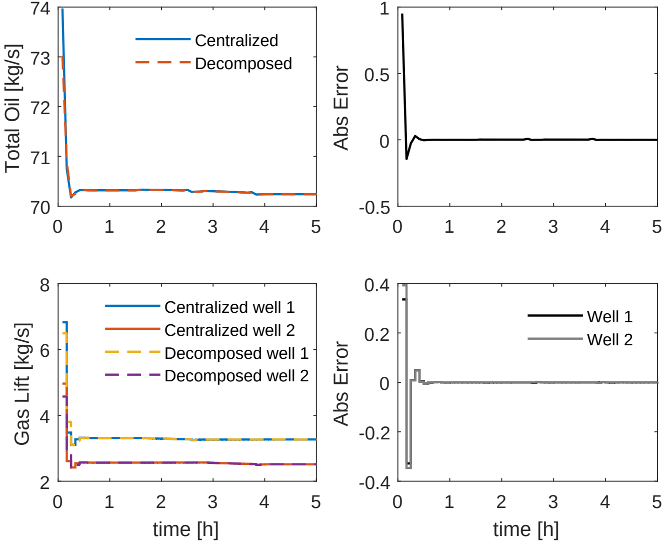
Simulation Results - Primal Decomposition

Number of iterations between the master and subproblems to update $\boldsymbol{t}_l$



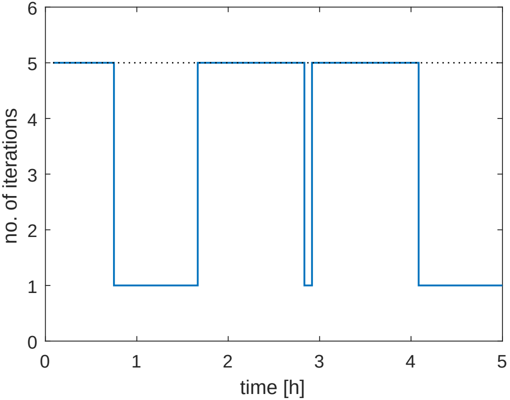
Simulation Results - Primal Decomposition

Number of iterations capped at a maximum of 5 iterations

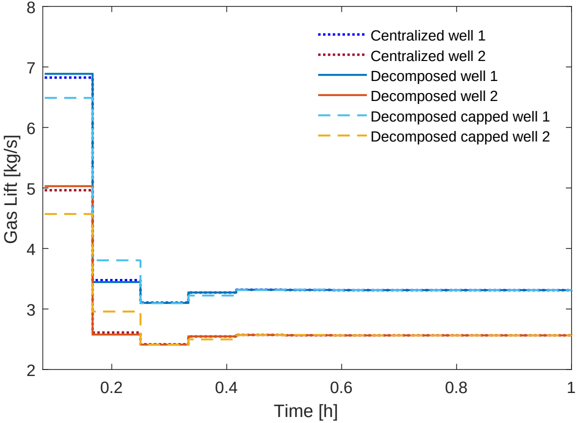


Simulation Results - Primal Decomposition

Number of iterations between the master and subproblems to update $\boldsymbol{t}_l$, capped at a maximum of 5 iterations



Simulation Results - Primal Decomposition



Summary

- Scenario decomposition using primal decomposition

Summary

- Scenario decomposition using primal decomposition
- Primal decomposition always ensures feasibility of non-anticipativity constraints

Summary

- Scenario decomposition using primal decomposition
- Primal decomposition always ensures feasibility of non-anticipativity constraints
- Attractive property for closed-loop implementation

Summary

- Scenario decomposition using primal decomposition
- Primal decomposition always ensures feasibility of non-anticipativity constraints
- Attractive property for closed-loop implementation
- Warm-starting the master problem leads to convergence eventually

Summary

- Scenario decomposition using primal decomposition
- Primal decomposition always ensures feasibility of non-anticipativity constraints
- Attractive property for closed-loop implementation
- Warm-starting the master problem leads to convergence eventually
- Demonstrated using a gas-lift optimization case study

Summary

- Scenario decomposition using primal decomposition
- Primal decomposition always ensures feasibility of non-anticipativity constraints
- Attractive property for closed-loop implementation
- Warm-starting the master problem leads to convergence eventually
- Demonstrated using a gas-lift optimization case study

Thank you !

Acknowledgement: SFI SUBPRO