A Primal decomposition algorithm for distributed multistage scenario model predictive control

Dinesh Krishnamoorthy[a], Bjarne Foss[b], Sigurd Skogestad[a,*]

[a]*Department of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*
[b]*Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

Abstract

This paper proposes a primal decomposition algorithm for efficient computation of multistage scenario model predictive control, where the future evolution of uncertainty is represented by a scenario tree. This often results in large-scale optimization problems. Since the different scenarios are only coupled via the so-called non-anticipativity constraints, which ensures that the first control input is the same for all the scenarios, the different scenarios can be decomposed into smaller subproblems and solved iteratively using a master problem to coordinate the subproblems. We review the most common scenario decomposition methods and argue in favour of primal decomposition algorithms, since it ensures feasibility of the non-anticipativity constraints throughout the iterations which is crucial for closed-loop implementation. We also propose a novel backtracking algorithm to determine a suitable step length in the master problem to ensure the feasibility of the nonlinear constraints. The performance of the proposed approach and the backtracking algorithm is demonstrated using a CSTR case study.

Keywords: model predictive control, primal decomposition, distributed optimization, uncertainty

[☆]The authors gratefully acknowledge the financial support from SUBPRO, which is financed by the Research Council of Norway, major industry partners and NTNU.

[*]Corresponding author
Email addresses: dinesh.krishnamoorthy@ntnu.no (Dinesh Krishnamoorthy), bjarne.foss@ntnu.no (Bjarne Foss), skoge@ntnu.no (Sigurd Skogestad)

1. Introduction

Model predictive control (MPC) is widely used in the process control industry due to its ability to handle multivariable systems subject to state and input constraints. In recent years, there has been an increasing trend in the use of economic objectives in the framework of nonlinear model predictive control, known as economic MPC. In many processes, the optimal economic operation often corresponds to driving the system to some of its constraints. In such cases, model uncertainty and process variations can easily lead to constraint violations. Explicit measures to handle uncertainty in the MPC problem then becomes very important.

Different approaches to handle uncertainty in the MPC problem have been proposed in the literature. For example, the min-max MPC formulation proposed in [1] computes an optimal input trajectory that minimizes the cost of the worst-case uncertainty realization. This often results in very conservative solutions, since it ignores the fact that new information will be made available in the future and a new control input trajectory will be re-computed. In other words, the notion of feedback is ignored when computing a single control trajectory that optimizes over all possible realizations of the uncertainty. A feedback min-max MPC framework, also known as multistage scenario MPC was introduced in [2] and later in [3], where the future evolution of uncertainty is represented using discrete scenarios and the notion of feedback is explicitly taken into account by optimizing over different control trajectories instead of a single control trajectory (closed-loop optimization).

One of the main drawbacks of this approach is that the problem size grows exponentially with 1) the number of uncertain parameters, 2) the number of finite realizations of the uncertainty, and 3) the length of the prediction horizon. As justified in [3], one way to curb the exponential growth of the problem size is by considering the branching of the tree only up to a certain number of samples in the prediction horizon, known as robust horizon. Another solu-

tion is to exploit the structure of the problem to decompose the problem into several smaller subproblems. The different scenarios are independent and additively separable, except for the non-anticipativity constraints, which capture the effect that the optimal inputs cannot anticipate the future realization of the uncertainty. To this end, decomposition methods can be used to solve the different scenarios independently and later use a master problem to iteratively co-ordinate the different subproblems.

Scenario decomposition using dual decomposition methods were proposed in [4] and [5], where the different subproblems are solved by relaxing the non-anticipativity constraints. A master problem then updates the Lagrangian multipliers corresponding to the non-anticipativity constraints iteratively. The non-anticipativity constraints are thus feasible only upon convergence of the master and subproblem iterations. Dual decomposition methods may require a relatively large number of iterations between the master problem and the subproblems to converge. This leads to challenges with practical implementation as noted in [5]. If the iterations between the master problem and the subproblems do not converge within the required sample time of the MPC, the non-anticipativity constraints remains infeasible. As a result, the different scenarios may give different optimal control inputs at the first sample, which is not acceptable for real-time closed-loop implementation.

To overcome the closed-loop implementation issue with dual decomposition, we recently proposed a primal decomposition algorithm for scenario decomposition in [6]. In contrast to dual decomposition, primal decomposition produces a primal feasible solution with monotonically decreasing objective value at each iteration [7]. Primal decomposition thus ensures that the non-anticipativity constraints are always feasible through out the iterations. This implies that even if the master problem and subproblem iterations are prematurely terminated, the non-anticipativity constraints are still feasible and the first control input provided by all the scenarios are the same. This is an important property for closed-loop implementation of the multistage scenario MPC problem.

This paper extends the conference paper in [6] with a more detailed dis-

cussion and introduces a novel back-tracking algorithm to select a suitable step-length in the master problem update. The proposed primal decomposition approach and the novel backtracking algorithm are demonstrated using a continuously stirred tank reactor (CSTR) case study.

The remainder of the paper is organized as follows. The multistage scenario MPC framework is described in Section 2. Section 3 describes the different decomposition approaches for scenario decomposition. Section 4 introduces a novel backtracking algorithm for choosing a suitable step length in the master problem. The proposed method is then demonstrated using a CSTR case study in Section 5. Finally, discussions are provided in Section 6 before concluding the paper in Section 7

2. Multistage Scenario MPC

Consider a discrete-time nonlinear system of the form

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

where $\mathbf{x} \in \mathbb{R}^{n_x}$ denotes the states, $\mathbf{u} \in \mathbb{R}^{n_u}$ denotes the manipulated inputs and $\boldsymbol{p} \in \mathbb{R}^{n_p}$ denotes the vector the uncertain parameters or disturbances with an *a-priori* known distribution $\boldsymbol{p} \in \mathcal{U}$. The system model is represented by $\mathbf{f} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_x}$.

The uncertainty space $\mathcal{U}$ is discretized to get M finite realizations of (1). The future evolution of the uncertainty in the prediction horizon is then represented by a scenario tree as shown in Fig.1, where a scenario is represented by the path from the root node to a leaf node [3]. As mentioned earlier, to curb the exponential growth of the problem size, the branching is stopped after a certain number of time samples, known as robust horizon N_r, after which the uncertain parameters are treated as constants. Consequently, the number of scenarios and the number of MPC problems to be solved, is given by $S = M^{N_r}$. Allowing for the different cost weights ω_j to represent the likeliness of the different scenarios

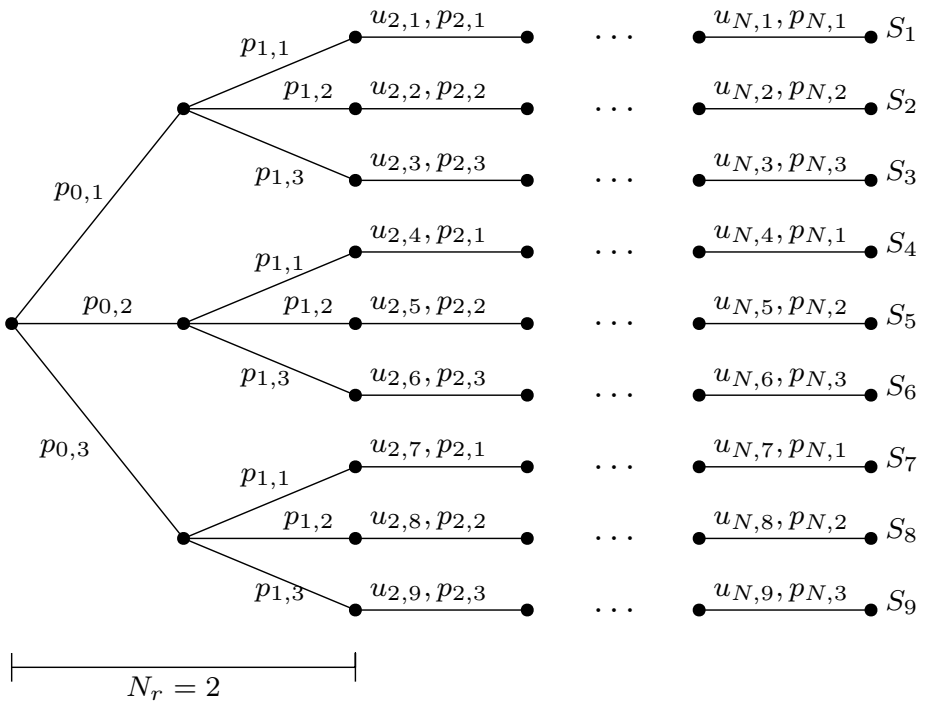

Figure 1: Schematic representation of a scenario tree with $M = 3$ and $N_r = 2$ leading to $S = 9$ scenarios.

j, the resulting dynamic optimization problem can be formulated as,

$$\min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \sum_{j=1}^{S} \omega_j \left[\sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \right] \tag{2a}$$

$$\text{s.t} \quad \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \tag{2b}$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0 \tag{2c}$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \tag{2d}$$

$$\sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j = \mathbf{0} \tag{2e}$$

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

$$\forall k \in \{0, \cdots, N-1\}$$

where the subscript $(\cdot)_{k,j}$ represents the j^{th} scenario at time sample k. The cost function is given by $\mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j})$ and $\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j})$ represents the nonlinear inequality constraints. Initial condition are enforced in (2d) for all the scenarios, where $\hat{\boldsymbol{x}}$ denotes the state measurements or estimates at the current time step. (2e) enforces the non-anticipativity constraints, which ensures that the states that branch at the same parent node, has the same control input. Note that $\mathbf{u}_j$ here represents the sequence of optimal control input for the j^{th} scenario, i.e. $\mathbf{u}_j = \left[\mathbf{u}_{0,j}^T \cdots \mathbf{u}_{N-1,j}^T\right]^T \in \mathbb{R}^{n_u N}$ and $\bar{\mathbf{E}}_j$ is given by

$$\begin{aligned} \bar{\mathbf{E}} &= \left[\begin{array}{c|c|c|c|c} E_{1,2} & -E_{1,2} & & & \\ & E_{2,3} & -E_{2,3} & & \\ & & \ddots & \ddots & \\ & & & E_{S-1,S} & -E_{S-1,S} \end{array}\right] \\ &= \left[\begin{array}{c|c|c|c} \bar{\mathbf{E}}_1 & \bar{\mathbf{E}}_2 & \cdots & \bar{\mathbf{E}}_S \end{array}\right] \end{aligned} \tag{3}$$

where,

$$E_{j,j+1} = \left[\begin{array}{ccc|ccc} I_{n_u} & & & 0 & \cdots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & I_{n_u} & 0 & \cdots & 0 \end{array}\right] \tag{4}$$

If $n_{o,(j,j+1)}$ denotes the number of common nodes between two consecutive scenarios j and $j+1$, then $E_{j,j+1} \in \mathbb{R}^{r \times n_u N}$ and $\bar{\mathbf{E}}_j \in \mathbb{R}^{q \times n_u N}$ where,

$$q = n_u \sum_{j=1}^{S} n_{o,(j,j+1)} \quad \text{and} \quad r = n_u n_{o,(j,j+1)}$$

as described in [8]. Formulating the non-anticipativity constraints using this chain structure also results in sparse structures, which may be exploited by many solvers [8].

3. Distributed Multistage Scenario MPC

The multistage scenario MPC problem (2) consists of S independent MPC problems, except for the non-anticipativity constraints (2e) which couple the different scenarios together. Different decomposition approaches can be used to solve the different scenarios (in parallel) and use a master problem to co-ordinate the different subproblems.

3.1. Dual Decomposition based approaches

Scenario decomposition using dual decomposition approaches is the most common strategy. Here each scenario subproblem is solved by relaxing the non-anticipativity constraints, see for example [4], [5] and [8].

The scenario optimization problem (2) can be rewritten as,

$$\begin{aligned}
\min_{\mathbf{x}_{k,j}, \mathbf{u}_{k,j}} \quad & \sum_{j=1}^{S} \left[\omega_j \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \right] + \boldsymbol{\lambda}^T \sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}_j \\
\text{s.t.} \quad & \\
& \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \\
& \mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0 \\
& \mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \\
& \forall j \in \{1, \cdots, S\}, \forall k \in \{0, \cdots, N-1\}
\end{aligned} \tag{5}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^q$ is the Lagrange multiplier corresponding to the non-anticipativity constraint (2e). It can be seen that (5) is now additively separable in $\mathbf{x}$ and

$\mathbf{u}$ and each j^{th} scenario subproblem can be reformulated as a function of $\boldsymbol{\lambda}$ as shown below,

$$\begin{aligned}\Gamma_j(\boldsymbol{\lambda}) := \min_{\mathbf{x}_{k,j},\mathbf{u}_{k,j}} \quad & \omega_j \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j},\mathbf{u}_{k,j}) + \boldsymbol{\lambda}^T \bar{\mathbf{E}}_j \mathbf{u}_j \\ \text{s.t.} \\ & \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\boldsymbol{p}_{k,j}) \\ & \mathbf{g}(\mathbf{x}_{k,j},\mathbf{u}_{k,j},\boldsymbol{p}_{k,j}) \leq 0 \\ & \mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \\ & \forall k \in \{0,\cdots,N-1\}\end{aligned} \tag{6}$$

The Lagrange multiplier $\boldsymbol{\lambda}$ is iteratively updated in the master problem and the non-anticipativity constraints becomes feasible upon convergence of $\boldsymbol{\lambda}$.

The master problem is then given by

$$\min_{\boldsymbol{\lambda}} \quad \sum_{j=1}^{S} \Gamma_j(\boldsymbol{\lambda}) \tag{7}$$

Proposition 1. *Let $(\mathbf{x}^*_{k,j},\mathbf{u}^*_{k,j})$ be the optimal solution for the j^{th} subproblem, then a solution to the master problem (7) can be expressed as the gradient descent step,*

$$\boldsymbol{\lambda}^+ = \boldsymbol{\lambda} + \alpha \sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}^*_j \tag{8}$$

where $\boldsymbol{\lambda}^+$ represents the updated lagrange multiplier for the next iteration and α is a suitable step length.

Proof. The descent direction for the master problem (7) is given by the subgradient $\sum_{j=1}^{S} \nabla_{\boldsymbol{\lambda}} \Gamma_j(\boldsymbol{\lambda}) = \sum_{j=1}^{S} \bar{\mathbf{E}}_j \mathbf{u}^*_j$. The solution to the master problem along the descent direction using a suitable step length α can then be expressed by (8), see [9] and [7]. □

Different forms of augmented Lagrangian decomposition methods were also presented in [5], where an additional quadratic penalty term is added to (6) to improve the convergence properties. However, the additional quadratic penalty

terms makes the different subproblems nonseparable in $\mathbf{x}$ and $\mathbf{u}$ and hence cannot be solved in parallel. In such cases the subproblems must be solved sequentially using the alternating directions method of multipliers (ADMM) approach [10].

However, the main challenge of dual decomposition approach is that, relaxing the non-anticipativity constraints may impede real-time closed-loop implementation. In the receding horizon control framework, at each time step, the first control move is implemented in the plant. In multistage scenario MPC, the non-anticipativity constraints ensure that the first control move is equal for all the scenarios to enable closed-loop implementation. However, if the master problem and subproblems fail to converge within the required sampling time, the non-anticipativity constraints are not satisfied. Consequently, the first control input computed by the different scenarios are different, thus impeding closed-loop implementation.

One way to address this issue is to take a weighted average of the manipulated inputs at the first sample based on the probabilities of the different scenarios [11]. However, this may not be a good approach since the weighted average can lead to an infeasible solution. The authors in [5] proposed to compute an average of the control inputs at the first sample such that the worst-case constraint violation for the local subproblems is minimized, which is given by solving an additional linear programming (LP) problem. In this paper, we instead propose a primal decomposition approach to solve this issue, which always ensures the feasibility of the non-anticipativity constraints.

3.2. Primal Decomposition based approaches

To address the issue of non-anticipativity constraint feasibility, we propose a primal decomposition algorithm, which always produces a primal feasible point by iterating directly on the shared variables [7]. Therefore, at any point in time, the non-anticipativity constraints are always feasible and the first control move provided by all the scenarios are the same, which is a desirable property for closed-loop implementation as described earlier.

The primal subproblem for the j^{th} subproblem can be written by introducing a new auxiliary variable $\mathbf{t}_l \in \mathbb{R}^{n_u}$, $\forall l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{m-1}\}$ for each non-anticipativity constraints. Note that the number of non-anticipativity constraints is given by $\sum_{m=1}^{N_r} M^{m-1}$. Each scenario subproblem is then expressed as a function of the auxiliary variables as shown below,

$$\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j) = \min_{\mathbf{x}_{k,j}, \mathbf{u}_{k,j}} \sum_{k=0}^{N-1} \mathbf{J}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}) \tag{9a}$$

$$\text{s.t} \qquad \mathbf{x}_{k+1,j} = \mathbf{f}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \tag{9b}$$

$$\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0 \tag{9c}$$

$$\mathbf{x}_{0,j} = \hat{\boldsymbol{x}} \tag{9d}$$

$$\bar{\mathbf{E}}_j \mathbf{u}_j = \bar{\boldsymbol{\tau}}_j \qquad \forall k \in \{0, \cdots, N-1\} \tag{9e}$$

where $\bar{\boldsymbol{\tau}}$ is given by

$$\bar{\boldsymbol{\tau}} = \left[\begin{array}{c|c|c|c|c} \tau_{1,2} & -\tau_{1,2} & & & \\ & \tau_{2,3} & -\tau_{2,3} & & \\ & & \ddots & \ddots & \\ & & & \tau_{S-1,S} & -\tau_{S-1,S} \end{array}\right] \tag{10}$$

$$= \left[\begin{array}{c|c|c|c} \bar{\boldsymbol{\tau}}_1 & \bar{\boldsymbol{\tau}}_2 & \cdots & \bar{\boldsymbol{\tau}}_S \end{array}\right] \tag{11}$$

and $\tau_{j,j+1} \in \mathbb{R}^{n_u n_{o,(j,j+1)}}$ is a matrix that is composed of the auxiliary variables $\boldsymbol{t}_l \in \mathbb{R}^{n_u}$.

The master problem to update the auxiliary variables $\boldsymbol{t}_l$ is then given by,

$$\min_{\boldsymbol{t}_l} \sum_{j=1}^{S} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j) \tag{12}$$

Proposition 2. *The solution to the master problem* (12) *can be expressed as the gradient descent step,*

$$\boldsymbol{t}_l^+ = \boldsymbol{t}_l + \alpha_l (\sum_{j=1}^{S} \nabla_{\boldsymbol{t}_l} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)), \qquad \forall l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{m-1}\} \tag{13}$$

Proof. The sesarch direction for the master problem (12) is given by the subgradient $\sigma = \sum_{j=1}^{S} \nabla_{\boldsymbol{t}_l} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$, which are simply the lagrange multipliers that

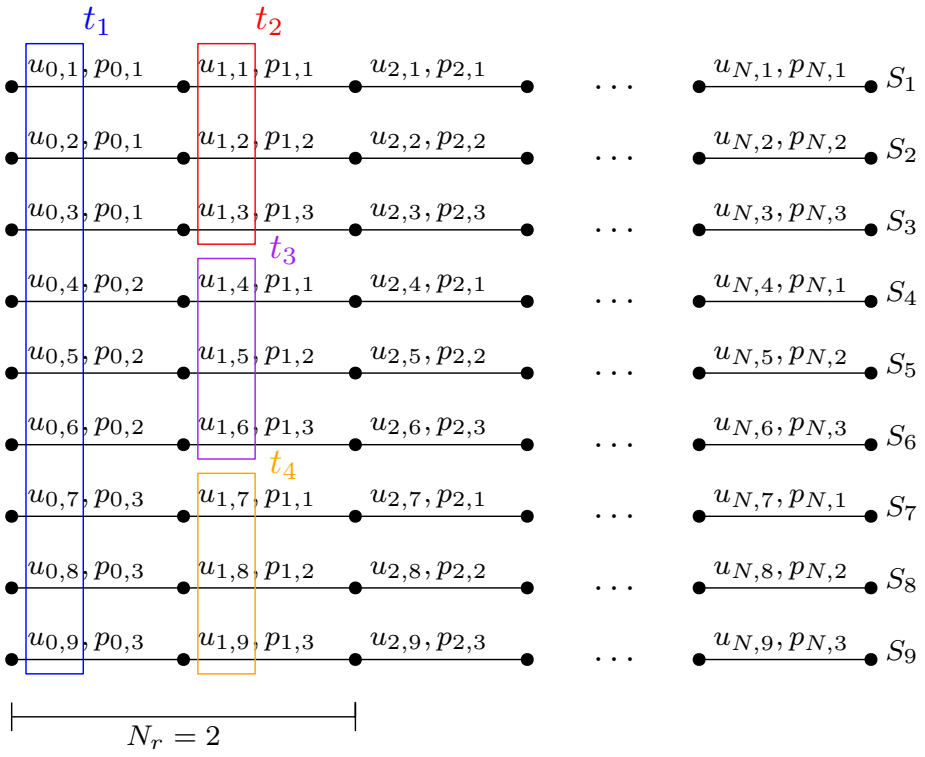

Figure 2: Schematic representation of the decomposed scenarios showing the non-anticipativity constraints enforced using the auxiliary variables $\boldsymbol{t}_l$.

corresponds to the non-anticipativity constraints (9e), which are computed by solving the different scenario subproblems [9]. Hence the subgradient for the master problem is essentially available for "free" without the need for any additional computations.

The solution to the master problem along the descent direction with a suitable step length α can then be given by (13), see [7] and [9]. □

One commonly used stopping criteria for the master problem and subproblem iterations is that the change in $\boldsymbol{t}_l$ between two subsequent iterations, denoted by $\Delta \boldsymbol{t}_l = \|\boldsymbol{t}_l^+ - \boldsymbol{t}_l\|$ must be less than a certain user-defined tolerance ϵ.

To illustrate this, consider a scenario tree with $M = 3$, $N_r = 2$ and $S = 9$

as shown in Fig.1. For such a tree $l = 4$ and $\boldsymbol{t}_l \in \{\boldsymbol{t}_1^T, \boldsymbol{t}_2^T, \boldsymbol{t}_3^T, \boldsymbol{t}_4^T\}$. The non-anticipativity constraints for the scenario tree is such that the control input for all the scenarios at the first control sample is the same.

$$\mathbf{u}_{1,j} = \mathbf{u}_{1,j+1} = \boldsymbol{t}_1, \qquad \forall j \in \{1, \ldots, S-1\} \tag{14}$$

where $S - 1 = 8$ in this case.

The non-anticipativity constraints at the second time sample is then given by

$$\begin{aligned} \mathbf{u}_{2,1} = \mathbf{u}_{2,2} = \mathbf{u}_{2,3} = \boldsymbol{t}_2 \\ \mathbf{u}_{2,4} = \mathbf{u}_{2,5} = \mathbf{u}_{2,6} = \boldsymbol{t}_3 \\ \mathbf{u}_{2,7} = \mathbf{u}_{2,8} = \mathbf{u}_{2,9} = \boldsymbol{t}_4 \end{aligned} \tag{15}$$

This is schematically represented in Fig.2. The auxiliary variables are then updated in the master problem using the gradient descent step (13),

$$\boldsymbol{t}_1^+ = \boldsymbol{t}_1 + \alpha_1 \sum_{j=1}^{9} \nabla_{\boldsymbol{t}_1} \Phi$$

where $\sum_{j=1}^{9} \nabla_{\boldsymbol{t}_1} \Phi = \sum_{j=1}^{9} \lambda_{1,j}$

$$\boldsymbol{t}_2^+ = \boldsymbol{t}_2 + \alpha_2 \sum_{j=1}^{9} \nabla_{\boldsymbol{t}_2} \Phi$$

where $\sum_{j=1}^{9} \nabla_{\boldsymbol{t}_2} \Phi = \lambda_{2,1} + \lambda_{2,2} + \lambda_{2,3}$

$$\boldsymbol{t}_3^+ = \boldsymbol{t}_3 + \alpha_3 \sum_{j=1}^{9} \nabla_{\boldsymbol{t}_3} \Phi$$

where $\sum_{j=1}^{9} \nabla_{\boldsymbol{t}_3} \Phi = \lambda_{2,4} + \lambda_{2,5} + \lambda_{2,6}$

$$\boldsymbol{t}_4^+ = \boldsymbol{t}_4 + \alpha_4 \sum_{j=1}^{9} \nabla_{\boldsymbol{t}_4} \Phi$$

where $\sum_{j=1}^{9} \nabla_{\boldsymbol{t}_4} \Phi = \lambda_{2,7} + \lambda_{2,8} + \lambda_{2,9}$, and the notation $\lambda_{k,j}$ denotes the Lagrange multipliers corresponding to the the non-anticipativity constraints at time step k for the j^{th} scenario.

4. Back-tracking algorithm

As mentioned in the previous section, when using primal decomposition, we solve the subproblems by fixing the manipulated inputs for the non-anticipativity constraints to be equal to an auxiliary variable $\boldsymbol{t}_l$ for all the scenarios by means of the equality constraint (9e). The auxiliary variable $\boldsymbol{t}_l$ is then iteratively updated using a step length α as shown in (13). The equality constraint (9e) ensures that the non-anticipativity constraints are always feasible throughout the iterations. However, if the step length α is not suitably chosen, then the nonlinear constraints $\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0$ may become infeasible by fixing the control input at $\boldsymbol{t}_l^+$ using the equality constraint (9e). Choosing a step length too small on the other hand leads to a very slow convergence. Hence, careful selection of the step length α is important in the presence of nonlinear constraints.

Therefore, in this paper we propose a feasibility ensuring backtracking algorithm to suitably choose the step length α, such that we can choose a sufficiently large step length and backtrack when required to ensure that the nonlinear constraints $\mathbf{g}(\mathbf{x}_{k,j}, \mathbf{u}_{k,j}, \boldsymbol{p}_{k,j}) \leq 0$ remain feasible throughout iterations. The proposed backtracking algorithm is based on a forward integration of the system dynamics and the nonlinear constraints one time step ahead using the proposed step length. In other words, a one-step-ahead model prediction for each scenario using the prospective $\boldsymbol{t}_l^+$ is used to evaluate the nonlinear constraint feasibility before $\boldsymbol{t}_l^+$ is fixed in the subproblems using the equality constraint (9e) in the next iteration. The prospective $\boldsymbol{t}_l^+$ is given by $\boldsymbol{t}_l^+ = \boldsymbol{t}_l + \alpha_l \sigma$, where $\sigma = (\sum_{j=1}^{S} \nabla_{\boldsymbol{t}_l} \boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j))$ is the search direction or the subgradient. The step length α is backtracked until the nonlinear constraints in the one-step-ahead model prediction is feasible. This is illustrated in Algorithm 1.

In order to show that the proposed backtracking algorithm converges, assume that a feasible point $\boldsymbol{t} \in \mathcal{F}$ is available. As shown in [12, Ch.9], the subgradient σ is feasible by construction. Hence there is an $\overline{\alpha}$ such that,

$$\boldsymbol{t} + \alpha\sigma \in \mathcal{F} \quad \forall 0 < \alpha \leq \overline{\alpha} \tag{16}$$

The task of the feasibility ensuring backtracking algorithm is then to find this

Algorithm 1 Feasibility-ensuring backtracking algorithm

Define $c < 1$.

Input: at each iteration between master problem and subproblem: initial state $\hat{\boldsymbol{x}}$, initial α_0, $\boldsymbol{t}$ and subgradient $\sigma = \sum_{j=1}^{S} \nabla_{\boldsymbol{t}} \boldsymbol{\Phi}(\boldsymbol{t}, \boldsymbol{p}_j)$

for $j = 1, 2, \ldots, S$ **do**

$\alpha \leftarrow \alpha_0$

Evaluate $\mathbf{x}_{k+1,j} = \mathbf{f}(\hat{\boldsymbol{x}}, (\boldsymbol{t} + \alpha\sigma), \boldsymbol{p}_{k,j})$,

$\forall k \in \{1, \ldots, N_r\}$

while $\mathbf{g}(\mathbf{x}_{k+1,j}, (\boldsymbol{t} + \alpha\sigma), \boldsymbol{p}_{k,j}) > 0$ **do**

$\alpha \leftarrow c\alpha$

end while

end for

Output: α

upper bound $\overline{\alpha}$ using Algorithm 1 such that the updated value $\boldsymbol{t}^+ = \boldsymbol{t} + \alpha\sigma$ is feasible, i.e. $\boldsymbol{t}^+ \in \mathcal{F}$. The availablity of an initial feasible guess for $\boldsymbol{t}$ is discussed in Section 6.2.

However, the backtracking algorithm proposed here backtracks the step length alpha based on the *constraint* evaluation to ensure that the nonlinear constraints are feasible.

The sketch of the proposed primal decomposition algorithm for multistage scenario MPC problem using the feasibility ensuring backtracking algorithm is given in Algorithm 2.

5. Case Study

In this section, we test the proposed primal decomposition-based distributed multistage scenario MPC on a continuous stirred tank reactor (CSTR) process from [13] and [14]. Consider a reversible exothermic reaction where component

Algorithm 2 Distributed multistage scenario MPC using Primal decomposition

Define tolerance $\epsilon > 0$.

Input: at each time step: initial state $\hat{\boldsymbol{x}}$ and $\Delta \boldsymbol{t}_l > \epsilon$, initial α

---

while $\Delta \boldsymbol{t}_l > \epsilon$ **do**

  for $j = 1, 2, \ldots, S$ **do**

    $[\boldsymbol{X}^*, \boldsymbol{\lambda}^*] \leftarrow$ solution NLP $\boldsymbol{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$

  end for

  for $l \in \{1, \ldots, \sum_{m=1}^{N_r} M^{N_r - 1}\}$ **do**

    Update subgradients $\nabla_{\boldsymbol{t}_l} \Phi(\boldsymbol{t}, \boldsymbol{p}_j)$

    Backtrack α using Algorithm 1

    Update $\boldsymbol{t}_l^+ = \boldsymbol{t}_l + \alpha(\nabla_{\boldsymbol{t}} \boldsymbol{\Phi}(\boldsymbol{t}, \boldsymbol{p}_j))$

    Update $\Delta \boldsymbol{t}_l = \|\boldsymbol{t}_l^+ - \boldsymbol{t}_l\|$

  end for

end while

Reset α to initial guess.

Output: $\boldsymbol{X}^*(\boldsymbol{p}_j), \forall j \in \{1, \ldots, S\}$

A is converted to component B

$$A \rightleftharpoons B \tag{17}$$

The reaction rate is given as $r = k_1 C_A - k_2 C_B$ where $k_1 = C_1 e^{\frac{-E_1}{RT}}$ and $k_2 = C_2 e^{\frac{-E_2}{RT}}$. The process model consists of two mass balances and an energy balance:

$$\frac{dC_A}{dt} = \frac{1}{\tau}(C_{A,i} - C_A) - r \quad \text{where} \quad \tau = \frac{H}{F} \tag{18a}$$

$$\frac{dC_B}{dt} = \frac{1}{\tau}(C_{B,i} - C_B) + r \tag{18b}$$

$$\frac{dT}{dt} = \frac{1}{\tau}(T_i - T) + \frac{-\Delta H_{rx}}{\rho C_p} r \tag{18c}$$

where the concentrations $[mol/l]$of the two components in the reactor are denoted by C_A and C_B, respectively. $C_{A,i}$ and $C_{B,i}$ denote the feed concentrations. T_i and T are the inlet and reaction temperatures, respectively. The model parameters are given in Table 1.

Table 1: Nominal values for CSTR process

	Description	Value	Unit
F^*	Feed rate	1	mol/min
C_1	Constant	5000	s^{-1}
C_2	Constant	10^6	s^{-1}
C_p	Heat capacity	1000	$cal/kg/K$
E_1	Activation energy	10^4	cal/mol
E_2	Activation energy	15000	cal/mol
$C^*_{A,i}$	Inlet A concentration	1	mol/l
$C^*_{B,i}$	Inlet B concentration	0	mol/l
R	Universal Gas Constant	1.987	$cal/mol/K$
ΔH_{rx}	Heat of reaction	-5000	cal/mol
ρ	Density	1	kg/l
H	reactor holdup	1	mol
τ	Time constant	60	s

The objective is to maximize the product concentration C_B while penalizing the utility cost of heating the input stream using the inlet temperature $\mathbf{u} = T_i$ as

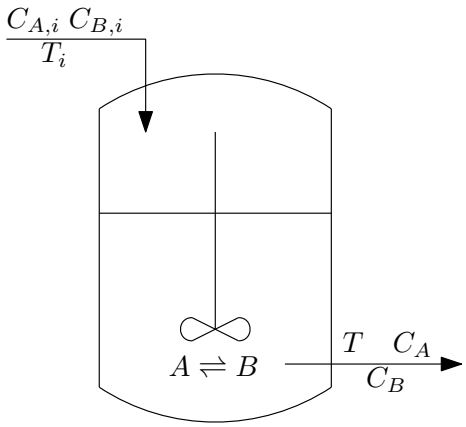

Figure 3: Case 1: Exothermic reactor process

the manipulated variable. In addition, the reactor temperature has a maximum limit of 425K.

$$\min_{T_i} J = -[2.009C_B - (1.657 \times 10^{-3}T_i)^2]$$

$$s.t. \qquad (18) \qquad (19)$$

$$T \leq 425$$

We assume that the concentration of component B in the feed stream is uncertain and is known to vary in the range $C_{B,i} \in [0, 0.2]mol/l$. We design a multistage scenario MPC with a prediction horizon of $T = 300$s equally divided into $N = 20$ samples. For the scenario tree, we consider $M = 3$ discrete realizations of the uncertainty, namely, $C_{B,i} \in \{0.0, 0.1, 0.2\}mol/l$. The system dynamics (18) are discretized using third-order direct collocation. The multistage scenario MPC was implemented in `CasADi` version 3.1.0 [15] using the `MATLAB` programming environment. The resulting nonlinear programming problem was solved using `IPOPT` version 3.12.2 running with a `MUMPS` linear solver. The simulations were carried out on a 2.6GHz workstation with 16GB memory. The plant simulator was implemented using the `IDAS` integrator [16].

5.1. Simulation case 1

In the first simulation case, we design the multistage MPC with a robust horizon of $N_r = 1$, leading to $S = M^{N_r} = 3$ scenarios. The process was simulated for a total simulation time of 600s. The concentration of component B in the feed stream changes at time $t = 300$s from $0.15mol/l$ to $0.0mol/l$. The process was first simulated with a fully centralized multistage scenario MPC problem (2) to be used as a benchmark (shown in thick yellow curves in Fig.4).

The process was then simulated using the proposed primal decomposition-based distributed multistage scenario MPC (9), where the step length α_l was initialized with $\alpha = 2000$. At each iteration, the step length α was suitably adjusted using the proposed back-tracking algorithm introduced in Section 4. At each time step, warm-starting was implemented for the auxiliary variables $\boldsymbol{t_l}$ using the predicted control trajectories from the previous time step. The simulation results are compared with the fully centralized approach in Fig.4. The cost function J, outlet temperature T, and the inlet temperature T_i are shown in the left hand side subplots (solid red curves). The corresponding absolute errors compared to the fully centralized approach is shown in the right hand side subplots (solid red curves). From the plots, it can be clearly seen that the primal decomposition approach of solving the multistage scenario MPC problem results in the same solution as the centralized solution, thus indicating proper formulation. The total number of master and subproblem iterations taken at each time step are shown in the bottom right subplot and the step-length α value obtained from the proposed backtracking algorithm is shown in the bottom left subplot.

As mentioned earlier, one of the main motivations to use primal decomposition is that it enables closed-loop implementation even if the master problem and subproblems have not fully converged. In order to test this, the iterations between the master problem and the subproblems were capped at 5 iterations to prematurely terminate the iterations. The simulation results are shown in Fig.4 using black dashed curves. It can be seen that by prematurely terminating the iterations, the non-anticipativity constraints remain feasible, however, the

closed-loop solution is sub-optimal. However, by warm starting the auxiliary variables, the proposed primal decomposition approach eventually converges to the optimal solution provided by the centralized approach even when the master and subproblem iterations are prematurely terminated. This is clearly seen in the error subplots, where the absolute error compared to the centralized approach (shown in black dashed lines) diminishes quickly over time.

Effect of the step-length size. The step-length α backtracked using the proposed backtracking algorithm is shown in the bottom left subplot. When the disturbance in the input feed stream changes at time $t = 300$s, the optimal solution drives the process to the constraint on reactor temperature. At time $t = 410$s, the step-length α is backtracked to a small value when operating close to the constraint. Keeping the step length constant at $\alpha = 2000$, resulted in infeasibility of the reactor temperature constraint. This is because for the operating conditions after about 400s of simulation, the upper bound on the step length $\overline{\alpha}$ for which the master problem remains feasible as shown in (16) is much lower than the initally used step length value $\alpha = 2000$. By using the proposed backtracking algorithm, the step length was backtracked to find the upper bound $\overline{\alpha}$ as shown in the bottom-left suboplot in Fig. 4, such that it ensures the nonlinear process constraints also remain feasible throughout the iterations while updating the auxiliary variables in the master problem.

5.2. Simulation case 2

In this subsection, we simulate the system with a robust horizon of $N_r = 2$, leading to $S = M^{N_r} = 9$ scenarios. The process was simulated for a total simulation time of 600s. The concentration of component B in the feed stream changes at time $t = 300$s from $0.15 mol/l$ to $0 mol/l$ just as in simulation case 1.

The process was then simulated using the primal decomposition based scenario decomposition, where the step length α_l was initialized with 500 for all l. At each iteration, the step lengths were suitably adjusted using the proposed back-tracking algorithm in Section 4. The simulation results and the corresponding absolute errors for this simulation case is shown in Fig.5.

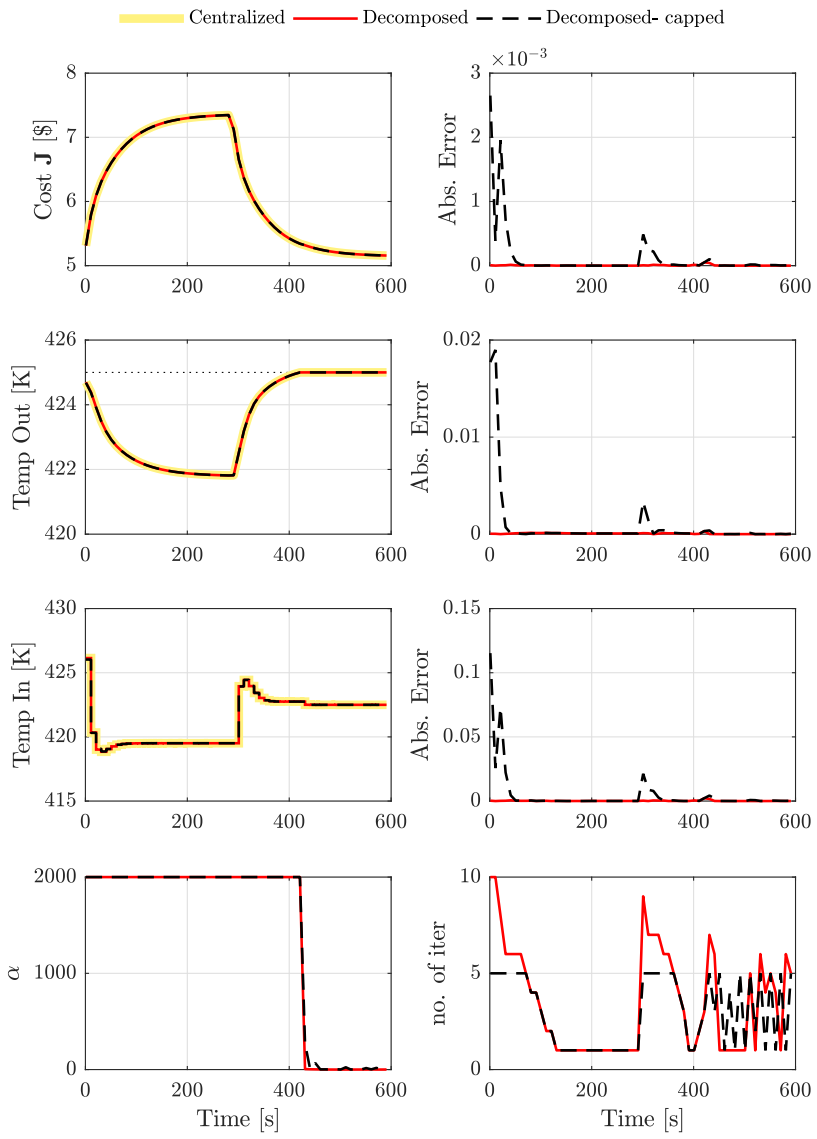

Figure 4: Simulation results with $N_r = 1$ showing the optimal solution provided by the centralized approach (thick yellow lines), primal decomposition approach (solid red lines) and the primal decompostion approach with the maximum number of iterations capped at 5 (black dashed lines).

Table 2: CPU times (in sec) for simulation 1 and 2

	$N_r = 1$			$N_r = 2$		
	max	avg	min	max	avg	min
Centralized	0.219	0.168	0.154	0.549	0.429	0.377
Decomposed	0.064	0.051	0.037	0.077	0.050	0.036

The proposed method was also simulated with the total number of iterations capped at 15 iterations. The simulation results are shown in Fig.5 using black dashed curves. It can be seen that the proposed primal decomposition approach eventually converges to the optimal solution provided by the centralized approach even when the master and subproblem iterations are prematurely terminated.

As can be seen from the simulation results from Fig. 4 and Fig. 5, the solution obtained by the proposed primal decomposition approach is almost identical to the one provided by solving the multistage problem in a centralized fashion. The Primal decomposition method was also shown to enable closed-loop implementation when the iterations between the master and scenario subproblems are prematurely terminated. The computation times for the multistage problem solved as a fully centralized problem and using the primal decomposition are also shown in Table. 2, which shows that, by using the proposed primal decomposition approach for multistage MPC, the same solution can be obtained at lesser computation times.The proposed primal decomposition based multistage scenario decomposition approach was also demonstrated using an oil and gas production optimization case study in our recent work [6].

6. Discussions

6.1. Scenario decomposition using Primal decomposition

One of the key challenges today in real-time implementation of optimizing controllers such as model predictive control is the computation time. The late

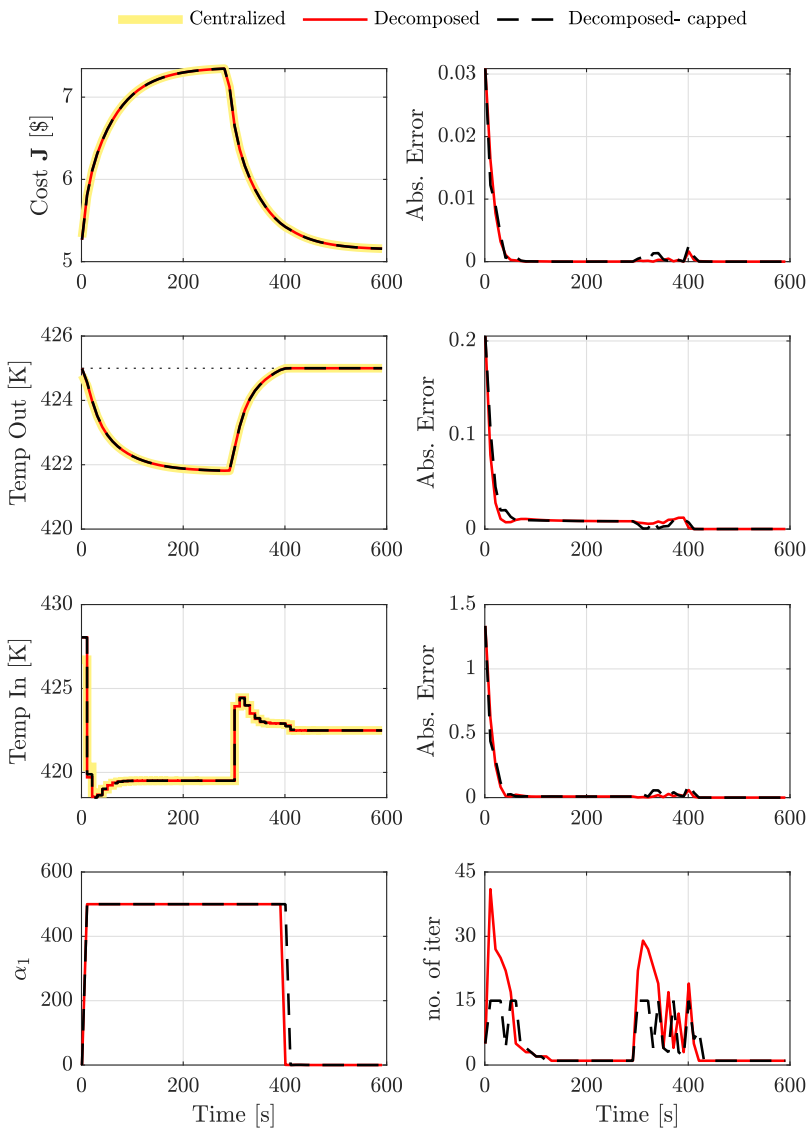

Figure 5: Simulation results with $N_r = 2$ showing the optimal solution provided by the centralized approach (thick yellow lines), primal decomposition approach (solid red lines) and the primal decompostion approach with the maximum number of iterations capped at 15 (black dashed lines).

arrival of a solution in many cases may simply not be acceptable. A solution to the optimization problem must ideally be available within the sampling time. As the author in [17] puts it, *the correctness of a computation is a function of time*.

Decomposition methods for scenario decomposition has its roots in multistage stochastic optimization problems studied in the operations research community, see [18, 19, 20, 21] to name a few. Problems studied in the operations research field do not focus on real-time closed-loop implementation in the same fashion as MPC in the process control community. The nature of the problems studied in operations research community often call for offline optimization problems as opposed to MPC applications in the process industries. For example, in many process control applications, a new control input must be computed at every sampling instant, which may be in the time scale of seconds to minutes. As a result, the dual decomposition methods developed for multistage stochastic optimization problems may not be directly applicable for the multistage MPC problem. Keeping this in mind, it may seem that *approximate solution now is better than accurate solution tomorrow*. Many optimization solvers are also based on this strategy, see for example [22]. We have used a similar motivation in this paper to use primal decomposition approach to efficiently solve the multistage scenario decomposition problem. Initializing the optimization routine for the very first time can be done by for example initializing the auxiliary variables using any feasible sub-optimal operating point, such as the last known operating value of the control inputs.

Since primal decomposition provides a primal feasible solution with monotonically decreasing objective value with each iteration, premature termination of the iterations only results in suboptimal operation and does not violate the non-anticipativity constraints. By warm-starting the subsequent time steps, the solution eventually converges to the true optimal solution. This was also seen in the error plots in Fig.4 and Fig.5. We believe, the primal decomposition approach addresses the practical implementation issues of distributed multistage scenario MPC problem.

6.2. Feasibility-ensuring backtracking algorithm

In this paper, we also proposed a feasibility ensuring backtracking algorithm to suitably choose the step length size in the master problem update such that the nonlinear constraints in the subproblems remain feasible throughout the iterations. In algorithm 1, we check the forward simulation for all the discrete realizations of the uncertainty used in the scenario tree to backtrack the step length used in the master problem. However, if the worst-case realization of the uncertainty is known a-priori, we then need to check the feasibility of the local constraints only for the worst case scenario w.r.t to the nonlinear constraints instead of all the scenarios. This is justified because if the local constraints within a subproblem are feasible for the worst case scenario, then it must also be feasible for all other scenarios.

With the proposed primal decomposition approach, it is important to note that the initial guess of the auxiliary variables $\boldsymbol{t}$ must be a feasible guess with respect to the nonlinear constraints. As described in 4, with the assumption of a feasible initial guess, the backtracking algorithm used here always ensures the feasibility of the nonlinear constraints in the scenario subproblems according to (16). One simple aopproach to get an initial feasible guess is by warm starting the auxiliary variable using the predicted control trajectory. The predicted control trajectory ranges from $\mathbf{u}_1$ to $\mathbf{u}_N$ for each scenario. The first control input $\mathbf{u}_1$ (which is the same for all the scenarios due to the non-anticipativity constraints) is implemented on the plant. For the next MPC iteration, the auxiliary variable can be initalized using the predicted control trajectory starting from the second time step $\mathbf{u}_2$ to $\mathbf{u}_{N_r+1}$ in the prediction horizon corresponsing to the worst case scenario. By initializing the auxiliary variables using the predicted control input for the worst case scenario, the initial guess will be feasible for all other scenarios as well. Additional back-off on predicted control input from the worst case scenario may also be used when initializing the auxiliary variables at each time step to ensure a feasible initial guess.

6.3. Scenario Decomposition using NLP sensitivities

The performance of the scenario-based MPC scheme can be further improved by using NLP sensitivities. Since the different scenario subproblems $\mathbf{\Phi}(\boldsymbol{t}_l, \boldsymbol{p}_j)$ differ only in the parameter $\boldsymbol{p}_j$, the scenario decomposition problem can be recast in the framework parametric optimization problem. By doing so, we then need to solve only one subproblem as a full NLP problem. The subsequent subproblems can be solved by exploiting the parametric nature using a predictor-corrector quadratic problem (QP) that approximates the NLP [23], [24].

The predictor-corrector QP has been shown to provide good approximations of the NLP solution for small variations in the parameters. If the different discrete realizations of the uncertain parameters are not in the small neighbourhood of each other, then a path-following predictor-corrector QP can be applied. In the path-following approach, a series of QP problems are solved sequentially similar to an Euler integration scheme for ordinary differential equations. This is explained in more detail and applied to a scenario decomposition problem in [25].

7. Conclusion

We propose a primal decomposition approach (9) to solve scenario-based MPC problems as an alternative to dual decomposition approaches. Primal decomposition enables real-time closed-loop implementation even in the case where the iterations between the master problem and subproblems are terminated prematurely. We also present in Section 4 a novel feasibility-ensuring backtracking algorithm to suitably choose the size of the step length in the master problem update. A CSTR case study demonstrates the effectiveness of the proposed method.

References

[1] P. J. Campo, M. Morari, Robust model predictive control, in: American Control Conference, 1987, IEEE, 1987, pp. 1021–1026.

[2] P. Scokaert, D. Mayne, Min-max feedback model predictive control for constrained linear systems, IEEE Transactions on Automatic control 43 (8) (1998) 1136–1142.

[3] S. Lucia, T. Finkler, S. Engell, Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty, Journal of Process Control 23 (9) (2013) 1306–1319.

[4] S. Lucia, S. Subramanian, S. Engell, Non-conservative robust nonlinear model predictive control via scenario decomposition, in: Control Applications (CCA), 2013 IEEE International Conference on, IEEE, 2013, pp. 586–591.

[5] R. Martí, S. Lucia, D. Sarabia, R. Paulen, S. Engell, C. de Prada, Improving scenario decomposition algorithms for robust nonlinear model predictive control, Computers & Chemical Engineering 79 (2015) 30–45.

[6] D. Krishnamoorthy, B. Foss, S. Skogestad, A distributed algorithm for scenario-based model predictive control using primal decomposition., IFAC papers-online (ADCHEM) 51 (2018) 351–356.

[7] D. P. Bertsekas, Nonlinear programming, Athena Scientific, 1999.

[8] E. Klintberg, J. Dahl, J. Fredriksson, S. Gros, An improved dual newton strategy for scenario-tree mpc, in: Decision and Control (CDC), 2016 IEEE 55th Conference on, IEEE, 2016, pp. 3675–3681.

[9] S. Boyd, L. Xiao, A. Mutapcic, J. Mattingley, Notes on decomposition methods, Notes for EE364B, Stanford University (2007) 1–36.

[10] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends® in Machine Learning 3 (1) (2011) 1–122.

[11] R. T. Rockafellar, R. J.-B. Wets, Scenarios and policy aggregation in optimization under uncertainty, Mathematics of operations research 16 (1) (1991) 119–147.

[12] L. S. Lasdon, Optimization theory for large scale systems, Series of Operation Research.

[13] V. Alstad, Studies on selection of controlled variables, PhD thesis (2005).

[14] J. Jäschke, S. Skogestad, NCO tracking and self-optimizing control in the context of real-time optimization, Journal of Process Control 21 (10) (2011) 1407–1416.

[15] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, M. Diehl, CasADi – A software framework for nonlinear optimization and optimal control, Ph.D. thesis (In Press, 2018).

[16] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker, C. S. Woodward, SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers, ACM Transactions on Mathematical Software (TOMS) 31 (3) (2005) 363–396.

[17] E. C. Kerrigan, G. A. Constantinides, A. Suardi, A. Picciau, B. Khusainov, Computer architectures to close the loop in real-time optimization, in: Decision and Control (CDC), 2015 IEEE 54th Annual Conference on, IEEE, 2015, pp. 4597–4611.

[18] J. R. Birge, Decomposition and partitioning methods for multistage stochastic linear programs, Operations research 33 (5) (1985) 989–1007.

[19] A. Ruszczyński, An augmented lagrangian decomposition method for block diagonal linear programming problems, Operations Research Letters 8 (5) (1989) 287–294.

[20] A. Ruszczyński, Parallel decomposition of multistage stochastic programming problems, Mathematical programming 58 (1-3) (1993) 201–228.

[21] T. Helgason, S. W. Wallace, Approximate scenario solutions in the progressive hedging algorithm, Annals of Operations Research 31 (1) (1991) 425–444.

[22] A. Shahzad, E. C. Kerrigan, G. A. Constantinides, A stable and efficient method for solving a convex quadratic program with application to optimal control, SIAM Journal on Optimization 22 (4) (2012) 1369–1393.

[23] V. Kungurtsev, J. Jaschke, A predictor-corrector path-following algorithm for dual-degenerate parametric optimization problems, SIAM Journal on Optimization 27 (1) (2017) 538–564.

[24] E. Suwartadi, V. Kungurtsev, J. Jäschke, Sensitivity-based economic NMPC with a path-following approach, Processes 5 (1) (2017) 8.

[25] D. Krishnamoorthy, E. Suwartadi, B. Foss, S. Skogestad, J. Jäschke, Improving scenario decomposition for multistage MPC using a sensitivity-based path-following algorithm, IEEE Control Systems Letters 4 (2) (2018) 581–586.