Learning to Recycle Benders Cuts for Mixed Integer Model Predictive Control

Ilias Mitrai,a Prodromos Daoutidisa\*

a Department of Chemical Engineering and Materials Science, University of Minnesota, Minneapolis, 55455 MN, USA

daout001@umn.edu

Abstract

Mixed integer MPC problems arise frequently in cases where the operation of a system depends on continuous and discrete decisions, leading to mixed integer optimization problems. However, the online solution of such problems is computationally challenging. In this work, we develop a machine learning approach to determine which cuts should be used as a warm start (recycled) for Generalized Benders Decomposition for the solution of mixed integer MPC problems. Computational results on a case study regarding the operation of chemical processes show that the proposed approach leads to a significant reduction in solution time (up to reduction).

**Keywords**: Model Predictive Control, Benders Decomposition, Mathematical Optimization, Machine Learning

* 1. Introduction

Model Predictive Control (MPC) is a widely used optimization-based control strategy for handling disturbances that affect the operation of process systems (Rawlings et al., (2017), Daoutidis et al., (2018)). The efficiency of an MPC strategy depends on the efficient online solution of an optimization problem. Despite significant advances in optimization theory and algorithms, the monolithic solution of mixed integer optimization (MIP) problems can be slow for online applications. Such problems arise in various applications such as unit commitment and online scheduling (Risbeck et al., (2020), McAllister and Rawlings (2022)). The standard approach to reduce the computational time is either to approximate the discrete problem with a continuous one (Masti et al., 2020) or use decomposition-based algorithms, such as Generalized Benders Decomposition (GBD) (Mitrai and Daoutidis (2022b)). Application of GBD to mixed integer MPC (MIP-MPC) decomposes the optimization problem into a master problem, which considers the integer (and potentially some continuous) variables, and a subproblem that considers the dynamic behavior of the system and whose solution depends on the values of the variables of the master problem (called complicating variables). These problems are solved repeatedly and coordinated via Benders cuts, which inform the master problem about the dynamic behavior of the system. Despite the reduction in solution time, the off-the-shelf implementation of GBD can be slow for online applications.

In recent work, we have shown that adding an initial set of cuts to the master problem (warm start) can reduce the solution time since the cuts contain information about the subproblem (Mitrai and Daoutidis, 2022b). However, determining which cuts to add is not apparent since the possible number of cuts can be very large. In this work, we propose a machine learning-based cut-recycling strategy for determining which cuts to recycle and ultimately accelerate GBD. Given a MIP-MPC problem, first, a set of high-quality integer feasible solutions is computed using a machine learning branch and check algorithm (Mitrai and Daoutidis, 2023b). The cuts associated with these integer feasible solutions are added (recycled) to the master problem, and GBD is implemented to obtain the solution of the MIP-MPC problem. We apply the proposed approach to a case study on the operation of an isothermal CSTR affected by changes in product demand and the inlet conditions. The results show that the proposed approach leads to a 40% reduction in solution time compared to the standard application of GBD.

* 1. Learning to recycle Benders cuts
     1. Generalized Benders Decomposition

We assume that the following optimization problem must be solved

|  |  |
| --- | --- |
|  | (1) |

where are the decision variables and are the constraints related to the behaviour of the system, such as discretized differential equations. Based on the number of variables and constraints, the online solution of this problem can be computationally expensive. GBD is based on the observation that if variables and are fixed, then the resulting problem is continuous and is equal to

|  |  |
| --- | --- |
|  | (2) |

where is the value function of the subproblem and we define as is the Lagrangean multipliers of the equality constraint . The original problem can be written as

|  |  |
| --- | --- |
|  | (3) |

The above problem cannot be solved directly since the value function is not known explicitly. In GBD, the value function is approximated via Benders cuts (Geoffrion, 1972)

|  |  |
| --- | --- |
|  | (4) |

where the index denotes the number of cuts used (the set contains all the cuts added). Given this approximation, the master problem can be written as

|  |  |
| --- | --- |
|  | (5) |

The algorithm alternates between the solution of the master problem, which provides a lower bound, and the subproblem, which provides an upper bound.

* + 1. Learning to recycle Benders cuts

In GBD, the cuts are added iteratively as dictated by the solution of the master problem. Adding an initial set of cuts, which we will refer to as warm start cuts, can reduce the solution time since the master problem has a better approximation of the value function in the first iteration. However, determining which cuts to add is nontrivial since the possible number of cuts can be large, and the addition of warm start cuts increases the complexity of the master problem. Therefore, an oracle is needed to determine the cuts that provide the maximum amount of information while minimizing the increase in the complexity of the master problem. We use a recently proposed machine learning based GBD algorithm as an oracle to compute high quality integer feasible solutions (Mitrai and Daoutidis, 2023b) which determine the cuts that are recycled.

* + - 1. Machine Learning branch and check GBD

The standard application of GBD can be slow for online applications since the master problem and subproblem are solved repeatedly. Recently, a machine learning-based branch and check GBD algorithm has been proposed to obtain high-quality integer feasible solutions. In this approach, the master problem is solved using branch and bound, and once an integer feasible solution is found at a node approximate Benders cuts are added to all the open nodes in the tree and the branch and bound procedure continues until the bounds converge. The approximate cuts, , are obtained using surrogate models to approximate the value function of the subproblem and the multipliers . The solution that is obtained is feasible, given that the subproblem is always feasible and the original problem is feasible since the cuts approximate the value function of the subproblem.

* + - 1. Cut Recycling strategy

The branch and check GBD algorithm provides a high-quality integer feasible solution and a set of integer feasible solutions that are explored during branch and check. We use these feasible solutions to determine the cuts that should be added (recycled) to the master problem in the first iteration. We define set which contains all the cuts that can be potentially added. This set is obtained offline by solving the subproblem for different values of the complicating variables. We also define set which contains the integer feasible solutions found during branch and check. For each integer feasible solution , the value function and multipliers are obtained from the set and the associated cut is added to the master problem. Once these warm start cuts are added, GBD is implemented.

* 1. Mixed Integer MPC formulation

We consider the application of the proposed approach to the optimal operation of chemical processes that can manufacture products over a time horizon discretized into slots. It is assumed that the operation of the system is affected by disturbances in product demand and inlet conditions of the process, and a mixed integer MPC controller is used to determine the production sequence and dynamic behavior of the system. Under this setting, the system is either performing transitions between the products, is manufacturing a product, or is performing a transition from an intermediate state to the product manufactured in the first slot (see Mitrai and Daoutidis (2023a) for a detailed description of the problem).

* + 1. Optimization model

We define as the set of products and the set of slots. We define binary variable which is equal to one if product is manufactured in slot and zero otherwise, variable   which is equal to one if a transition occurs from product to in slot , and variable   which is equal to one if a transition occurs from an intermediate state to product in the first slot. The logic constraints regarding the production sequence are

|  |  |
| --- | --- |
|  |  |
|  | (6) |
|  |  |

The starting and ending time of slot are , the production time of product in slot is , the transition time in slot is , and are the transition time from product to in slot and from to product . The timing constraints are

|  |  |
| --- | --- |
|  |  |
|  | (7) |
| . |  |

The production rate of product is , the inventory of product in slot is (the initial inventory is ), the amount of product sold in slot is , the demand is and is satisfied in the end of the time horizon . The inventory constraints are

|  |  |
| --- | --- |
|  | (8) |

where for . The dynamic behaviour of the system is described by a set of differential equations   where and are the state and manipulated variables of the system. The differential equations are discretized using orthogonal collocation points and finite elements (. We consider simultaneously all the transitions between the products and define variables as the value of the state and manipulated variables for a transition from product to in slot and discretization point . Similarly, we define variables for a transition from the intermediate state to product . The equations that describe the dynamic behaviour of the system have the following general form

|  |  |
| --- | --- |
|  | (9) |
|  | (10) |

The objective of the optimization problem has three terms. The first term is the profit minus the operating cost, the second term is the transition cost between the products, and the last term the transition cost from the intermediate state. The terms are equal to

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

where is the price and operating cost of product in slot , is the inventory cost, is the fixed transition cost for product to , is a weight coefficient, is the collocation matrix, and is the time at discretization point for a transition from product to in slot and for the transition from to product . The goal of the optimization problem is to maximize the profit subject to constraints presented in Eq. 6-10.

* + 1. Decomposition-based solution approach

The above problem is a large-scale Mixed Integer Nonlinear Programming problem which cannot be solved efficiently monolithically. However, if the scheduling related variables and transition times are fixed, the resulting optimization problem is to maximize subject to constraints 9,10. This problem can be decomposed into independent subproblems which consider the transitions between the products for given transition time and the transition from for a given  . The subproblems are

|  |  |
| --- | --- |
|  |  |
|  |  |

and the original problem can be written as

|  |  |
| --- | --- |
|  |  |

Since the value functions are not known explicitly they are approximated via Benders cuts as follows where are the Lagrangean Multipliers of the equality constraints and respectively. Given the above problem decomposition, the Benders cuts approximate the transition costs and we apply the proposed cut recycling strategy to determine which cuts to recycle as a warm start for accelerating GBD.

Table 1 Average solution time (in seconds) for the standard application of the multicut GBD, hybrid multicut GBD, and the proposed cut recycling approach.

|  |  |  |  |
| --- | --- | --- | --- |
| Algorithm | Standard Implementation | Initialization strategy | |
| OptSol | IntFeas |
| Multicut GBD | 64 | 54 (16) | 38 (40) |
| Hybrid multicut GBD | 40 | 36 (10) | 30 (25) |

* 1. Case study and results

We will assume that the system is an isothermal CSTR where a third-order irreversible reaction occurs, and six products are manufactured by adjusting the inlet flow rate. The time horizon is 35 hours, and at a random time point in the first 15 hours of operation, the demand of the products and the inlet concentration of the reactor change simultaneously. We consider the effect of the proposed approach on accelerating two GBD algorithms (multicut and hybrid multicut GBD) from the literature (Mitrai and Daoutidis, 2022a) where no warm start cuts are added to the master problem. In the multicut algorithm, cuts are added only for given transitions, as dictated by the master problem, whereas in the hybrid multicut algorithm, the cuts for a given transition are added for all transitions . Furthermore, we compare the proposed approach with the case where only the optimal solution returned by the ML-based branch and check GBD algorithm is used to determine which cuts to recycle (OptSol). We generate 20 random feasible disturbances, and the solution time for the different approaches is presented in Table 1. From the results we observe that the proposed cut recycling approach leads to a significant reduction in solution time compared to standard and accelerated GBD. Specifically, the solution time of multicut GBD is 64 seconds and the solution time with the proposed approach is 38 seconds, a reduction. Similar results are obtained for the accelerated GBD algorithm, where the solution time is reduced from 40 to 30 seconds (25 ­ reduction). Also, we observe that using all the cuts related to the integer feasible solutions leads to lower solution time, on average, compared to using the cuts from the optimal solution obtained from the ML-based Branch and Check GBD algorithm. This result shows that for the case of GBD, adding information related to high-quality integer feasible solutions is more beneficial than adding information related only to an approximation of the optimal solution, which is the common practice for monolithic solution methods (Bengio et al., 2021).

* 1. Conclusions

Mixed integer MPC applications arise in a wide range of applications, however, the efficient solution of the underlying optimization problem is computationally challenging. In this work we proposed a machine learning based cut recycling strategy to improve the computational performance of GBD. Application of the proposed approach to a case study on the operation of chemical processes, shows that the proposed approach leads to a significant reduction in solution time by exploiting information about multiple integer feasible solutions.

* 1. Acknowledgments

Financial support from NSF-CBET (award number 2313289) is gratefully acknowledged.

References

A. M. Geoffrion, 1972, Generalized benders decomposition. *Journal of Optimization Theory and Applications*, *10*, pp.237-260.

D. Masti, T. Pippia, A. Bemporad, and D. De Schutter, 2020, Learning approximate semi-explicit hybrid MPC with an application to microgrids. *IFAC-PapersOnLine*, *53*(2), pp.5207-5212.

I. Mitrai and P. Daoutidis, 2022a, A multicut generalized benders decomposition approach for the integration of process operations and dynamic optimization for continuous systems. *Computers & Chemical Engineering*, *164*, p.107859.

I. Mitrai and P. Daoutidis, 2022b, An adaptive multi-cut decomposition based algorithm for integrated closed loop scheduling and control. In *Computer Aided Chemical Engineering* (Vol. 49, pp. 475-480). Elsevier.

I. Mitrai, and P. Daoutidis, 2023a, Taking the human out of decomposition-based optimization via artificial intelligence: Part II. Learning to initialize. *arXiv preprint arXiv:2310.07082*.

I. Mitrai and P. Daoutidis, 2023b, Computationally efficient solution of mixed integer model predictive control problems via machine learning aided Benders Decomposition. *arXiv preprint arXiv:2309.16508*.

J.B. Rawlings, D.Q. Mayne, and M. Diehl, 2017, *Model predictive control: theory, computation, and design* (Vol. 2). Madison, WI: Nob Hill Publishing.

M.J. Risbeck, C.T. Maravelias, J.B. Rawlings, and R.D. Turney, 2020, Mixed-integer optimization methods for online scheduling in large-scale HVAC systems. *Optimization Letters*, *14*, pp.889-924.

P. Daoutidis, J.H. Lee, I. Harjunkoski, S. Skogestad, M. Baldea, and C. Georgakis, 2018, Integrating operations and control: A perspective and roadmap for future research. *Computers & Chemical Engineering*, *115*, pp.179-184.

R.D. McAllister, and J.B. Rawlings, 2022, Advances in mixed-integer model predictive control. In *2022 American Control Conference (ACC)* (pp. 364-369). IEEE.

Y. Bengio, A. Lodi, A. and A. Prouvost, 2021, Machine learning for combinatorial optimization: a methodological tour d’horizon. *European Journal of Operational Research*, *290*(2), pp.405-421.