



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS

Pedro Henrique Valderrama Bento da Silva

Hierarchical Decompositions for MPC of Linear Systems with Resource and Activation Constraints

Florianópolis

2020

Pedro Henrique Valderrama Bento da Silva

Hierarchical Decompositions for MPC of Linear Systems with Resource and Activation Constraints

Dissertação submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do título de Mestre em Engenharia de Automação e Sistemas

Advisor: Prof. Eduardo Camponogara, Dr.

Co-advisor(s): Prof. Laio Oriel Seman, Dr.,
Dr. Helton Fernando Scherer

Florianópolis

2020

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

da Silva, Pedro Henrique Valderrama Bento
Hierarchical Decompositions for MPC of Linear Systems
with Resource and Activation Constraints / Pedro Henrique
Valderrama Bento da Silva ; orientador, Eduardo
Camponogara, coorientador, Laio Oriel Seman, coorientador,
Helton Fernando Scherer, 2020.
96 p.

Dissertação (mestrado) - Universidade Federal de Santa
Catarina, Centro Tecnológico, Programa de Pós-Graduação em
Engenharia de Automação e Sistemas, Florianópolis, 2020.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2. Decomposições
hierárquicas, Decomposição de Benders, Outer Approximation.
3. Bilevel optimization. 4. Controle preditivo baseado em
modelo. 5. Restrição de recursos, Restrição de ativação. I.
Camponogara, Eduardo. II. Seman, Laio Oriel. III. Scherer,
Helton Fernando IV. Universidade Federal de Santa
Catarina. Programa de Pós-Graduação em Engenharia de
Automação e Sistemas. V. Título.

Pedro Henrique Valderrama Bento da Silva

Hierarchical Decompositions for MPC of Linear Systems with Resource and Activation Constraints

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

Profa. Luciana Salete Buriol, Dra.
Universidade Federal do Rio Grande do Sul

Prof. Daniel Martins Lima, Dr.
Universidade Federal de Santa Catarina

Prof. Felipe Gomes de Oliveira Cabral, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre em Engenharia de Automação e Sistemas.



Documento assinado digitalmente
Werner Kraus Junior
Data: 06/08/2020 10:40:33-0300
CPF: 531.085.239-53

Prof. Werner Kraus Junior, Dr.
Coordenador do Programa



Documento assinado digitalmente
Eduardo Camponogara
Data: 05/08/2020 21:42:48-0300
CPF: 513.988.180-20

Prof. Eduardo Camponogara, Dr.
Orientador

Florianópolis, 30 de Julho de 2020.

*Aos meus pais, minha irmã, namorada e a toda minha família que,
com muito carinho e apoio, não mediram esforços para que eu alcançasse
esse marco em minha vida.*

AGRADECIMENTOS

A Deus por me proporcionar perseverança durante toda a minha vida.

Aos meus pais Sergio e Elizeth pelo apoio e incentivo que serviram de alicerce para as minhas realizações. A minha irmã, Ana Clara pela amizade e atenção dedicadas quando sempre precisei, e pequenas brigas aqui e ali. À minha querida namorada Fernanda pelo seu amor incondicional, paciência nessa distância que ficamos, ajuda em todos os momentos, e por compreender minha dedicação ao projeto de pesquisa.

Um agradecimento em especial ao meu professor orientador Eduardo Camponogara pelo conhecimento me pasado, pela amizade nesses últimos dois anos e valiosas contribuições dadas durante todo o processo do trabalho. Ao meu coorientador Laio Oriel Seman, por tantos dias e noites de conversa que levaram a evolução do trabalho, você teve um papel fundamental na elaboração deste trabalho.

A todos os meus amigos do mestrado, conhecidos do Mestrandos Anônimos que compartilharam dos inúmeros desafios que enfrentamos, risadas e horas passadas no LTIC – UFSC, vocês foram muito importantes para amenizar as dificuldades deste trabalho.

Aos meus amigos Mané-Maringaenses, que saíram todos da cidade de Maringá e criaram esta família em Florianópolis, obrigado por todas as vezes que me ajudaram no processo do meu trabalho, em especial, ao meu eterno mestre Feres Azevedo Salem por todas as conversas e conselhos, tanto para introdução ao mestrado como na parceria de morar em Florianópolis.

Deixo também meu agradecimento ao *Group of Optimization Systems* – GOS, pelo convívio e conhecimentos que trocamos ao longo desses dois anos.

Também quero agradecer à Universidade Federal de Santa Catarina – UFSC, ao Departamento de Automação e Sistemas – DAS, ao Programa de Pós graduação em Automação e Sistemas – PPGEAS, e o seu corpo docente que demonstrou estar comprometido com a qualidade e excelência do ensino.

*"There's no such thing as a painless lesson.
They just don't exist. Sacrifices are necessary.
You can't gain anything without losing something first.
Although, if you can endure that pain and walk away
from it, you'll find you now have a heart strong enough
to overcome any obstacle – a heart made fullmetal"
(Edward Elric, Fullmetal Alchemist).*

RESUMO

A interconexão de subsistemas dinâmicos que compartilham recursos limitados pode ser encontrada em muitas aplicações, e o controle desse sistemas de subsistemas é um objeto extensivo de estudos. O controle preditivo baseado em modelo (MPC) tornou-se uma técnica popular de controle, possivelmente por sua capacidade de lidar com dinâmica complexa e restrições do sistema. Os algoritmos MPC encontrados na literatura são principalmente centralizados, com um único controlador coletando sinais e realizando os cálculos de sinais de saída. No entanto, a estrutura distribuída desses subsistemas interconectados não é necessariamente explorada pelo MPC padrão. Para esse fim, este trabalho propõe decomposições hierárquicas com a finalidade de dividir os cálculos entre um problema mestre (componente centralizado) e um conjunto de subproblemas desacoplados (componentes distribuídos), que traz flexibilidade organizacional e capacidade de computação distribuída. Três métodos gerais são considerados para controle e otimização hierárquica: otimização em dois níveis, decomposição de Benders e Aproximação Externa. Os resultados são relatados a partir de uma análise numérica das decomposições e de uma aplicação simulada ao gerenciamento de energia, na qual uma fonte limitada de energia é distribuída entre as baterias de veículos elétricos. Em seguida, para validar o uso de restrições de ativação, com o uso da decomposição de Benders e aproximação externa, novas análises e simulações numéricas foram realizadas no carregamento de baterias de veículos elétricos.

Palavras-chaves: Decomposições hierárquicas, Decomposição de Benders, Aproximação externa, Programação de dois níveis, Controle preditivo, Restrição de recursos.

RESUMO EXPANDIDO

Introdução

A interconexão de subsistemas dinâmicos que compartilham recursos limitados pode ser encontrada em muitas aplicações, e o controle desse sistemas de subsistemas é um objeto extensivo de estudos. Tais sistemas podem apresentar um alto grau de acoplamento entre as unidades interconectadas. O controle preditivo baseado em modelo (MPC) tornou-se uma técnica popular de controle, possivelmente por sua capacidade de lidar com dinâmica complexa e restrições do sistema. Os algoritmos MPC encontrados na literatura são principalmente centralizados, com um único controlador coletando sinais e realizando os cálculos de sinais de saída. No entanto, essa interconexão de subsistemas, com uma estrutura distribuída, não é necessariamente explorada pelo MPC padrão. Este trabalho propõe uma estrutura de decomposição hierárquica para dividir os cálculos entre um problema *master* (componente centralizado) e um conjunto de subproblemas desacoplados (componentes distribuídos), o que gera flexibilidade organizacional e computação distribuída.

Objetivos

O principal objetivo desta dissertação é aplicar e avaliar metodologias de decomposição para otimização e controle de sistemas dinâmicos com restrições de recursos e ativação. Além disto, os objetivos específicos são os seguintes: (a) Propor abordagens de otimização para problemas de Controle Preditivo baseado em Modelo (MPC), considerando restrições de recursos limitadas; (b) Projetar, implementar e testar estratégias de decomposição para o controle ótimo, considerando metodologias como otimização em dois níveis, decomposição de Benders e aproximação externa; (c) Realizar experimentos numéricos e análises computacionais das metodologias de otimização de dois níveis e Benders para o Controle Preditivo de uma classe de problemas de sistemas dinâmicos com restrição de recursos, com dinâmica linear de tempo discreto; (d) Estender o Controle Preditivo de sistemas dinâmicos com restrições de recursos para considerar a ativação/desativação de unidades de controle, o que implica o uso de variáveis binárias e a aplicação da decomposição de Benders e Aproximação Externa; (e) Apresentar resultados de experimentos numéricos e análises computacionais da aplicação de metodologias de Benders e de Aproximação Externa para o problema de MPC com recursos limitados com variáveis de ativação/desativação; (f) Ilustrar a aplicação dos métodos de decomposição ao problema de recarga de baterias de veículos elétricos, com e sem variáveis de ativação/desativação.

Metodologia

Este trabalho propõe uma estrutura de decomposição hierárquica para dividir os cálculos entre um problema *master* (componente centralizado) e um conjunto de subproblemas desacoplados (componentes distribuídos), o que gera flexibilidade organizacional e computação distribuída. Três métodos gerais são considerados para controle e otimização hierárquica: otimização em dois níveis, decomposição de Benders e o método de aproximação externa. A otimização de dois níveis é uma metodologia para reformular a estrutura centralizada em uma estrutura distribuída com coordenação simples que abre um grande potencial para computação em paralelo, visando arquiteturas *multi-core* ou ainda de computação distribuída. A decomposição de Benders e o método de aproximação externa tem como objetivo principal resolver problemas com variáveis inteiras que, quando temporariamente fixadas, produzem um problema significativamente mais fácil de se lidar, pois explora a estrutura do problema e descentraliza a carga computacional, podendo tratar o problema em uma estrutura distribuída com potencial de cálculos em paralelo.

Resultados e Discussão

O benefício das decomposições hierárquicas se dá de forma principalmente organizacional, pois permitem que o sistema de controle seja reconfigurado localmente e expandido com coordenação reduzida. Os sinais comunicados entre o *master* e os subsistemas são relativamente simples, consistindo em alocações de recursos (do *master* para os subsistemas), cortes e derivadas/sensibilidades (dos subsistemas para o *master*). Nessa estrutura, não é necessário que o *master* tenha informações detalhadas sobre os subproblemas. Os resultados mostram, a partir de uma análise numérica das decomposições e de uma aplicação simulada de carregamento de baterias de veículos elétricos, que a otimização de dois níveis apresentou melhores resultados computacionais se comparado com a decomposição de Benders. E então com o objetivo de validar o uso de restrições de ativação, com o uso da decomposição de Benders e aproximação externa, foram realizadas novas análises numéricas e também simulações em carregamento de baterias de veículos elétricos, onde a aplicação da técnica de aproximação externa apresentou um desempenho superior à decomposição de Benders.

Considerações Finais

A interconexão de subsistemas dinâmicos que compartilham recursos limitados, são sistemas que apresentam um alto grau de acoplamento entre as unidades conectadas. Com o objetivo criar uma estrutura hierárquica de controle e desacoplar essas unidades conectadas, são propostas técnicas de decomposição para solução dos problemas de forma hierárquica. Os capítulos apresentam a modelagem do problema de MPC com restrição de recursos, e então uma reformulação utilizando estratégias de decomposição hierárquica. Além disto, a dissertação propõe um conjunto de instâncias do problema para análise computacional, além da aplicação em um problema exemplo de carregamento de baterias de veículos elétricos. Os trabalhos futuros vão no sentido de: (a) introduzir múltiplos cortes, onde pode-se considerar um corte para cada subproblema na decomposição de Benders; (b) introdução de *Lazy-Constraints* (Restrições sob Demanda); (c) fazer uso de técnicas de regularização, como por exemplo, Regularização de Nível, a fim de se obter melhor desempenho computacional.

Palavras-chaves: Decomposições hierárquicas, Decomposição de Benders, Aproximação externa, Programação de dois níveis, Controle preditivo, Restrição de recursos.

ABSTRACT

The interconnection of dynamic subsystems that share limited resources are found in many applications, and the control of such systems of subsystems has driven significant attention from scientists and engineers. For the operation of such systems, model predictive control (MPC) has become a popular technique, arguably for its ability to deal with complex dynamics and system constraints. The MPC algorithms found in the literature are mostly centralized, with a single controller receiving the signals and performing the computations of output signals. However, the distributed structure of such interconnected subsystems is not necessarily explored by standard MPC. To this end, this work proposes hierarchical decomposition to split the computations between a master problem (centralized component) and a set of decoupled subproblems (distributed components), which brings about organizational flexibility and distributed computation. Three general methods are considered for hierarchical control and optimization, namely bilevel optimization, Benders decomposition, and outer approximation. Results are reported from a numerical analysis of the decompositions and a simulated application to energy management, in which a limited source of energy is distributed among batteries of electric vehicles. Then, in order to validate the use of activation constraints, with the use of Benders decomposition and outer approximation, new numerical analyzes and simulations were carried out on battery charging of electric vehicles.

Key-words: Hierarchical decompositions, Benders decomposition, Outer Approximation, Bilevel programming, Model predictive control, Resource Constraints.

LIST OF FIGURES

Figure 1 – HVAC system in a building	27
Figure 2 – Water distribution system	28
Figure 3 – Distributed control scheme	28
Figure 4 – Centralized control scheme	28
Figure 5 – Hierarchical control scheme	29
Figure 6 – Example of feasibility region in a optimization problem	32
Figure 7 – Illustration of maximum and minimum, local and global optimum	33
Figure 8 – Example of convex and non-convex sets	35
Figure 9 – Example of a Convex and a Concave functions	35
Figure 10 – The duality gap between primal and dual objective values	37
Figure 11 – Sensitivity analysis in a convex problem with one inequality constraint.	39
Figure 12 – General sketch of a bilevel problem.	40
Figure 13 – Block diagram for solving MINLP	45
Figure 14 – Flow chart of the outer approximation algorithm	48
Figure 15 – Trajectory of the solution bounds.	49
Figure 16 – Geometrical interpretation of linearizations in master problem.	49
Figure 17 – Benders Algorithm flow chart	51
Figure 18 – Representation of MPC predictions with reference tracking in a time instant.	54
Figure 19 – MPC block diagram	54
Figure 20 – Percentage distance of the solution obtained by Bilevel optimization with respect to the global optimum, for the case $M = 40$ and $T = 6$	70
Figure 21 – Trajectory of the lower and upper bounds produced by Benders decomposition, for the case $M = 20$ and $T = 4$	71
Figure 22 – Comparison between decomposition techniques according to speed-up (su_{ratio}) and parallel efficiency metrics (pe).	72
Figure 23 – Application of bilevel and Benders decomposition to the battery charging problem.	74
Figure 24 – Trajectory of the bounds in Benders decomposition solution, for problem $M = 8$ and $T = 6$	84
Figure 25 – Trajectory of the bounds in Outer Approximation solution, for problem $M = 8$ and $T = 6$	84
Figure 26 – Application of Benders decomposition and Outer Approximation to the battery charging problem, with activation constraints.	86

LIST OF TABLES

Table 1 – Number of variables of synthetic problems, for each decomposition strategy.	67
Table 2 – Problem data related to Table 3	68
Table 3 – Computational Analysis of Benders and Bilevel Decompositions	69
Table 4 – Number of variables of synthetic problems with binary variables, for analysis of decomposition strategies.	82
Table 5 – Problem data related to Table 6	82
Table 6 – Computational analysis of the Outer Approximation and Benders decomposition.	83

LIST OF ABBREVIATIONS AND ACRONYMS

BLP	Bilevel Programming
BD	Benders Decomposition
DMC	Dynamic Matrix Control
GPC	Generalized Predictive Control
IP	Integer Programming
KKT	Karush-Kuhn-Tucker
LP	Linear Programming
LR	Level Regularization
MP	Master Problem
MILP	Mixed-Integer Linear Programming
MINLP	Mixed-Integer Non-Linear Programming
MPC	Model Predictive Control
NLP	Non-Linear Programming
OA	Outer Approximation
SP	Subproblem
SQP	Sequential Quadratic Problem

LIST OF SYMBOLS

α_B	Benders Decomposition master problem variable
α_{OA}	Outer Approximation master problem variable
$\mathcal{C}$	Convex set
$\mathcal{D}$	Domain of dual problem
$\mathcal{F}$	Set of Feasibility Cuts
$\mathcal{M}$	Subsystems set
$\mathcal{N}$	Prediction Horizon set
$\mathcal{N}_u$	Control Horizon set
N_1	Prediction Horizon start value
N_2	Prediction Horizon final value
N_u	Control Horizon value
$\mathcal{O}$	Set of Optimality Cuts
$\mathcal{R}$	Set of resources
$\mathbf{R}$	Set of real numbers
$\mathbf{Z}$	Set of integer numbers
λ	Lagrangean multiplier associated with inequalities constraints
ν	Lagrangean multiplier associated with equality constraints
m	Subsystem value
x_m	System state
y_m	Predicted output
w_m	Desired output trajectory
u_m	Predicted control input
Δu_m	Predicted control variation
Q_m	Positive definite matrix that penalize the errors on trajectory tracking

W_m	Positive definite matrix that penalize the errors on control variation
$y_m^{\min}$	Lower bound on the predicted output
$y_m^{\max}$	Upper bound on the predicted output
$u_m^{\min}$	Lower bound on the predicted input
$u_m^{\max}$	Upper bound on the predicted input
$\Delta u_m^{\min}$	Lower bound on the predicted control variation
$\Delta u_m^{\max}$	Upper bound on the predicted control variation
$s_r^{\max}$	Amount of resource r available
$s_{r,m}$	Rate of consumption by subsystem m .
δ_m	Activation/deactivation variable on control of subsystem m

CONTENTS

1	Introduction	27
1.1	Motivation (Problem statement)	27
1.2	Objectives	29
1.2.1	General objective	29
1.2.2	Specific objectives	30
1.3	Organization of the dissertation	30
2	Fundamentals in Optimization	31
2.1	Overview of Optimization	31
2.1.1	Continuous and Discrete Optimization	32
2.1.2	Unconstrained and Constrained Optimization	33
2.1.3	Global and Local Optimization	33
2.2	Convex Optimization	34
2.2.1	Duality	36
2.2.2	Optimality Conditions and Sensitivity	37
2.3	Bilevel Programming	39
2.4	Decomposition Strategies	43
2.4.1	Outer Approximation	45
2.4.2	Benders Decomposition	48
2.5	Summary	52
3	Fundamentals in Model Predictive Control	53
3.1	Basic Ideas	53
3.2	State Space MPC	55
3.2.1	Optimization Problem in MPC	58
3.2.2	Obtaining the Control Law	59
3.3	Summary	60
4	Decompositions for MPC of Resource-Constrained Dynamic Systems	61
4.1	Bilevel optimization	62
4.2	Benders Decomposition	64
4.3	Numerical experiments	66
4.4	Battery Charging Application	72
4.4.1	Application to an Example Instance	73
4.5	Summary	74
5	Decompositions for MPC of Dynamic Systems with Activation Constraints	77

5.1	Benders Decomposition and Outer Approximation	77
5.1.1	Optimality and Feasibility Subproblems	78
5.1.2	Master Problem of the Benders Decomposition	79
5.1.3	Outer Approximation Master Problem	80
5.2	Numerical Experiments	81
5.3	Batteries Charging Application with activation constraints	85
5.3.1	Application to a Sample Instance	86
5.4	Summary	87
6	Final Remarks	89
6.1	Conclusion	89
6.2	Future work proposal	90
6.3	Contributions	91
	Bibliography	93

1 INTRODUCTION

1.1 MOTIVATION (PROBLEM STATEMENT)

Several systems found in the industry and society emerge from the interconnection of dynamic subsystems that share limited resources (SCHERER et al., 2013; SCHERER et al., 2015). Representative systems include stations for recharging batteries of electric vehicles, energy building management, and distribution of cooling fluid in buildings, among others.

In Figure 1, it is shown a building that has a Heating, Ventilation, and Air Conditioning (HVAC), as an example of resource constrained dynamic system. This building has several rooms, all with its energetic demand, depending on the number of people, the wall materials, so the cooling fluid is a shared resource available for these rooms (subsystems). Another example appears in Figure 2, which shows a city with industries, houses, and commercial centers that have their water demands, thus the water available in the water reservoir represents the limited resource in a specific time.

Figure 1 – HVAC system in a building

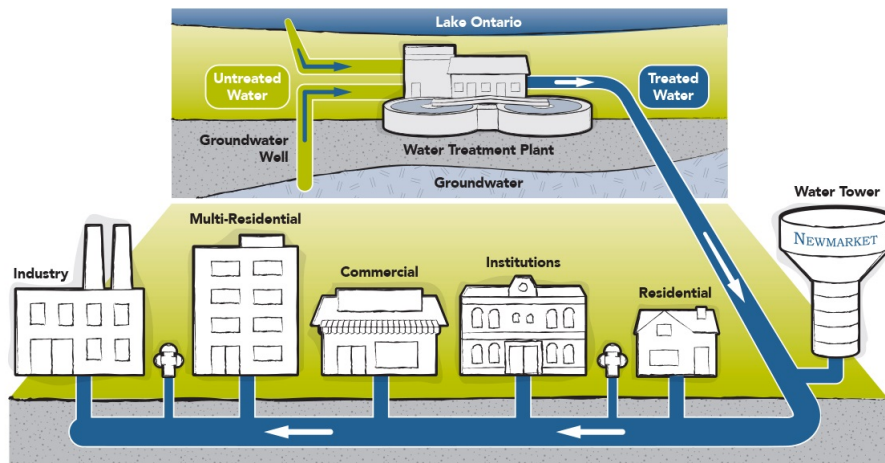


Source: Author(2020)

There are two classes of strategies that can be approached for controlling such systems, the centralized control (CAMACHO; BORDONS, 2007), and the decentralized control (SANDELL N. et al., 1978). Although decentralized control is fast and scalable, the lack of coordination between distributed units can lead to poor performance or render the operation infeasible. Figure 3 gives a glance on the structure of this type of controller, where there is a controller for each system, and these controllers communicate with each other.

On the other hand, centralized control is capable of optimal performance, but the computational cost can become high, and the monolithic approach is less flexible. Figure 4 present a

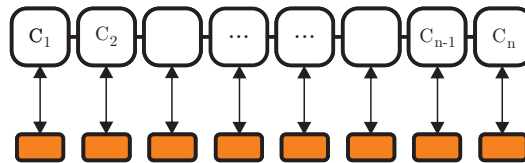
Figure 2 – Water distribution system



Source: Author(2020)

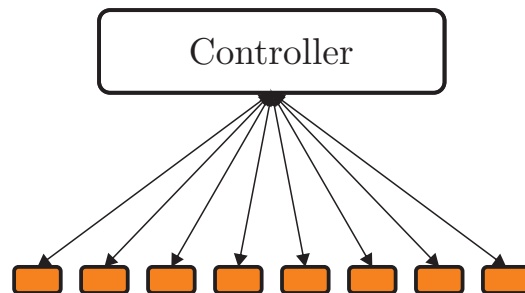
scheme for the centralized control, where there is just one controller that communicates with all systems.

Figure 3 – Distributed control scheme



Source: Author(2020)

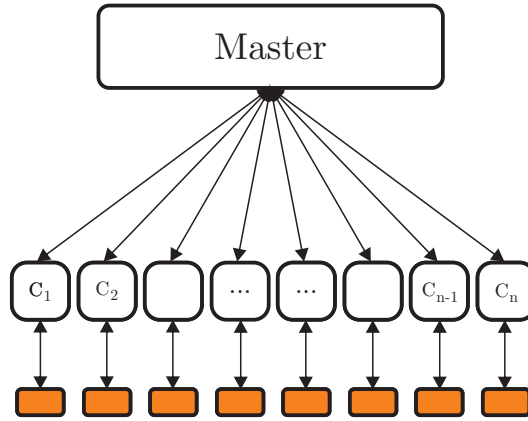
Figure 4 – Centralized control scheme



Source: Author(2020)

As an alternative, some approaches combine the characteristics of decentralized and centralized controls, with emphasis on decomposition strategies that enable hierarchical control, which can be seen in Figure 5, where there is a master problem that coordinates other controllers, so these controllers can be considered decoupled, having independent dynamics. Bilevel decomposition (COLSON; MARCOTTE; SAVARD, 2007), Lagrangean decomposition (GUIGNARD; KIM, 1987), Benders decomposition (BENDERS, 1962), and Outer Approximation (DURAN; GROSSMANN, 1986) are examples of such approaches.

Figure 5 – Hierarchical control scheme



Source: Author(2020)

According to Colson, Marcotte and Savard (2007), bilevel decomposition is a methodology to reformulate the centralized structure into a coordinated distributed structure that enables parallel computations, aiming multi-core or distributed computing architectures.

Benders decomposition and Outer Approximation, such as the Bilevel decomposition, allow us to harness the structure of the problem, in the same manner, using parallel computing. Under specific and general conditions on the problems, Benders decomposition and Outer Approximation ensure convergence to the optimal solution (GROSSMANN, 2002). These two decomposition strategies can be used to model activation variables, i.e., binary variables, to enable or disable the input of control signals into the system if necessary.

Despite this potential of parallel computation, hierarchical and distributed algorithms are often slower than their centralized counterparts. Thus, the main benefit of a hierarchical decomposition is not computational, but rather organizational as it facilitates the expansion and reconfiguration of the control system, a feature that stems from the simple coordination scheme and reduced information communication (CAMPONOGARA et al., 2020).

In this context, this dissertation aims to present Bilevel optimization, Benders decomposition, and Outer Approximation approaches for MPC of a resource-constrained dynamic system, considering a problem that has a coupling constraint. The cited decompositions take advantage of the hierarchical structure, which enables the decoupling of the control task into a set of subproblems that can be solved in a distributed or parallel manner.

1.2 OBJECTIVES

1.2.1 General objective

The present dissertation seeks to develop and apply decomposition methodologies for optimization and control of dynamic systems with resource and activation constraints.

1.2.2 Specific objectives

- Propose optimization approaches for Model Predictive Control problems, considering limited resource constraints.
- Design, implement, and test decomposition strategies for optimal control, considering methodologies such as Bilevel optimization, Benders decomposition, and Outer Approximation.
- Perform numerical experiments and computational analysis of the Bilevel and Benders methodologies for the Model Predictive Control of a class of resource-constrained dynamic systems, with linear discrete-time dynamics.
- Extend the Model Predictive Control of resource-constrained dynamic systems to consider activation/deactivation of control units, which entails the use of binary variables and the application of Benders decomposition and Outer Approximation.
- Present results from numerical experiments and computational analysis of the application of Benders and Outer Approximation methodologies for the resource-constrained MPC problem with activation/deactivation variables.
- Illustrate the application of the decomposition methods to the problem of recharging batteries of electric vehicles, with and without activation/deactivation variables.

1.3 ORGANIZATION OF THE DISSERTATION

This dissertation is divided in six chapters. Chapter 2 presents an overview of mathematical optimization, its formulations, and details on specific topics such as convex optimization and large scale optimization strategies. Chapter 3 presents a review on Model Predictive Control, focused on the formulations used in this work and giving details on the implementation of algorithms. In Chapter 4, a formulation for MPC of resource-constrained dynamic systems is presented, and two hierarchical decomposition strategies are applied and analyzed in representative systems. Chapter 5 consists of an extension of the previous chapter, which introduces the capacity of activation/deactivation of control units into the model, such as the description of the algorithms that are capable of solving these problems, and numerical analysis of such strategies. Finally, Chapter 6 presents conclusions of the dissertation and directions for future work.

2 FUNDAMENTALS IN OPTIMIZATION

This chapter introduces concepts of mathematical optimization and discusses their importance in making decisions and analyzing physical systems. Then the chapter gives an introduction to convex optimization and presents some key topics for the dissertation, including reformulation strategies and problem decompositions that enable a more straightforward problem solution.

2.1 OVERVIEW OF OPTIMIZATION

In precise terms, an optimization problem is a problem of finding values for decision variables that minimize (or maximize) an objective function subject to constraints that define a feasible space (NOCEDAL; WRIGHT, 2006). A general mathematical optimization problem, or just optimization problem, has the form:

$$\text{minimize } f(x) \quad (2.1a)$$

$$\text{subject to } g_i(x) \leq 0, \quad i = 1, \dots, m, \quad (2.1b)$$

$$h_i(x) = 0, \quad i = 1, \dots, l, \quad (2.1c)$$

$$x \in \mathcal{X}. \quad (2.1d)$$

Where, the vector $x = (x_1, \dots, x_n)$ collects the optimization variables of the problem, the function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is the objective function, and the constraints $h_i(x) = 0$ for $i = 1, \dots, l$ and $g_i(x) \leq 0$ for $i = 1, \dots, m$ are the equality and inequality constraint functions, respectively. The set $\mathcal{X}$ typically includes lower and upper bounds on the variables which, even when implied by the other constraints, can play a useful role in some algorithms (BAZARAA; SHERALI; SHETTY, 2006).

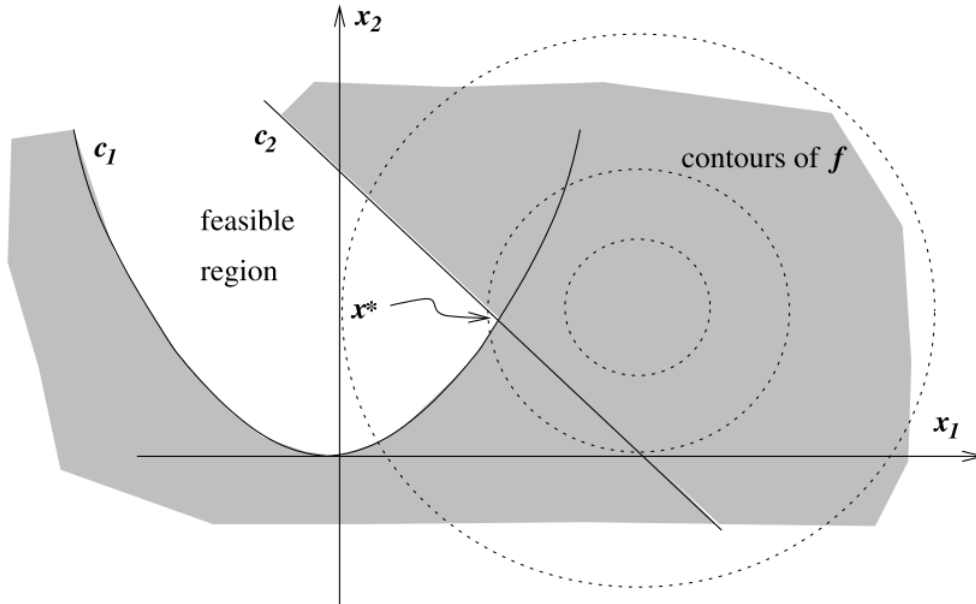
Vanderbei (2014) discusses that it often seems that real-world problems are most naturally formulated as minimization problems, a pessimist way to model problems, but of course, it is possible to maximize in optimization. For example, to minimize cost or maximize profit in manufacturing, converting one to another is trivial: minimize $f \equiv \text{maximize } -f$, enabling a more optimistic point of view.

A vector x^* is called optimum or optimal solution for the problem (2.1), if it has the smallest objective value among all vectors that are inside the feasible region, which is the set of points satisfying all the constraints (the region within the constraint boundaries).

Figure 6 illustrates the feasible region of an optimization problem, which aims to minimize a function f subject to two inequalities constraints, c_1 and c_2 . The shaded part of the figure

is the infeasible region of the problem, so the optimal solution x^* lies inside the set created by the two constraints.

Figure 6 – Example of feasibility region in a optimization problem



Source: Nocedal and Wright (2006)

In optimization, it is essential to know what kind of problem is there to solve, i.e., to classify the optimization model, since the algorithms for solving these problems are designed from specific characteristics of each problem class. Optimization problems can be classified into certain types, such as:

- Continuous and Discrete Optimization.
- Unconstrained and Constrained Optimization.
- Global and Local Optimization.

2.1.1 Continuous and Discrete Optimization

Certain optimization problems only make sense if they have variables defined within the set of integers. The mathematical formulation of such problems includes integrality constraints, which have the form $x_i \in \mathbf{Z}$, where $\mathbf{Z}$ is the set of integers, or binary constraints, which have the form $x_i \in \{0, 1\}$.

Problems of this type are called integer programming (IP) problems. Classical problems of IP are the 0-1 Knapsack Problem and the Traveling Salesman Problem. In a variation of these problems, where some variables are not restricted to be just integer or binary variables, but which contain some variables that can take on any real value, i.e., have the form $x_i \in \mathbf{R}$, they are called mixed-integer programming problems (MIP) (NOCEDAL; WRIGHT, 2006).

Integer programming problems are the type of discrete optimization problems, and those that do not have an integer set of variables are the type of continuous optimization problems.

2.1.2 Unconstrained and Constrained Optimization

In optimization, problems can be classified according to the nature of the objective function and their constraints (linear, nonlinear, and convex). Also, an essential classification regarding the latter is whether there are restrictions on the variables or not.

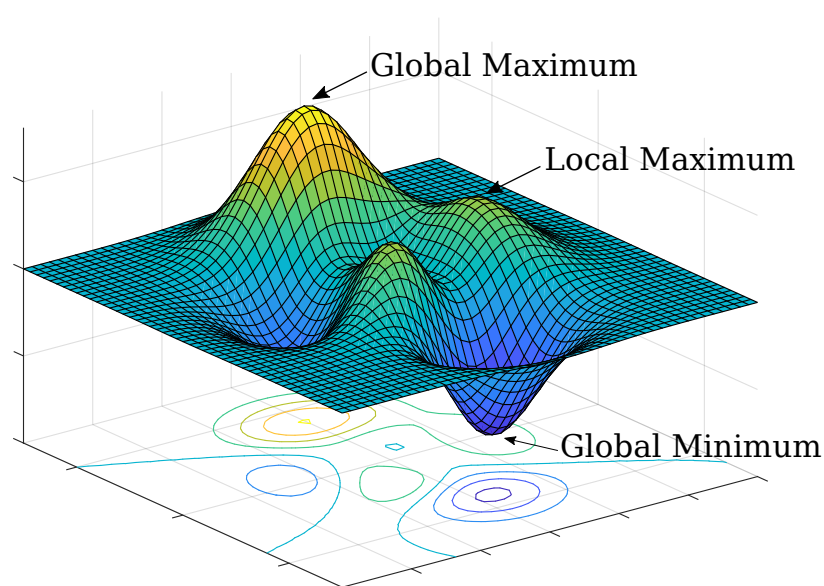
Unconstrained optimization problems are of the general form (2.1) where the constraints h and g do not exist. Many algorithms solve a constrained problem by converting it into a sequence of unconstrained problems via Lagrangean multipliers, or via penalty and barrier functions, e.g., interior-point methods (BAZARAA; SHERALI; SHETTY, 2006).

On the other hand, constrained optimization problems emerge from models in which constraints play an essential part. These constraints can be simple bounds on the variables, such as $0 \leq x_1 \leq 10$, to systems of equalities and inequalities that model complex relationships among the variables.

2.1.3 Global and Local Optimization

Real-world problems, generally, are described with nonlinear functions. These functions may exhibit various behaviors, many points of maximum and minimum, as shown in Figure 7. Global optimization is used for problems with a small number of variables, where computing time is not critical, and the value of finding the truly global solution is very high.

Figure 7 – Illustration of maximum and minimum, local and global optimum



Local optimization methods can solve large-scale problems in short time, and are widely applicable since they only require, for some algorithms, differentiability of the objective and constraint functions. These methods are commonly used in applications where there is value in finding a good point, if not the very best (BOYD; VANDENBERGHE, 2012).

Nocedal and Wright (2006) affirm that many algorithms for nonlinear optimization problems seek only a local solution, which is a point at which the objective function is smaller than all other feasible nearby points, but not the lowest among all the feasible set, which is the global solution of the problem. For some problems the local minimum is the global minimum, e.g., convex optimization problems.

An arbitrary local minimum is easy to obtain when using classical local optimization methods. However, finding the global minimum of a function can be tricky. The use of numerical solution strategies, for global optimization, often leads to serious problems and makes the computational cost higher.

2.2 CONVEX OPTIMIZATION

Convex optimization is a particular class of mathematical optimization, which includes well-known problems, such as least-squares and linear programming (LP) problems. Briefly, LP are optimization problems that have linear equations describing the objective function and its constraints. Convex optimization has been studied extensively for more than a century.

However, recent studies have shown that this type of problem is related to many practical fields, such as control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modelling, statistics, and finance (BOYD; VANDENBERGHE, 2012).

Another relevant point is that algorithms that were first developed to solve LP problems in the 1980s, such as the interior-point algorithm, described in Karmarkar (1984), can also be used to solve convex optimization problems.

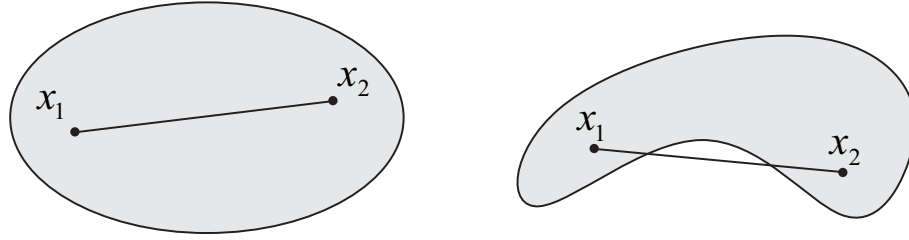
In optimization, there are functions and sets that are defined as convex in certain conditions. A set $\mathcal{C} \subseteq \mathbf{R}^n$ is a convex set if a straight line segment connecting any two points in $\mathcal{C}$ lies entirely inside $\mathcal{C}$. In other words, given any two points $x_1, x_2 \in \mathcal{C}$, and $x_1 \neq x_2$, then

$$(1 - \theta)x_1 + \theta x_2 \in \mathcal{C}, \forall \theta \in [0, 1]. \quad (2.2)$$

Figure 8 provides some examples of convex and non-convex sets. The first set is a convex set, which contains a line entirely within the shaded region, and the second set is non-convex because there is a region where the points of the line are not within the shaded area.

Some basic properties are presented to accurately describe convex problems. A function f is a convex function, if its domain $\mathcal{C}$ is a convex set and if for two points x_1 and x_2 in $\mathcal{C}$, the

Figure 8 – Example of convex and non-convex sets



Source: Adapted from Boyd and Vandenberghe (2012)

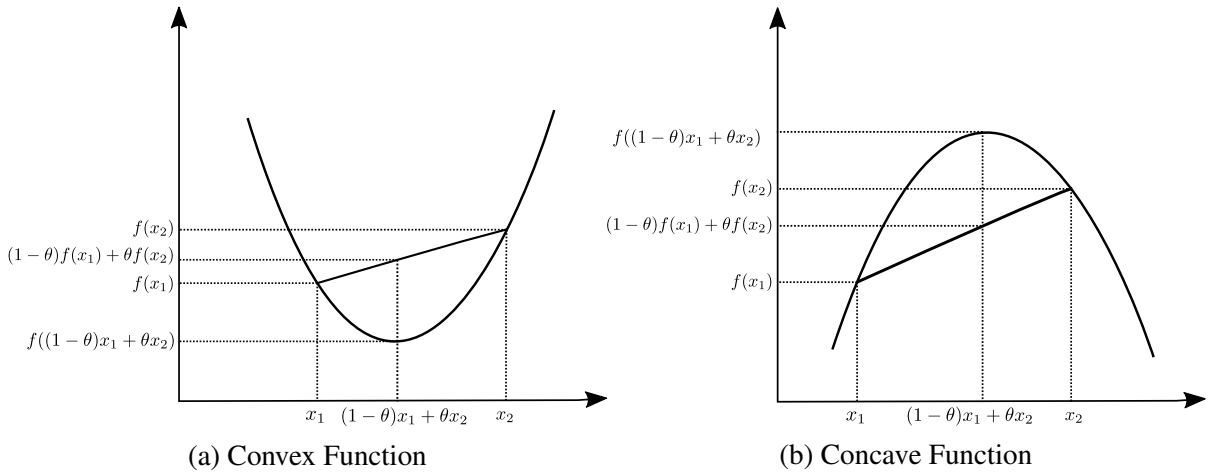
following is satisfied,

$$f((1 - \theta)x_1 + \theta x_2) \leq (1 - \theta)f(x_1) + \theta f(x_2), \quad \forall \theta \in [0, 1]. \quad (2.3)$$

This inequality establishes that the line defined by two points, x_1 and x_2 , for a convex function is above the function applied at these two points, and for a concave function the line is below the function. Figure 9 shows that behavior.

It is possible to say if f is convex, $-f$ is concave. If the objective function in the optimization problem (2.1) and the feasible region are both convex, then any local solution of the problem is, in fact, a global solution.

Figure 9 – Example of a Convex and a Concave functions



Source: Adapted from Floudas (1995)

A convex optimization problem in the standard form is one of the form,

$$\text{minimize} \quad f(x) \quad (2.4a)$$

$$\text{subject to} \quad g_i(x) \leq 0, \quad i = 1, \dots, m, \quad (2.4b)$$

$$h_i(x) = 0, \quad i = 1, \dots, l, \quad (2.4c)$$

where f and $g_1, \dots, g_m$ are convex functions and $h_1, \dots, h_l$ are affine functions. In contrast with (2.1), the problem (2.4) has three additional requirements:

- the objective function is convex,
- the equality constraint functions must be affine, and
- the inequality constraint functions must be convex.

2.2.1 Duality

In optimization, duality is the term used to refer to the dual of a problem. Every optimization problem, defined as Eq. (2.1), which is known as primal problem, has another manner of approaching the problem, known as dual problem, or Lagrange dual problem.

The dual problem is often used to find certificates of optimality, that is, a way to establish that a solution to a given problem is an optimal solution or just a good solution, which can be improved. In special cases, the dual provides an easier way to solve the problem.

It is essential to define the Lagrangean function and Lagrange multipliers. The main idea is to take the equality and inequality constraints of problem (2.1) into account by augmenting the objective function with a weighted sum of the constraint functions. Problem convexity is not assumed (BOYD; VANDENBERGHE, 2012).

Assuming that the domain $\mathcal{D} = \bigcap_{i=1}^m \text{dom}(g_i) \cap \bigcap_{i=1}^l \text{dom}(h_i)$ is nonempty, and denoting the primal optimal value of (2.1) as p^* , the Lagrangean $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^l \rightarrow \mathbf{R}$ associated with (2.1) is

$$L(x, \lambda, \nu) = f(x) + \sum_{i=1}^m \lambda_i g_i(x) + \sum_{i=1}^l \nu_i h_i(x). \quad (2.5)$$

The value λ_i is the Lagrange multiplier associated with the i th inequality constraint, and similarly, the value ν_i is referred to the Lagrange multiplier associated with the i th equality constraint.

Secondly, the dual function $g : \mathbf{R}^m \times \mathbf{R}^l \rightarrow \mathbf{R}$ is defined as the minimum value of the Lagrangean over variable x . For $\lambda \in \mathbf{R}^m$ and $\nu \in \mathbf{R}^l$, the Lagrangean dual function is given by

$$g(\lambda, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda, \nu). \quad (2.6)$$

For each pair (λ, ν) with $\lambda \geq 0$, the dual function brings out an important property: the value $g(\lambda, \nu)$ of the optimal solution to (2.6) is a lower bound to the optimal objective p^* .

Notice that the solution of (2.6), which induces a lower bound, depends on the values the parameters λ and ν . An improved lower bound for the primal problem is obtained by solving the following,

$$\text{maximize } g(\lambda, \nu) \quad (2.7a)$$

$$\text{subject to } \lambda \geq 0, \quad (2.7b)$$

which is called the Lagrange dual problem associated with problem (2.1). The values λ^* and ν^* are referred as the optimal Lagrange multipliers, which are the optimal solution for (2.7). The Lagrange dual problem is a concave optimization problem, since the objective is to maximize a concave function and the constraints are convex.

2.2.2 Optimality Conditions and Sensitivity

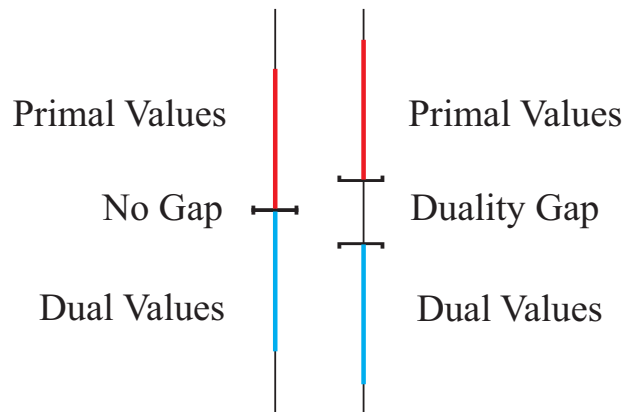
The dual problem brings about certain optimality conditions, or certificates, that must be satisfied by any solution point. Given any dual feasible pair of Lagrange multipliers (λ, ν) , a lower bound on the primal optimal is established, such that $p^* \geq g(\lambda, \nu)$.

Dual feasible points allow to bound how suboptimal a given feasible point is, without knowing the exact value of p^* . If x is primal feasible and (λ, ν) is dual feasible, then,

$$f(x) - p^* \leq f(x) - g(\lambda, \nu). \quad (2.8)$$

The gap between primal and dual objectives, $f(x) - g(\lambda, \nu)$, is called the duality gap associated with the primal and dual feasible points. In a convex problem, for example, where a solution is always a global solution for the problem, the duality gap is zero, i.e., $f(x^*) = g(\lambda^*, \nu^*)$ for optimal primal and dual points, respectively x^* and (λ^*, ν^*) . This equality relation is referred to as strong duality property. Figure 10 demonstrates the difference between strong and weak duality.

Figure 10 – The duality gap between primal and dual objective values



Source: Author (2020)

For any optimization problem with differentiable objective and constraint functions, for which strong duality holds, any pair of primal and dual optimal points must satisfy the Karush-Kuhn-Tucker (KKT) conditions, which are first-order conditions for optimality (BAZARAA; SHERALI; SHETTY, 2006).

These conditions are defined as follows, consider x^* and (λ^*, ν^*) to be any primal and dual optimal points with zero duality gap. Since x^* minimizes $L(x, \lambda^*, \nu^*)$ over x , the gradient

is zero at x^* , leading to,

$$g_i(x^*) \leq 0, \quad i = 1, \dots, m \quad (2.9a)$$

$$h_i(x^*) = 0, \quad i = 1, \dots, l \quad (2.9b)$$

$$\lambda_i^* \geq 0, \quad i = 1, \dots, m \quad (2.9c)$$

$$\lambda_i^* g_i(x^*) = 0, \quad i = 1, \dots, m \quad (2.9d)$$

$$\nabla f(x^*) + \sum_{i=1}^m \lambda_i^* \nabla g_i(x^*) + \sum_{i=1}^l \nu_i^* \nabla h_i(x^*) = 0. \quad (2.9e)$$

The KKT conditions play an important role in optimization in general. When the primal problem is convex, the KKT conditions are also sufficient for the points to be primal and dual optimal.

If a convex optimization problem, with differentiable objective and constraint functions, satisfies Slater's condition, then the KKT conditions provide necessary and sufficient optimality conditions (BOYD; VANDENBERGHE, 2012). So, it is possible to obtain sensitivity information about the solution, when strong duality holds. The optimal dual variables enable us to perform a sensitivity analysis of the optimal value concerning perturbations of the constraints.

Considering a problem in the following form,

$$\text{minimize} \quad f(x) \quad (2.10a)$$

$$\text{subject to} \quad g_i(x) \leq u_i, \quad i = 1, \dots, m, \quad (2.10b)$$

$$h_i(x) = v_i, \quad i = 1, \dots, l, \quad (2.10c)$$

which is the same problem of (2.1) when $u = 0$ and $v = 0$.

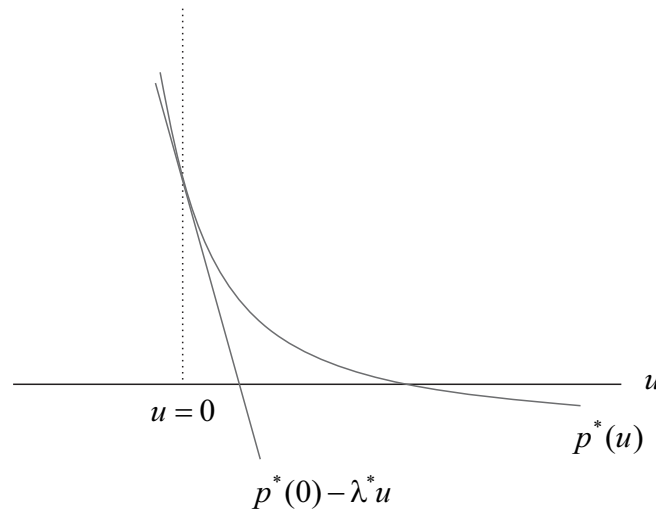
The sensitivity is associated with how a constraint can change the solution of a problem, when varying the parameters u and v of (2.10). Whether a constraint has some impact on the problem or not depends on the constraint being active or inactive.

There are two types of possible analyses, global and local analysis. When considering the local sensitivity analysis, an essential property is obtained related to the Lagrange multipliers associated with the constraints of the primal problem. Assuming $u = 0$ and $v = 0$, this property is expressed as:

$$\lambda_i^* = -\frac{\partial p^*(0,0)}{\partial u_i}, \quad \nu_i^* = -\frac{\partial p^*(0,0)}{\partial v_i}. \quad (2.11)$$

The optimal dual variables λ^*, ν^* associated with the constraints of the problem are related to the gradient of the optimal objective of the problem, considering that the problem is convex and strong duality holds.

Figure 11 – Sensitivity analysis in a convex problem with one inequality constraint.



Source: Adapted from Boyd and Vandenberghe (2012)

The Lagrange multiplier λ_i^* gives the rate at which the objective will decrease if the i th inequality is marginally enlarged, in the case of the constraint being binding. If the i th inequality is not binding, then $\lambda_i^* = 0$, e.g., as can be seen in Figure 11, the optimal value $p^*(u)$ of a convex problem with one constraint $f_1(x) \leq u$, as a function of u . For $u = 0$, we have the original unperturbed problem; for $u < 0$ the constraint is tightened, and for $u > 0$ the constraint is loosened. The affine function $l(u) = (p^*(0) - \lambda^*u)$ is a lower bound on p^* .

2.3 BILEVEL PROGRAMMING

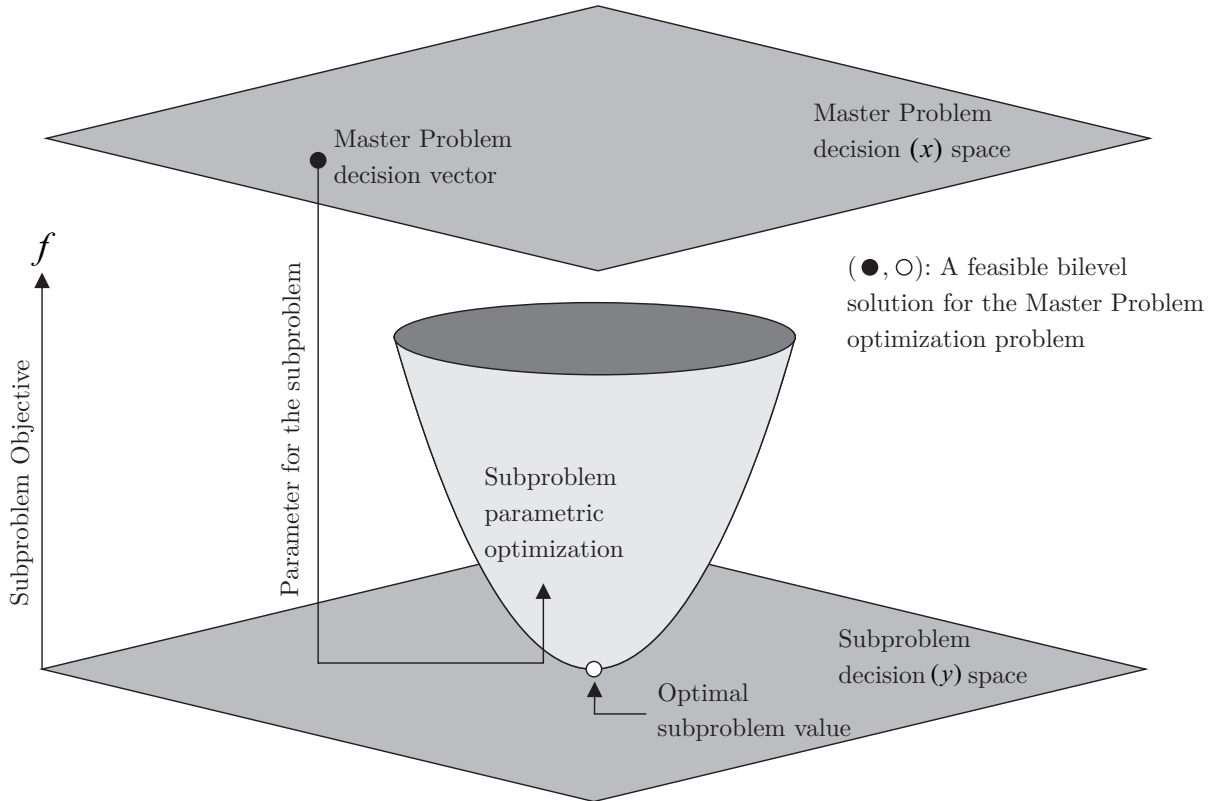
Bilevel programming problems (BLP) arise from a particular case of multilevel programming problems, which is an optimization problem that contains other optimization problems embedded in its constraints. Historically, bilevel programs were derived from problems found in military operations, as well as in production and marketing decision making (BRACKEN; MCGILL, 1974; BRACKEN; MCGILL, 1978). By that time, such problems were called *mathematical programs with optimization problems in the constraints*. Bilevel programming is also related to game theory, where these problems can be interpreted as a hierarchical game of two decision-makers (or players) which make their decisions according to a hierarchical order.

Figure 12 illustrates a general bilevel problem-solving structure involving coupled optimization and decision-making tasks at both levels. The figure shows that for any given upper-level decision vector, there is a corresponding (parametric) lower-level optimization problem to be solved, providing a rational (optimal) solution of the follower for the leader's decision.

Colson, Marcotte and Savard (2007) state that hierarchical relationships between two decision levels are encountered in fields as diverse as management (facility

location, environmental regulation, credit allocation, energy policy, hazardous materials), economic planning (social and agricultural policies, electric power pricing, oil production), engineering (optimal design, structures and shape), chemistry, environmental sciences, optimal control, etc. For one, it can be argued that most management decisions are of a bilevel nature, in the sense that they impact systems with some degree of autonomy and conflicting objectives. However few real-life studies have adopted this paradigm.

Figure 12 – General sketch of a bilevel problem.



Source: Adapted from Sinha, Malo and Deb (2018)

The general formulation of a bilevel programming problem is,

$$\min_{x \in \mathcal{X}, y} F(x, y) \quad (2.12a)$$

$$\text{s.t. } G(x, y) \leq 0, \quad (2.12b)$$

$$\min_y f(x, y), \quad (2.12c)$$

$$\text{s.t. } g(x, y) \leq 0, \quad (2.12d)$$

where the variables are divided into two classes, $x \in \mathbf{R}^{n_1}$ are the variables of the upper-level problem, or master problem, and $y \in \mathbf{R}^{n_2}$ are the variables of the lower-level problem, or subproblem. The functions $F : \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} \rightarrow \mathbf{R}$ and $f : \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} \rightarrow \mathbf{R}$ are the upper-level and lower-level objective functions respectively, while the vector-valued functions $G : \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} \rightarrow \mathbf{R}^{m_1}$ and $g : \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} \rightarrow \mathbf{R}^{m_2}$ are the upper-level and lower-level constraints respectively.

To provide more details about the solution of the BLP, a relaxed problem associated with the problem (2.12) is introduced as follows,

$$\min_{x \in \mathcal{X}, y} F(x, y) \quad (2.13a)$$

$$\text{s.t. } G(x, y) \leq 0, \quad (2.13b)$$

$$g(x, y) \leq 0, \quad (2.13c)$$

whose optimal value is a lower bound for the optimal value of problem (2.12). The relaxed feasible region (or constraint region) is,

$$\Omega = \{(x, y) \in \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} : x \in \mathcal{X}, G(x, y) \leq 0, g(x, y) \leq 0\}. \quad (2.14)$$

For a given (fixed) vector $\bar{x} \in \mathcal{X}$, the subproblem feasible set is defined by

$$\Omega(\bar{x}) = \{y \in \mathbf{R}^{n_2} : g(\bar{x}, y) \leq 0\}. \quad (2.15)$$

The subproblem solution set is defined as follows,

$$R(\bar{x}) = \{y \in \mathbf{R}^{n_2} : y \in \operatorname{argmin}\{f(\bar{x}, \hat{y}) : \hat{y} \in \Omega(\bar{x})\}\}. \quad (2.16)$$

For a given x , $R(x)$ is an implicitly defined multi-valued function of x that may be empty for some values of its argument. Finally, the set

$$\mathcal{IR} = \{(x, y) \in \mathbf{R}^{n_1} \times \mathbf{R}^{n_2} : x \in \mathcal{X}, G(x, y) \leq 0, y \in R(x)\}, \quad (2.17)$$

defines the feasible points of the BLP, which corresponds to the feasible set of the master problem, also known as the induced region (KALASHNIKOV et al., 2015). This set is usually nonconvex and it can even be disconnected or empty in the presence of upper-level constraints (COLSON; MARCOTTE; SAVARD, 2007).

Gümüř and Floudas (2001) discuss that linear BLP has the favorable property that the solution occurs at an extreme point of the feasible set. However, this condition does not hold for the nonlinear BLP. For being intrinsically complex, only the most straightforward classes of bilevel programming problems have been addressed, namely problems with linear or convex objective and constraints. In particular, the most studied instance of bilevel programming problems has been for a long time the linear BLP, in which all functions are linear. In general, non-convex and non-differentiable BLPs, even for simple cases, have been shown to be $\mathcal{NP}$ -hard (SINHA; MALO; DEB, 2018).

The conventional solution approach to the nonlinear BLP is to transform the original two-level problem into a single level one, by replacing the lower level optimization problem with the set of equations that define its KKT conditions. The problem (2.12) becomes,

$$\min_{x \in \mathcal{X}, y, \lambda} F(x, y) \quad (2.18a)$$

$$\text{s.t. } G(x, y) \leq 0, \quad (2.18b)$$

$$\lambda_i \geq 0, \quad i = 1, \dots, m_2 \quad (2.18c)$$

$$g(x, y) \leq 0, \quad (2.18d)$$

$$\lambda_i g_i(x, y) = 0, \quad i = 1, \dots, m_2 \quad (2.18e)$$

$$\nabla_y f(x, y) + \sum_{i=1}^{m_2} \lambda_i \nabla_y g_i(x, y) = 0. \quad (2.18f)$$

Even under suitable convexity assumptions on the functions F , G and the set $\mathcal{X}$, the above mathematical program is hard to solve due to the nonconvexities that appear in the complementarity and Lagrangean constraints. While the Lagrangean constraint is linear in certain important cases (linear or convex quadratic functions), the complementarity constraint (2.18e) is intrinsically combinatorial (COLSON; MARCOTTE; SAVARD, 2007).

The result is a special case of the mathematical program with equilibrium constraints (MPEC). MPECs are nonconvex optimization problems. Algorithms for solving MPECs compute local optimal solutions or stationary points (DEMPE et al., 2015). MPECs can be viewed as bilevel programs where the lower-level problem consists in a variational inequality. For a given function $\Psi : \mathbf{R}^n \rightarrow \mathbf{R}^n$ and a convex set $\mathcal{C} \subseteq \mathbf{R}^n$, the vector $x^* \in \mathcal{C}$ represents the solution of the variational inequality if it satisfies,

$$(x - x^*)^T \Psi(x^*) \geq 0, \quad \forall x \in \mathcal{C}. \quad (2.19)$$

Variational inequalities are mathematical programs that allow the modelling of many equilibrium phenomena encountered in engineering, physics, chemistry or economics, hence the origin of the name of MPEC.

The general formulation of an MPEC is defined by the following:

$$\min_{x,y} F(x, y) \quad (2.20a)$$

$$\text{s.t. } (x, y) \in \mathcal{Z}, \quad (2.20b)$$

$$y \in \mathcal{S}(x), \quad (2.20c)$$

where $\mathcal{Z} \subseteq \mathbf{R}^{n_1+n_2}$ is a nonempty closed set and $\mathcal{S}(x)$ is the solution set of the parametrized variational inequality. So the constraint (2.20c) can be substituted by,

$$y \in \mathcal{S}(x) \Leftrightarrow y \in \mathcal{C}(x) \text{ and } (v - y)^T \Psi(x, y) \geq 0, \quad \forall v \in \mathcal{C}(x), \quad (2.21)$$

defined over the closed convex set $\mathcal{C}(x) \subset \mathbf{R}^{n_2}$. As for bilevel problems, x and y represent the master problem and subproblems variables respectively.

Consider a particular case where the mapping $\Psi(x, \cdot)$ is the partial gradient map of a real-valued continuously differentiable function $f : \mathbf{R}^{n_1+n_2} \rightarrow \mathbf{R}$, that is,

$$\Psi(x, y) = \nabla_y f(x, y). \quad (2.22)$$

Substituting (2.22) in (2.21) and the resulting expression in the problem (2.20), the following is obtained:

$$\min_{x,y} F(x, y) \quad (2.23a)$$

$$\text{s.t. } (x, y) \in \mathcal{Z}, \quad (2.23b)$$

$$(v - y)^T \nabla_y f(x, y) \geq 0, \forall v \in \mathcal{C}(x) \quad (2.23c)$$

$$y \in \mathcal{C}(x). \quad (2.23d)$$

Thus, for any fixed x , the variational inequality (2.23c) defines the set of stationary conditions of the optimization problem:

$$\min_y f(x, y) \quad (2.24a)$$

$$\text{s.t. } y \in \mathcal{C}(x). \quad (2.24b)$$

Then, if $\mathcal{C}(x)$ takes the form

$$\mathcal{C}(x) = \{y : g(x, y) \leq 0\}, \quad (2.25)$$

the problem (2.24) is the subproblem defined in (2.12c)-(2.12d). This states that MPEC become bilevel programs when the latter involves a convex and differentiable subproblem. On the other hand, an MPEC can be formulated as a bilevel program if the lower-level variational inequality (2.23c) is reformulated by an optimization problem of the form (2.24)-(2.25).

2.4 DECOMPOSITION STRATEGIES

A wide diversity of real-world and industrial problems are described with nonlinear models. Consequently, they become nonlinear optimization problems, and commonly with this class of problems are those that involve integer or discrete variables such as in integer programming problems introduced in Section 2.1.1.

When discrete and continuous variables are mixed in a linear problem, the problem becomes mixed-integer linear programming (MILP), further assuming that constraints and objective are linear.

If the problem consists of a set of nonlinear functions, with discrete and continuous variables, the problem is said to be mixed-integer nonlinear programming or MINLP. The coupling of the integer, with the continuous domain, along with their associated nonlinearities, renders the class of MINLP problems very challenging from the theoretical, algorithmic, and computational point of view (FLOUDAS, 1995).

MINLP is fitted in many applications such as process synthesis in chemical engineering, design, scheduling, and planning of batch processes. Also, in other areas such as facility location

problems in a multi-attribute space, the optimal unit allocation in an electric power system, and the facility planning of an electric power generation.

A general form of MINLP is stated as,

$$\min_{x,y} \quad Z = f(x, y) \quad (2.26a)$$

$$\text{s.t.} \quad g(x, y) \leq 0, \quad (2.26b)$$

$$h(x) = 0, \quad (2.26c)$$

$$x \in \mathcal{X}, \quad (2.26d)$$

$$y \in \mathcal{Y}, \quad (2.26e)$$

where $\mathcal{X}$ represents the set of continuous variables and $\mathcal{Y}$ is the integer variable set. In some works the set $\mathcal{Y}$ is named complicating variable set, since with a fixed y the problem becomes an easier optimization problem to be solved in x .

The set $\mathcal{X}$ is assumed to be a convex compact set, $\mathcal{X} = \{x \mid x \in \mathbf{R}^n, Dx \leq d, x^L \leq x \leq x^U\}$, and the complicating variables correspond to a polyhedral set of integer points, $\mathcal{Y} = \{y \mid y \in \mathbf{Z}^m, Ay \leq a\}$ which in most applications is restricted to 0–1 values, $y \in \{0, 1\}^m$.

Regarding MINLP problems, Geoffrion (1972) states some particularities on MINLP problems, that may arise under some assumptions:

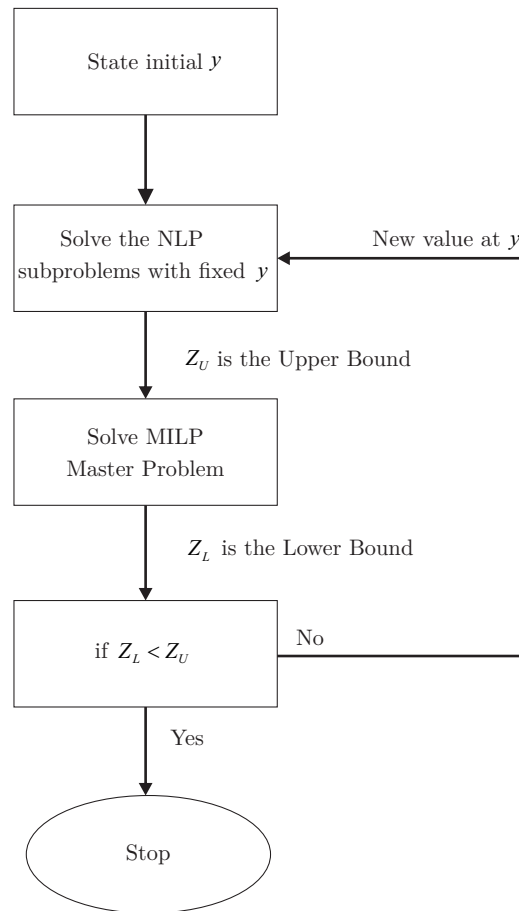
- (a) for fixed y , problem (2.26) separates into a number of independent optimization problems, each involving a different subvector of x ;
- (b) for fixed y , problem (2.26) assumes a well-known special structure, such the classical transportation form, for which efficient solution procedures are available; and
- (c) Problem (2.26) is not a convex program in x and y jointly, but fixing y renders it so in x .

Figure 13 illustrates how these particularities are handled in the process of solving MINLP in a hierarchical approach of solving the subproblems for fixed complicating variables, rendering an upper bound on the solution, and then solving an approximation of the original problem, called master problem that renders a lower bound on the solution.

Such situations occur in a large number of practical applications of mathematical programming and in the literature of large-scale optimization, where the central objective is to exploit particular structures such as the design of effective solution procedures. It has been more than fifty years of studies in this field, and methods have been proposed over the years for solving these problems, such as,

- Branch and Bound;
- Decompositions:

Figure 13 – Block diagram for solving MINLP



Source: Author (2020)

- Outer Approximation (DURAN; GROSSMANN, 1986; QUESADA; GROSSMANN, 1992),
 - Extended Cutting Planes (WESTERLUND; PETTERSSON, 1995),
 - Benders Decomposition, introduced by Benders (1962) and generalized by Geoffrion (1972).
- Combination between Branch and Bound and these Decompositions.

In this context, the techniques Outer Approximation and Benders decomposition are introduced in this work.

2.4.1 Outer Approximation

The main objective of the Outer Approximation (OA) method is to take problems with complicating variables, which, when temporarily fixed, yield a problem significantly easier to handle.

Some assumptions and modifications in the MINLP (2.26) are needed; and the concepts of the Outer Approximation method will allow a better view on the design of the master problem and the subproblem associated to Benders decomposition.

In most applications of interest, Grossmann (2002) considers that the objective and constraint functions are linear in y , and there are no nonlinear equations $h(x) = 0$. Under these assumptions, problem (2.26) becomes:

$$\min_{x,y} \quad z = f(x) + c^T y \quad (2.27a)$$

$$\text{s.t.} \quad g(x) + By \leq 0, \quad (2.27b)$$

$$Ay \leq a, \quad (2.27c)$$

$$y \in \{0, 1\}^m, \quad (2.27d)$$

$$x \in \mathcal{X}. \quad (2.27e)$$

To enable the application of the method, some assumptions are made such as the functions $f(\cdot)$ and $g(\cdot)$ being convex and differentiable. Considering a fixed y^k , a subproblem is associated with (2.27), which is an easier NLP to be solved:

$$S(y^k) : \min_x \quad z(y^k) = f(x) + c^T y^k \quad (2.28a)$$

$$\text{s.t.} \quad g(x) + By^k \leq 0, \quad (2.28b)$$

$$x \in \mathcal{X}. \quad (2.28c)$$

And if $S(y^k)$ is infeasible, a relaxed version is solved instead,

$$F(y^k) : \min \quad \gamma \quad (2.29a)$$

$$\text{s.t.} \quad g(x) + By^k \leq e\gamma, \quad (2.29b)$$

$$x \in \mathcal{X}, \quad (2.29c)$$

where e is a vector of ones of suitable size. Minimizing γ , means that the relaxed region of solution to S is minimized. If $\gamma^k > 0$, the NLP subproblem associated to the MINLP (2.27) is infeasible for $y = y^k$.

The OA method proposed by Duran and Grossmann (1986) arises when NLP subproblems S and F , and the MILP master problem, are solved successively in a cycle of iterations to generate the points (x^k, y^k) , in a relaxed form,

$$z = \min \alpha \quad (2.30a)$$

$$\text{s.t.} \quad \alpha \geq f(x) + c^T y, \quad (2.30b)$$

$$g(x) + By \leq 0, \quad (2.30c)$$

$$Ay \leq a, \quad (2.30d)$$

$$y \in \{0, 1\}^m, \quad (2.30e)$$

$$x \in \mathcal{X}. \quad (2.30f)$$

Even with the relaxation on the objective with the α variable, to obtain an equivalent form of MINLP (2.27) into a MILP, a first order Taylor-series approximation at $x^k \in \mathcal{X}$ of $f(\cdot)$ and $g(\cdot)$ on each iteration k is performed,

$$\left. \begin{aligned} f(x) &\geq f(x^k) + \nabla f(x^k)^T (x - x^k) \\ g(x) &\geq g(x^k) + \nabla g(x^k)^T (x - x^k) \end{aligned} \right\} x^k \in \mathcal{X}. \quad (2.31)$$

Given an optimal solution of K convex NLP subproblems $S(y^k)$ at $y^k = 1, \dots, K$, with points x^k , a relaxed MILP master problem of Outer Approximation is obtained,

$$M_{OA} : z_{OA}^k = \min \alpha_{OA} \quad (2.32a)$$

$$\text{s.t. } \alpha_{OA} \geq c^T y + f(x^k) + \nabla f(x^k)^T (x - x^k), \quad (2.32b)$$

$$g(x^k) + \nabla g(x^k)^T (x - x^k) + B y \leq 0, \quad (2.32c)$$

$$A y \leq a, \quad (2.32d)$$

$$y \in \{0, 1\}^m, \quad x \in \mathcal{X}, \quad \alpha_{OA} \in \mathbf{R}, \quad (2.32e)$$

$$k \in \mathcal{K}, \quad (2.32f)$$

where the set $\mathcal{K}$ is defined, at each iteration k , as

$$\mathcal{K} = \{k \mid x^k \text{ is set to } S(y^k) \text{ or } F(y^k), \forall y^k\}$$

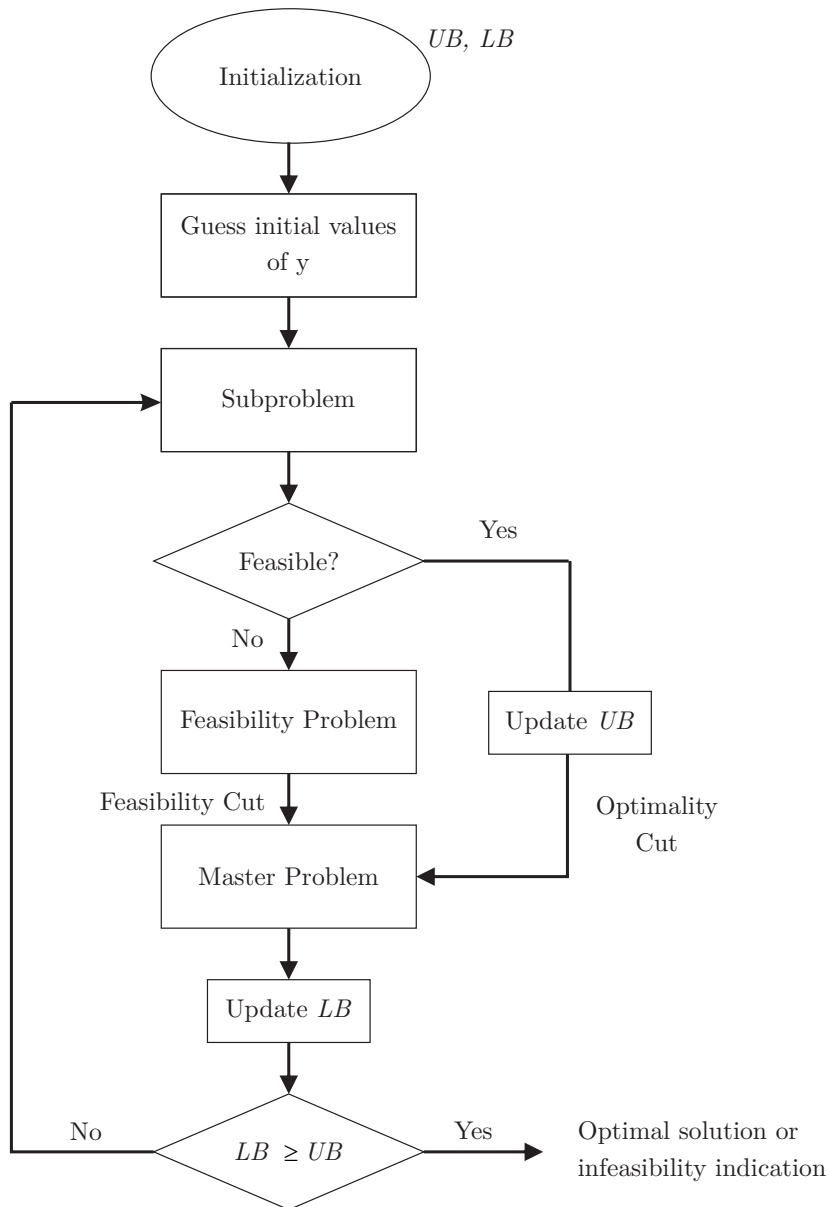
Figure 14 presents a flow chart of the OA algorithm, starting with a initial guess on the complicating variables, i.e., y , then alternating between the subproblems and master problem, until an optimal solution is found. To confirm the convergence of the algorithm, the gap between the UB and the LB can be calculated at each iteration. Figure 15 gives an example of the bounds converging as the iterations progress.

The solution of (2.32) yields a valid lower bound to problem (2.27). This bound is non decreasing with the number of linearization points K . A geometrical interpretation is illustrated in Figure 16. Figure 16a shows that the convex objective function, given a finite number of approximations, will be underestimated.

Considering $x^n = (x_1^n, x_2^n)$, Figure 16b shows that the convex feasible region is overestimated with linearizations (2.31). In order for these linearizations to hold in the process of solving the problem, some conditions must be imposed to add a cut for the feasible region:

- if x^k is strictly inside the feasible region, then the cuts should not be introduced.
- if x^k is outside the feasible region, then the feasibility subproblem $F(y^k)$ produces a cut.
- one should consider in the cut, only binding constraints. In other words, if there are infeasible subproblems, the optimal value u^* , given by problem $F(y^k)$, should be met at equality for the left-hand side of the constraints.

Figure 14 – Flow chart of the outer approximation algorithm



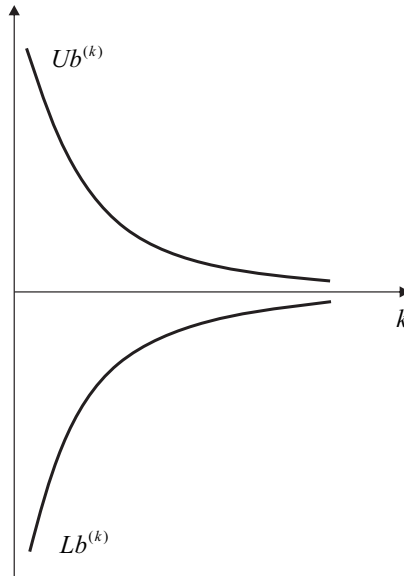
Source: Author (2020)

2.4.2 Benders Decomposition

Benders Decomposition (BD) is a method that decomposes a problem in several simple subproblems and then solves a master problem adding cuts to the feasible region. Given that Benders decomposition is similar to the outer approximation method, here we show how the Benders formulation can be derived from the OA formulation:

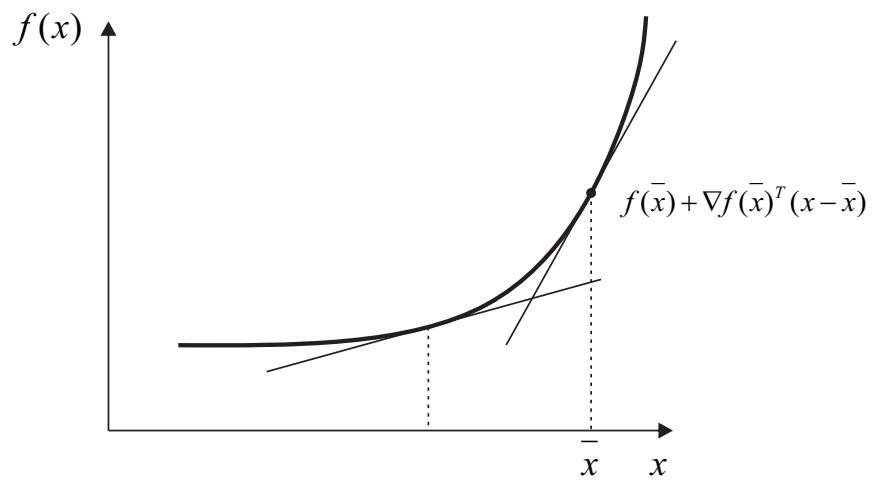
The Benders decomposition method is based on a sequence of projection, outer linearization, and relaxation operations. The model is first projected onto the subspace defined by the set of complicating variables. The resulting formulation is then dualized, and the associated extreme rays and points respectively define the feasibility requirements (feasibility cuts) and the projected costs (optimality cuts) of the complicating variables (RAHMANIANI et al., 2017).

Figure 15 – Trajectory of the solution bounds.

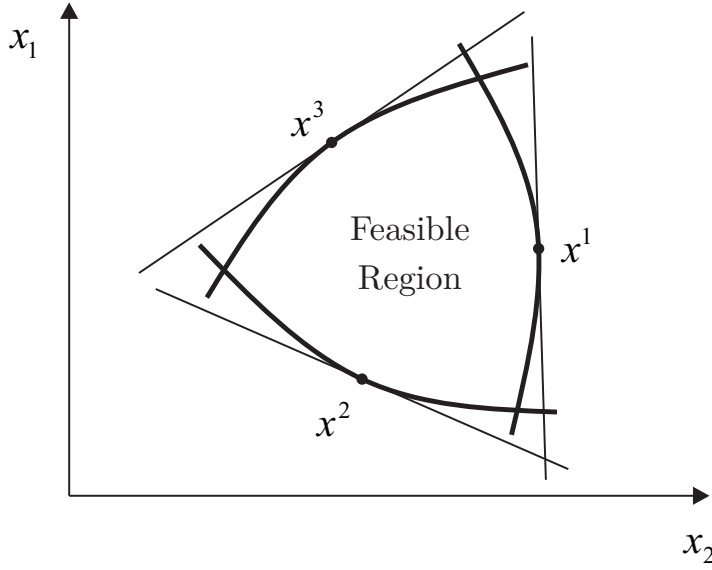


Source: Author (2020)

Figure 16 – Geometrical interpretation of linearizations in master problem.



(a) Underestimate Objective Function



(b) Overestimate Feasible Region

Source: Adapted from Grossmann (2002)

Hence, the method of Benders decomposition solves the equivalent model by applying feasibility and optimality cuts to a relaxation, yielding a master problem and a subproblem, which are iteratively solved to respectively guide the search process and generate the violated cuts.

Figure 17 illustrates the BD algorithm. After obtaining the initial master problem and subproblem, the algorithm alternates between them (starting with the master problem) until convergence is checked, as the OA algorithm. The master problem objective function gives a valid lower bound on the optimal cost because it is a relaxation of the equivalent Benders reformulation. On the other hand, if the solution y produces a feasible subproblem, then the solution of the subproblem provides a valid upper bound for the original problem.

To obtain the Benders decomposition formulation, some steps have to be performed, in order to develop a dual representation of y from the Outer Approximation at y^k given in (2.32).

Once solved for the convex NLP (2.28), let $\mu^k \geq 0$ be the optimal Lagrange multiplier of $g(x) + By^k \leq 0$. Thus, if (2.32c) is pre multiplied by μ^k and moving $\nabla g(x^k)^T(x - x^k)$ to the right hand side, we obtain

$$(\mu^k)^T [g(x^k) + By] \leq -(\mu^k)^T [\nabla g(x^k)^T(x - x^k)] . \quad (2.33)$$

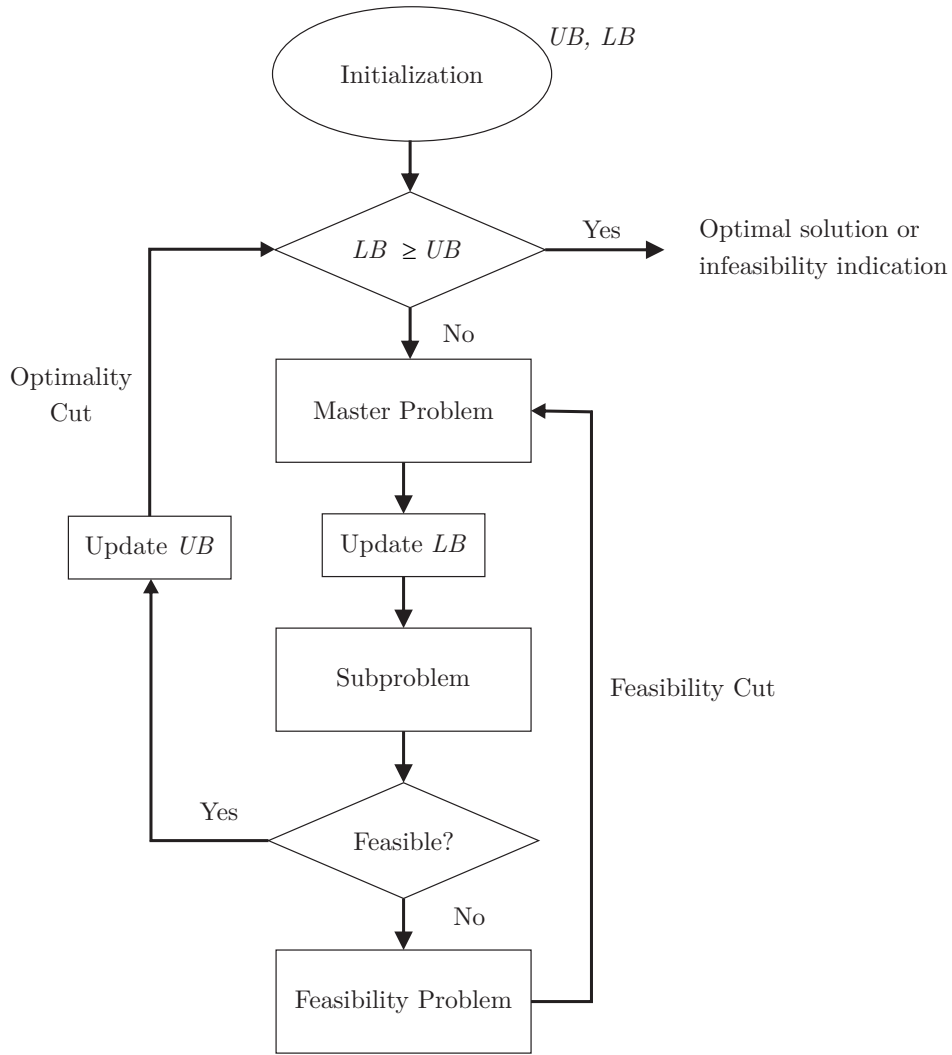
From the subproblem $S(y^k)$ (2.28), the KKT stationarity condition (2.9e) leads to

$$\nabla f(x^k) + \nabla g(x^k)\mu^k = 0. \quad (2.34)$$

If (2.34) is post multiplied by $(x - x^k)$,

$$\nabla f(x^k)^T(x - x^k) = -(\mu^k)^T [\nabla g(x^k)^T(x - x^k)] \quad (2.35)$$

Figure 17 – Benders Algorithm flow chart



Source: Author (2020)

From (2.33) and (2.35),

$$(\mu^k)^T[g(x^k) + By] \leq \nabla f(x^k)^T(x - x^k), \quad (2.36)$$

which substituted into (2.32b) results in

$$\alpha \geq c^T y + f(x^k) + (\mu^k)^T[g(x^k) + By], \quad (2.37)$$

which is called Benders cut, produced when the subproblem $S(y^k)$ (2.28) is feasible. This inequality is known as Benders optimality cut. If problem $S(y^k)$ is infeasible for y^k , a feasibility cut is produced by

$$(\mu^k)^T[g(x^k) + By] \leq 0, \quad (2.38)$$

where μ^k and x^k are obtained by solving $F(y^k)$.

Therefore, after these transformations, the reduced MILP master problem, for Benders decomposition, is stated as:

$$M_{GB} : z_{GB}^k = \min \alpha_{GB} \quad (2.39a)$$

$$\text{s.t. } \alpha_{GB} \geq c^T y + f(x^l) + (\mu^l)^T [g(x^l) + By], \quad l \in K_{feas} \quad (2.39b)$$

$$(\mu^l)^T [g(x^l) + By] \leq 0, \quad l \in K_{infeas} \quad (2.39c)$$

$$Ay \leq a, \quad (2.39d)$$

$$y \in \{0, 1\}^m, \quad x \in \mathcal{X}, \quad \alpha_{GB} \in \mathbf{R}, \quad (2.39e)$$

where K_{feas} is the set of iteration indices at which the subproblem $S(\cdot)$ is feasible, whereas K_{infeas} is the set of indices for which an infeasibility subproblem $F(\cdot)$ had to be solved. Grossmann (2002) offers some remarks on the similarities of Outer Approximation and Benders decomposition, such as:

- M_{OA} and M_{GB} are MILPs that accumulate linearizations as iterations proceed;
- Outer Approximation predicts stronger lower bounds than Benders decomposition;
- Outer Approximation requires fewer iterations;
- The master in Benders decomposition is much smaller.

2.5 SUMMARY

In this chapter the fundamentals in general mathematical optimization with a more detailed presentation of convex optimization, with focus on duality, optimality conditions, and sensitivity analysis have been presented. In the following, Bilevel optimization was introduced, giving some intuition on how multilevel optimization can be approached and how MPECs are a particular case of Bilevel optimization.

Moreover, in this chapter the Outer Approximation and Benders Decomposition as general methods to solve convex MINLPs were also presented. These two decomposition methods transform the MINLP into simpler problems by using outer linearizations or projections of the primal variables onto the dual space.

3 FUNDAMENTALS IN MODEL PREDICTIVE CONTROL

This chapter introduces some concepts of model predictive control (MPC). Some historical comments about this control strategy are stated, the state space MPC algorithm is introduced, and, some essential characteristics of the prediction model and the optimization problem related to the MPC are presented.

3.1 BASIC IDEAS

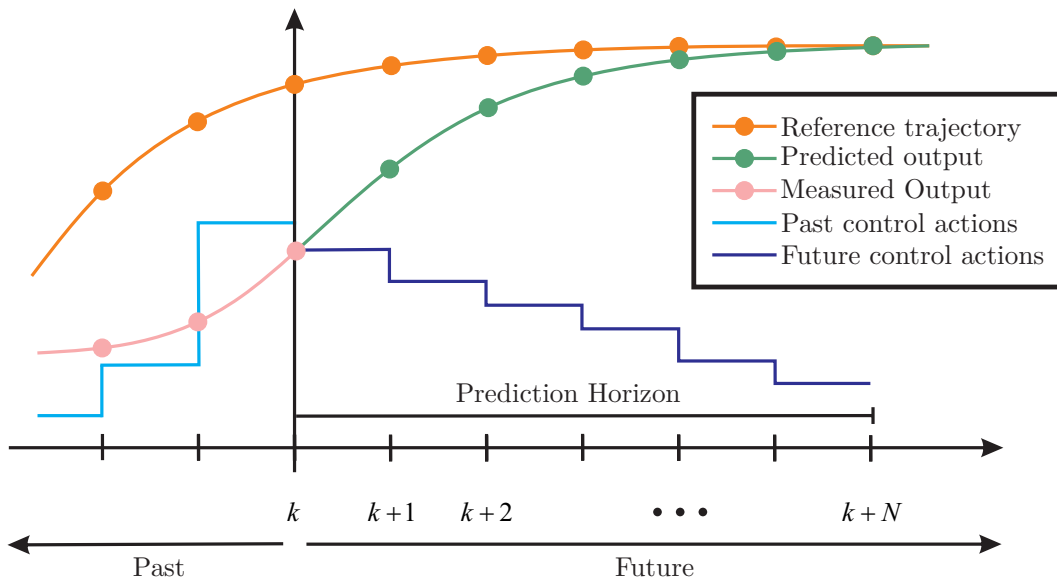
Model Predictive Control (MPC) has developed considerably over the last two decades; it is one of the few advanced control techniques that have achieved a significant impact on industrial process control. This impact can be attributed to the fact that MPC is, perhaps, the most general way of posing the process control problem in the time domain, and the possibility to be applied in SISO (Single Input Single Output) and MIMO (Multiple Inputs Multiple Outputs) systems. Soft and hard constraints can be included in the formulation of the control law, through the solution of optimization problems, by minimizing an objective function over a prediction horizon (NORMEY-RICO; CAMACHO, 2007).

MPC is not a specific control strategy but rather a denomination of a vast set of control methods that have been developed considering some standard ideas and based on the concept of prediction (MENDES, 2016). Figure 18 contains a representation of the output prediction in a time instant. It is possible to see in the figure that the proposed actions generate a predicted behavior that reduces the distance between the value predicted by the model and a reference trajectory.

In the literature, two of the most cited MPC strategies are the Dynamic Matrix Control (DMC) (CUTLER; RAMAKER, 1980) and Generalized Predictive Control (GPC) (CLARKE; MOHTADI; TUFFS, 1987a; CLARKE; MOHTADI; TUFFS, 1987b). Linear models are often used, in the case of DMC, it uses a step response model and in GPC a transfer function model.

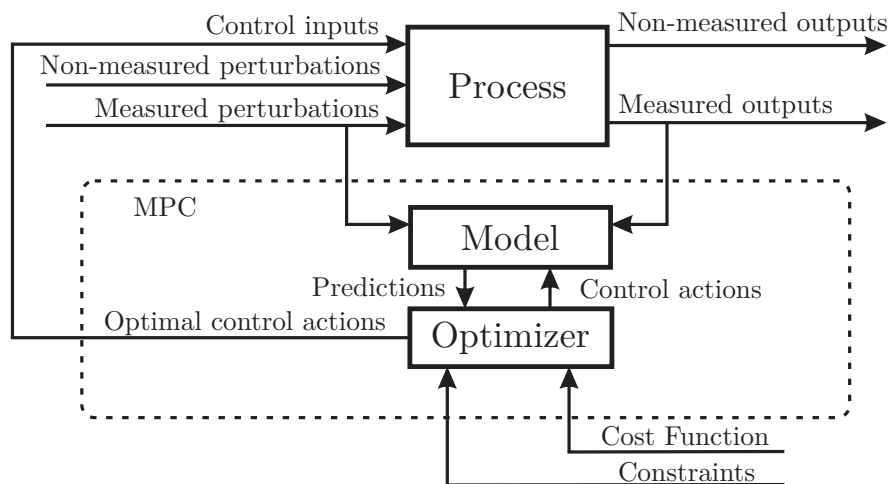
Figure 19 shows the MPC general structure. The MPC strategy has a discrete mathematical model based on the real process that is desired to control. From this model, compared to the output of the real process, a predicted output is calculated in a prediction horizon. One of the essential elements of the MPC is the optimizer, which proposes control actions following an iterative process, in an attempt to achieve a goal and respect restrictions. The objectives and constraints are mathematical expressions established in the design phase of the controller and can take many forms, e.g., quadratic functions are commonly used to penalize the error in the reference tracking.

Figure 18 – Representation of MPC predictions with reference tracking in a time instant.



Source: Adapted from Camacho and Bordons (2007)

Figure 19 – MPC block diagram



Source: Adapted from Brandão (2018)

This model and optimizer structure grant excellent flexibility regarding formulations to be used in the controllers. Such flexibility is confirmed with the several versions of MPC found in the literature.

According to Brandão (2018), there are several ways to classify these controllers taking into account characteristics such as:

- **Model linearity:**

Linear MPC: The model used for the predictions must be linear, such as transfer functions, models in response to step, and linear space of states, among others;

Nonlinear MPC: The model used for the predictions should be non-linear, such as Hammerstein, Wiener, and non-linear space of states, among others;

- **Treatment of uncertainties:**

Nominal MPC: Uncertainties are not addressed in the formulation of the controller;

Robust MPC: Uncertainties are addressed in the formulation of the controller, as long as their limits are known;

Stochastic MPC: Uncertainties are addressed in the formulation controller, as long as the probability distribution is known;

- **How the optimization problem is solved:**

Implicit MPC: The optimization problem is solved at each iteration of the controller;

Explicit MPC: The solution to the optimization problem of the controller is obtained before the start of the operation, according to parametric quadratic programming techniques.

The flexibility of MPC allows its use to solve numerous dynamic problems found in the industry and of academic interest. Such problems have been modeled and adapted to work with the MPC methodology and its variations, such as by Koehler et al. (2018) who integrated the holding and priority control strategies for bus rapid transit systems in a network approach, using deterministic models, that are represented by MPC strategies.

Mendes, Normey-Rico and Alba (2016), Mendes et al. (2016) presented approaches to economic energy management of microgrids that have, in a level, the objective of delivering the energy optimally, and in another level that takes into account the economic cost to provide the energy or even use the power from a consumer that is generating power as well.

Camponogara et al. (2002) introduced a distributed MPC formulation in order to coordinate the computations for the control of the distributed systems. Several applications can benefit from a distributed MPC approach, particularly large-scale applications such as in power systems, water distribution systems, traffic networks, manufacturing plants, and economic systems. In such applications, distributed or decentralized control schemes are desired, where local control inputs are computed using local measurements and reduced-order models of the local dynamics.

3.2 STATE SPACE MPC

Probably one of the essential elements of MPC is the prediction model. Ideally one would choose an appropriate prediction model, i.e., with the following characteristics: (i) adequately represents the dynamics of the process, (ii) allows the calculation of the predictions of the process output, (iii) is intuitive and, (iv) enables a theoretical analysis of the system.

MPC uses several forms to represent the relationships between manipulated inputs, disturbances, and process outputs. Some examples of models used are the impulsive response, the step response, transfer functions, and state space. Each of these models has its applications well defined, e.g., the step response, used by DMC and its variants, or the transfer function models used on GPC.

DMC is widely used in the industry due to the ease of obtaining this type of model, but they cannot be used with unstable systems (CAMACHO; BORDONS, 2007). State-space models have been increasingly used in the development of predictive controllers, because they allow to systematically describe multivariable and complex behaviors, in addition to the analysis of characteristics such as controllability and observability of the system.

According to Camacho and Bordons (2007), the state space model is, perhaps, more widespread in the academic research community as the derivation of the controller is very simple even for the multi-variable case. The state-space description allows a straightforward expression of stability and robustness criteria.

The prediction model is composed of the process and disturbance dynamics, which are the most important elements in predictive control algorithms. As MPC bases its control law computation in the prediction model, all the advantages of predictive control may be lost if the model is not faithful to the process dynamics (MENDES, 2016).

Consider a process with n_x states, n_u inputs, and n_y outputs which is described by the following continuous-time state-space equations:

$$\dot{x}(t) = f(x(t), u(t)) \quad (3.1a)$$

$$y(t) = g(x(t), u(t)), \quad (3.1b)$$

in which x is the state vector, u represents the input vector, and y is the output vector. This is a continuous representation of the dynamic of the states, Equation (3.1a), and the dynamic of the output of the system related to the measurement of the state, Equation (3.1b).

Due to the discrete nature of MPC algorithms, the discretization of continuous dynamic models is necessary. There are several ways to discretize a continuous-time system, such as Zero-Order-Hold (ZOH), Forward Euler, Backward Euler, and Tustin, among others (FADALI; VISIOLI, 2013).

The Forward Euler method, which consists of approximating the derivatives of the continuous model, is one of the simplest manners of discretization, like as follows,

$$\dot{x}(t) \approx \frac{x(t + T_s) - x(t)}{T_s} \quad (3.2)$$

where T_s is the sampling time. Substituting this Forward Euler relation in Equation (3.1), implies:

$$\frac{x(t + T_s) - x(t)}{T_s} = f(x(t), u(t)), \quad (3.3)$$

then,

$$x(k + 1) = x(k) + T_s f(x(k), u(k)) \quad (3.4a)$$

$$y(k) = g(x(k), u(k)). \quad (3.4b)$$

In this discrete-time state-space formulation, an approximation of the continuous time model is obtained. This method of discretization is used due to the ease in obtaining the new representation of the model. However, it is necessary to be careful when choosing the sampling time, T_s , and poor conditioning of the continuous model may render the use of the model unfeasible.

The most common MPC algorithms use linear models to solve their control problems due to their mathematical basis, and the solution of such models being well defined in the literature. The linear, discrete-time state space formulation is presented in the following.

$$x(k+1) = Ax(k) + Bu(k) \quad (3.5a)$$

$$y(k) = Cx(k) + Du(k), \quad (3.5b)$$

where $A \in \mathbf{R}^{n_x \times n_x}$, $B \in \mathbf{R}^{n_x \times n_u}$, $C \in \mathbf{R}^{n_y \times n_x}$, and $D \in \mathbf{R}^{n_y \times n_u}$ are the system matrices with suitable dimensions; x is the state vector, u represents the input vector, and y is the output vector. In this model k represents the discrete time (the sampling time is not written explicitly for simplicity).

The future predicted values of the process state $\hat{x}$ and the output $\hat{y}$ can be calculated recursively until time $k+j$ according to:

$$\hat{x}(k+j|k) = A^j x(k) + \sum_{i=0}^{j-1} A^{j-i-1} Bu(k+i) \quad (3.6a)$$

$$\hat{y}(k+j|k) = CA^j x(k) + \sum_{i=0}^{j-1} CA^{j-i-1} Bu(k+i) + Du(k+j). \quad (3.6b)$$

Here the notation $(\cdot|k)$ is introduced to mean the value calculated, over the prediction horizon N and the control horizon N_u in which $N_u \leq N$, with the information available at the instant k . From the predicted future control variation, the predictions on future states and outputs are calculated as follows,

$$\underbrace{\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+N|k) \end{bmatrix}}_{\hat{\mathbf{x}}} = \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\mathbf{A}} x(k) + \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & \sum_{i=0}^{N-N_u} A^i B \end{bmatrix}}_{\mathbf{B}} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}}_{\mathbf{u}} \quad (3.7a)$$

$$\underbrace{\begin{bmatrix} \hat{y}(k+1|k) \\ \hat{y}(k+2|k) \\ \vdots \\ \hat{y}(k+N|k) \end{bmatrix}}_{\hat{\mathbf{y}}} = \underbrace{\begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix}}_{\mathbf{C}} x(k)$$

$$+ \underbrace{\begin{bmatrix} CB & D & \dots & 0 \\ CAB & CB & \dots & D \\ \vdots & \vdots & \ddots & \vdots \\ CA^{N-1}B & CA^{N-2}B & \dots & \sum_{i=0}^{N-N_u} CA^i B \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N-1) \end{bmatrix}}_{\mathbf{u}} \quad (3.7b)$$

In a compact form the future states and output predictions are

$$\hat{\mathbf{x}} = \mathbf{A}x(k) + \mathbf{B}\mathbf{u} \quad (3.8a)$$

$$\hat{\mathbf{y}} = \mathbf{C}x(k) + \mathbf{D}\mathbf{u}. \quad (3.8b)$$

The relation $\mathbf{A}x(k)$ is also known as the free response of the system, and $\mathbf{B}\mathbf{u}$ is known as the forced response (CAMACHO; BORDONS, 2007).

For the prediction computation, as can be seen, the measurements of the state vector are needed. In various cases, the states can not be obtained directly from the process. However, in the literature, techniques can be found to retrieve state values, including state observers, Kalman filters (KALMAN, 1960), and even Machine Learning methods are broadly studied (CHING; BECK; PORTER, 2006).

3.2.1 Optimization Problem in MPC

Regarding the strategy chosen for modeling the problem, when using MPC, the objective function or cost function can be written in several forms in order to obtain the control law. The most common objective is to minimize the error between the future output y and desired reference w , which is achieved by computing a suitable control variation Δu that imposes a penalty in the objective function. Usually, the objective function is described as follows:

$$J = \sum_{j=N_1}^{N_2} \|y(k+j|k) - w(k+j)\|_Q^2 + \sum_{j=0}^{N_u-1} \|\Delta u(k+j|k)\|_R^2, \quad (3.9)$$

where N_1 and N_2 are the minimum and maximum prediction horizons respectively, and N_u is the control horizon. These indexes define the limits in which it is desirable for the output to follow the reference and when it is important to limit the control action. Certain modifications on the horizon values may be used to compensate for transmission delays and non-minimum phase (CAMACHO; BORDONS, 2007). By changing N_u it is possible to adjust the time instants at which the control action is penalized. Q and R are positive definite matrices that penalize the errors on trajectory tracking and control variation.

In fact, all real processes are subject to constraints, and these constraints are related to economic restrictions, operational restrictions, and minimum and maximum ranges for actuators. Mendes (2016) states that constructive reasons, such as safety or environmental ones, can cause limits on the process variables as states and outputs. The operating conditions are generally

defined by the intersection of certain constraints so that the control system will operate close to the boundaries.

The controller should be capable of anticipating these operational points and correcting the input values such that the process remains stable and does not violate the operational values. The MPC strategy is useful in predicting this violation on the operational values, because it is possible, within the MPC related optimization problem, to introduce boundaries into the constraints of the problem within the prediction horizon.

The solution of the constrained MPC is carried out by minimizing the objective function, through convex optimization algorithms, since the minimization of the quadratic convex function (3.9) subject to linear constraints renders a quadratic programming problem (QP). The algorithms solve similar problems, such the one that follows,

$$\min_u J = \sum_{j=N_1}^{N_2} \|y(k+j|k) - w(k+j)\|_Q^2 + \sum_{j=0}^{N_u-1} \|\Delta u(k+j|k)\|_R^2, \quad (3.10a)$$

$$\text{subject to: } x(k+j+1|k) = Ax(k+j|k) + Bu(k+j|k), \quad (3.10b)$$

$$y(k+j|k) = Cx(k+j|k) + Du(k+j|k), \quad (3.10c)$$

$$x(k|k) = x(k), \quad (3.10d)$$

$$y^{\min} \leq y(k+j|k) \leq y^{\max}, \quad (3.10e)$$

$$u^{\min} \leq u(k+j|k) \leq u^{\max}, \quad (3.10f)$$

$$\Delta u^{\min} \leq u(k+j|k) - u(k+j-1|k) \leq \Delta u^{\max}, \quad (3.10g)$$

in which, constraints (3.10b) and (3.10c) represent system dynamics, (3.10e) through (3.10g) define lower and upper bounds on the variables, (3.10d) is the constraint that refers to initial conditions.

3.2.2 Obtaining the Control Law

With the optimization problem stated, for a state-space MPC formulation, considering an optimal solution for the convex QP problem, a control sequence is obtained in the prediction horizon N . The following steps exemplify one way to use the MPC strategy to obtain the control law and apply it to real system.

To compute the control action the following steps are executed:

1. update the initial conditions of the states, $x(k)$;
2. compute $y(k+j)$ as a function of future control actions;
3. substitute y and w in the objective function J ;
4. minimize J considering the system dynamics and constraints.

It is crucial to emphasize the concept of receding horizon control. Despite the solution of the objective function resulting in an array of control actions through the horizon, only the first control action will be applied, and all control actions will be recalculated at the next step according to measurements from the real system (MENDES, 2016).

3.3 SUMMARY

In this chapter a brief overview of Model Predictive Control, such as its vast use in the industry and literature studies, arguably for being a reliable method in discrete-time control theory is presented. Several key concepts are introduced, like control horizon and prediction horizon, and the use of optimization to make use of measurements of a process and then attach it to a model in order to obtain an optimized control action sequence.

MPC methods are discussed, like DMC, GPC, and state-space MPC, the latter being presented in its continuous-time formulation. A simple discretization technique is shown for the linear state-space model, and its use on the prediction of the system output.

Using the discrete-time state-space formulation, the control law is obtained using the information of the model, in addition to the measurements of the process and techniques of optimization.

4 DECOMPOSITIONS FOR MPC OF RESOURCE-CONSTRAINED DYNAMIC SYSTEMS

In this chapter, the formulation for the MPC of a class of resource-constrained dynamic systems is introduced, and later, hierarchical decomposition strategies are formulated, namely Bilevel and Benders Decomposition.

The MPC refers to an ample range of control methods which make explicit use of a process model to obtain the control signals by optimizing an objective function over a prediction horizon (CAMACHO; BORDONS, 2007). At the current instant k , the optimization produces an optimal control sequence, but only the first control signal is applied to the process. At instant $k + 1$ with the measurements updated, the process is repeated over the prediction horizon.

Let $\mathcal{M} = \{1, \dots, M\}$ be the set of subsystems, $\mathcal{R} = \{1, \dots, R\}$ be the set of the resources, $\mathcal{N}_u = \{0, \dots, N_u\}$ define the control horizon, and N_1 and N_2 establish the minimum and maximum prediction horizon for the outputs. The MPC problem of interest is given by:

$$P : \min J = \sum_{m=1}^M \sum_{j=N_1}^{N_2} \|y_m(k+j|k) - w_m(k+j)\|_{Q_m}^2 + \sum_{m=1}^M \sum_{j=0}^{N_u-1} \|\Delta u_m(k+j|k)\|_{W_m}^2 \quad (4.1a)$$

while, for all $m \in \mathcal{M}$ being subject to:

$$x_m(k+j+1|k) = A_m x_m(k+j|k) + B_m u_m(k+j|k), \quad j = 0, \dots, N_2 - 1 \quad (4.1b)$$

$$y_m(k+j|k) = \begin{cases} C_m x_m(k+j|k) + D_m \Delta u_m(k+j|k), & j = N_1, \dots, N_u - 1 \\ C_m x_m(k+j|k), & j = N_u, \dots, N_2 \end{cases} \quad (4.1c)$$

$$u_m(k+j|k) = \begin{cases} u_m(k+j-1|k) + \Delta u_m(k+j|k), & j = 0, \dots, N_u - 1 \\ u_m(k+j-1|k), & j = N_u, \dots, N_2 - 1 \end{cases} \quad (4.1d)$$

$$x_m(k|k) = x_m(k) \quad (4.1e)$$

$$u_m(k-1|k) = u_m(k-1) \quad (4.1f)$$

$$y_m^{\min} \leq y_m(k+j|k) \leq y_m^{\max}, \quad j = N_1, \dots, N_2 \quad (4.1g)$$

$$u_m^{\min} \leq u_m(k+j|k) \leq u_m^{\max}, \quad j = 0, \dots, N_u - 1 \quad (4.1h)$$

$$\Delta u_m^{\min} \leq \Delta u_m(k+j|k) \leq \Delta u_m^{\max}, \quad j = 0, \dots, N_u - 1 \quad (4.1i)$$

and for $r = 1, \dots, R$, $j = 0, \dots, N_2 - 1$:

$$\sum_{m=1}^M s_{r,m}^T u_m(k+j|k) \leq s_r^{\max}(k+j) \quad (4.1j)$$

in which:

- $x_m(k)$ is the system state at time k and $x_m(k+j|k)$ is the state prediction for time $k+j$ calculated with the information available until time k ;
- $y_m(k+j|k)$ is the predicted output at time $k+j$;
- $w_m(k+j)$ is the desired output trajectory;
- $u_m(k+j|k)$ is the predicted control input and $\Delta u_m(k+j|k)$ is the predicted control variation;
- $Q_m = Q_m^T$ and $W_m = W_m^T$ are positive definite matrices that penalize the errors on trajectory tracking and control variation, respectively;
- $x_m(k)$ and $u_m(k-1)$ are known values with the initial conditions;
- A_m, B_m, C_m , and D_m are the system matrices of appropriate dimensions;
- $\|x\|_Q = \sqrt{x^T Q x}$ is the vector norm induced by a positive definite matrix Q ;
- $y_m^{\min}, y_m^{\max}, u_m^{\min}, u_m^{\max}, \Delta u_m^{\min}$, and $\Delta u_m^{\max}$ are the imposed bounds on outputs, control signals and control variation, respectively;
- $s_r^{\max}$ is the amount of resource r available at time k and $s_{r,m}$ defines the rate of consumption by subsystem m .

4.1 BILEVEL OPTIMIZATION

In order to present a more convenient form for bilevel optimization, as in (CAMPONOGARA et al., 2020), problem P is recast as follows,

$$\min_z f = \sum_{m=1}^M f_m(z_m) \quad (4.2a)$$

$$\text{s.t.: for } m = 1, \dots, M :$$

$$h_m(z_m) = 0 \quad (4.2b)$$

$$g_m(z_m) \leq 0 \quad (4.2c)$$

$$g(z) \leq s, \quad (4.2d)$$

where $z_m = (x_m(k+j), y_m(k+j), u_m(k+j), \Delta u_m(k+j) : \forall j)$ is a vector formed by all the variables of the subsystem m ; $s = (s_r^{\max}(k+j) : \forall r, j)$ is a vector formed by the resources available at all times and for all subsystems; and $z = (z_m \forall m \in \mathcal{M})$ has all the subsystem variables. Thus, the vector functions h_m , g_m and g represent the equalities (4.1b)-(4.1f), inequalities (4.1g)-(4.1i), and the resource inequalities (4.1j), respectively.

Let $ss = (ss_{r,m}(k+j) : \forall r, j, m)$ as the vector of resource allocations, in which $ss_{r,m}(k+j)$ is the amount of resource r allocated to subsystem m , at time $k+j$, problem P can be framed as a bilevel optimization problem, where the master problem is given by,

$$U : \min_{ss} f(ss) = \sum_{m=1}^M f_m(z_m(ss_m)) \quad (4.3a)$$

$$\text{s.t.: for } r = 1, \dots, R, j = 0, \dots, N_2 - 1 :$$

$$\sum_{m=1}^M ss_{r,m}(k+j) \leq s_r^{max}(k+j) \quad (4.3b)$$

$$ss_{r,m}(k+j) \geq 0, \forall m \in \mathcal{M} \quad (4.3c)$$

where $ss_m = (ss_{r,m}(k+j) : \forall r, j)$ are the resources allocated to subsystem m , such that $ss = (ss_1, \dots, ss_M)$. The lower-level problem L consists of solving a set of sub-problems $\{L_m\}$, one for each subsystem $m \in \mathcal{M}$, being defined by,

$$L_m(ss_m) : z_m(ss_m) = \underset{z_m}{\operatorname{argmin}} f_m(z_m) \quad (4.4a)$$

$$\text{s.t.: } h_m(z_m) = 0 \quad (4.4b)$$

$$g_m(z_m) \leq 0 \quad (4.4c)$$

$$\forall r \in \mathcal{R}, j = 0, \dots, N_2 - 1 :$$

$$s_{r,m}^T u_m(k+j|k) \leq ss_{r,m}(k+j), \quad (4.4d)$$

in which $\mathcal{R} = \{1, \dots, R\}$.

This work adopts an approach where the master problem optimizes over the decision space of resource allocations ss , followed by the optimization of the lower-level subproblems and the computation of sensitivities $\partial f_m / \partial ss$ to compute improving directions for the upper-level algorithm (CAMPONOGARA et al., 2020).

Since the objective of the master problem consists of the sum of the objectives of the lower-level problems, the sensitivities for the master problem can be easily calculated from the sensitivities of the lower problems. This property enables parallel computation of the sensitivities as part of the solution of the subproblems. The master problem in Eq. (4.3) can be solved by any gradient-based method, such as the interior-point (IP) or sequential quadratic programming (SQP). In this work, the master is solved by an interior-point method.

For use in the two-level approach, the sensitivities are retrieved from the solution of a reformulation of the subproblem L_m given in Eq. (4.4), as follows:

$$L_m(ss_m^*) : z_m(ss_m^*) = \underset{z_m}{\operatorname{argmin}} f_m(z_m) \quad (4.5a)$$

$$\text{s.t.: } h_m(z_m) = 0 \quad (4.5b)$$

$$g_m(z_m) \leq 0 \quad (4.5c)$$

$$\forall r \in \mathcal{R}, j = 0, \dots, N_2 - 1 :$$

$$s_{r,m}^T u_m(k+j|k) \leq ss_{r,m}(k+j) \quad (4.5d)$$

$$ss_{r,m}(k+j) - ss_{r,m}^*(k+j) = 0, \quad (4.5e)$$

such that the sensitivities are calculated as:

$$\left. \frac{\partial f(ss_m)}{\partial ss_{r,m}(k+j)} \right|_{ss=ss^*} = \lambda_{r,m}(k+j), \quad (4.6)$$

where $\lambda_{r,m}(k+j)$ is the Lagrange multiplier associated with constraint (4.5e) for the KKT conditions (BOYD; VANDENBERGHE, 2012), at the solution $z_m(ss_m^*)$. The Lagrange multipliers, or sensitivities, provide the directions along which the current solution to the master problem can be improved. Notice that the allocation vector ss_m^* is informed by the master problem, which is then taken as fixed in (4.5e), whereas ss_m are local variables.

Camponogara et al. (2020) point out that the bilevel approach just described assumes that the initial allocation ss renders the subproblems $\{L_m(ss_m)\}$ feasible, otherwise the sensitivities would not be defined. However, infeasibility could be handled with an l_1 penalty based on a constraint relaxation, akin to the phase I strategy used in convex optimization (BOYD; VANDENBERGHE, 2012). The phase I strategy defines the subproblem L_m , whereby a slack variable γ_m is minimized and the constraints are relaxed by this slack variable. The objective function of the master would be the sum of the γ_m variables. If a solution ss is found to drive the master's objective to zero, then ss is a feasible allocation which can be used as a starting point.

4.2 BENDERS DECOMPOSITION

Here we begin by introducing the subproblems rather than the master problem for the Benders decomposition. For a feasible resource allocation ss , the optimization subproblem, also known as primal subproblem, is given by

$$BO(ss) : bo(ss) = \min_z \sum_{m=1}^M f_m(z_m) \quad (4.7a)$$

$$\text{s.t.: for } m = 1, \dots, M$$

$$h_m(z_m) = 0 \quad (4.7b)$$

$$g_m(z_m) \leq 0 \quad (4.7c)$$

$$R_m z_m - S_m ss_m \leq 0, \quad (4.7d)$$

in which R_m and S_m are suitable matrices that define the constraints (4.5e). In the particular problem of concern, S_m is the identity matrix and $R_m z_m$ is effectively $\tilde{R}_m u_m$ since only the terms $s_{r,m}^T u_m(k+j|k)$ are needed in the constraint.

Notice that $bo(ss)$ induces an upper bound for a feasible ss , meaning one vector ss that satisfies constraints (4.3b)-(4.3c) and which renders $BO(ss)$ feasible. Clearly $bo(ss)$ can be

computed in parallel as follows

$$bo(ss) = \sum_{m=1}^M bo_m(ss_m) \quad (4.8)$$

for which

$$BO_m(ss_m) : bo_m(ss_m) = \min_{z_m} f_m(z_m) \quad (4.9a)$$

$$\text{s.t.: } h_m(z_m) = 0 \quad (4.9b)$$

$$g_m(z_m) \leq 0 \quad (4.9c)$$

$$R_m z_m - S_m ss_m \leq 0. \quad (4.9d)$$

At iteration p of the Benders algorithm, let $ss^{(p)}$ be the resource allocation vector and assume that $BO(ss^{(p)})$ is feasible. Let $\mu_m^{(p)}$ be the Lagrange multipliers associated with the resource constraint (4.9d), at the optimal solution $z_m^{(p)}$. Then the following optimality cut can be introduced to the Benders master problem:

$$\alpha_B \geq \sum_{m \in \mathcal{M}} f_m(z_m^{(p)}) + \sum_{m \in \mathcal{M}} \mu_m^{(p)T} [R_m z_m^{(p)} - S_m ss_m], \quad (4.10)$$

with α_B being the lower bound of the overall objective of P . Due to the complementarity conditions, the local constraints (4.9b) and (4.9c) do not play a part in the cutting plane. The decision space of the Benders master problem consists of the allocation vector ss and the lower bound α_B .

For an infeasible resource allocation, the following feasibility subproblem has to be solved:

$$BF(ss) : bf(ss) = \min_{\gamma \geq 0, z} \gamma \quad (4.11a)$$

$$\text{s.t.: for } m = 1, \dots, M$$

$$h_m(z_m) = 0 \quad (4.11b)$$

$$g_m(z_m) \leq \gamma \cdot e_m \quad (4.11c)$$

$$R_m z_m - S_m ss_m \leq \gamma \cdot e_m, \quad (4.11d)$$

with $e_m = (1, 1, \dots, 1)$ being a vector of suitable dimension and γ a nonnegative scalar. The optimal γ can be obtained by solving an auxiliary subproblem only for the infeasible BO_m . Let $\mathcal{M}_{infeas} = \{m \in \mathcal{M} : BO_m(ss_m) \text{ is infeasible}\}$. Then $BF(ss)$ is solved as follows

$$bf(ss) = \max \{bf_m(ss_m) : m \in \mathcal{M}_{infeas}\} \quad (4.12)$$

and, for all $m \in \mathcal{M}_{infeas}$, we solve the problem:

$$BF_m(ss_m) : bf_m(ss_m) = \min_{\gamma_m \geq 0, z_m} \gamma_m \quad (4.13a)$$

$$\text{s.t.: } h_m(z_m) = 0 \quad (4.13b)$$

$$g_m(z_m) \leq \gamma_m \cdot e_m \quad (4.13c)$$

$$R_m z_m - S_m s s_m \leq \gamma_m \cdot e_m. \quad (4.13d)$$

In fact if $s s_m$ is feasible for BO_m , then the corresponding BF_m will have an optimal value $bf_m(s s_m) = 0$ and the feasibility subproblem does not have to be solved. Assume that $s s^{(p)}$ is an infeasible allocation at iteration p . Let $\mathcal{M}^{(p)} \subset \mathcal{M}$ be the subset of subproblems m such that $\gamma_m = bf(ss)$. Then the feasibility cut is as follows

$$\sum_{m \in \mathcal{M}^{(p)}} \left\{ \mu_{m,c}^{(p)T} g_m(z_m) + \mu_{m,d}^{(p)T} [R_m z_m^{(p)} - S_m s s_m] \right\} \leq 0 \quad (4.14)$$

in which $\mu_{m,c}^{(p)}$ and $\mu_{m,d}^{(p)}$ are the Lagrange multipliers associated with (4.13c) and (4.13d), respectively.

At iteration p an optimality cut is obtained by successfully solving $BO(ss^{(p)})$, or else a feasibility cut by solving $BF(ss^{(p)})$. Let $\mathcal{O}^{(p)}$ and $\mathcal{F}^{(p)}$ be the indices of the iterations for which an optimality and feasibility cut was produced, respectively. Then, the Benders master problem at iteration p can be stated as follows:

$$BM(p) : \min_{ss \geq 0, \alpha_B} \alpha_B \quad (4.15a)$$

$$\text{s.t.: } R s s \leq s \quad (4.15b)$$

$$\alpha_B \geq \sum_{m \in \mathcal{M}} \left\{ f_m(z_m^{(i)}) + \mu_m^{(i)T} [R_m z_m^{(i)} - S_m s s_m] \right\}, i \in \mathcal{O}^{(p)} \quad (4.15c)$$

$$\sum_{m \in \mathcal{M}^{(i)}} \left\{ \mu_{m,c}^{(i)T} g_m(z_m) + \mu_{m,d}^{(i)T} [R_m z_m^{(i)} - S_m s s_m] \right\} \leq 0, i \in \mathcal{F}^{(p)} \quad (4.15d)$$

with R being a suitable matrix that models the resource constraint (4.3b). Notice that $BM(p)$ is a LP.

Algorithm 1 formalizes the Benders decomposition. The algorithm does not require a feasible starting point, since it can produce a feasible solution if one exists with the aid of the feasibility cuts.

4.3 NUMERICAL EXPERIMENTS

To validate the hierarchical decomposition approaches, these were implemented and tested within the solution of MPC problems with resource constraints. Also, to give insights into the use of parallel computing, concerning the impact in computation time as the number of processors or cores increases. To that end, problems were generated at random, as detailed in Table 2, where the numerical data is presented

Algorithm 1: Benders Decomposition

Input: initial lower bound $lb^{(0)} := -\infty$ and upper bound $ub^{(0)} := \infty$, and tolerance $\tau > 0$;
 $\mathcal{O}^{(0)} := \emptyset$, $\mathcal{F}^{(0)} := \emptyset$;
 $p := 0$;
repeat
 Solve master problem $BM(p)$ given in (4.15) to obtain a solution $ss^{(p)}$ and objective $\alpha_B^{(p)}$;
 Update lower bound: $lb^{(p+1)} := \max\{lb^{(p)}, \alpha_B^{(p)}\}$;
 Parallel solve $\{BO_m(ss_m^{(p)})\}_{m \in \mathcal{M}}$ and obtain a solution $z^{(p)}$ for $BO(ss^{(p)})$;
 if $z^{(p)}$ is feasible for $BO(ss^{(p)})$ **then**
 Update upper bound $ub^{(p+1)} := \min\{ub^{(p)}, f(z^{(p)})\}$;
 Add an optimality cut (4.10) to $\mathcal{O}^{(p)}$ and obtain $\mathcal{O}^{(p+1)}$;
 Let $\mathcal{F}^{(p+1)} := \mathcal{F}^{(p)}$;
 else
 Keep upper bound $ub^{(p+1)} := ub^{(p)}$;
 Parallel solve $\{BF_m(ss_m^{(p)})\}_{m \in \mathcal{M}}$ and obtain a solution $(ss^{(p)}, \gamma^{(p)})$ for $BF(ss^{(p)})$;
 Add a feasibility cut (4.14) to $\mathcal{F}^{(p)}$ and obtain $\mathcal{F}^{(p+1)}$;
 Let $\mathcal{O}^{(p+1)} := \mathcal{O}^{(p)}$;
 end
 $p := p + 1$;
until $\frac{ub^{(p)} - lb^{(p)}}{ub^{(p)}} \leq \tau$;
Output: $z^{(p)}$, $lb^{(p)}$, $ub^{(p)}$;

Table 1 gives a glance at the size of the proposed problems, for a given number of subsystems M and a stipulated prediction horizon T that is equal to the control horizon. Notice that the increase in the number of variables in the master problem and the subproblems is significant. The number of variables is all added together in the column of subproblems.

Table 1 – Number of variables of synthetic problems, for each decomposition strategy.

M	T	Bilevel		Benders Decomposition	
		Master Problem	Subproblems	Master Problem	Subproblems
20	4	60	540	61	480
	6	100	860	101	760
	8	140	1180	141	1040
40	4	120	1080	121	960
	6	200	1720	201	1520
	8	280	2360	281	2080
80	4	240	2160	241	1920
	6	400	3440	401	3040
	8	560	4720	561	4160

The Julia Programming Language (BEZANSON et al., 2017) was used for being a new

Table 2 – Problem data related to Table 3

M	T	seed	y_{min}/y_{max}	u_{min}/u_{max}	$\Delta u_{min}/\Delta u_{max}$	r_{max}
20	{4, 6, 8}	2	{−25, 25}	{0, 3}	{−3, 3}	2
40	{4, 6, 8}	2	{−25, 25}	{0, 3}	{−3, 3}	2
80	{4, 6, 8}	2	{−25, 25}	{0, 3}	{−3, 3}	2

approach to numerical computing. Julia is an open source language developed at Massachusetts Institute of Technology (MIT) in 2012. Besides being a general purpose language which can be used to write any application, many of its features are well-suited for numerical analysis and computational applications. Julia is dynamically typed, feels like a scripting language, and has good support for interactive use. Julia has been downloaded over 13 million times and the Julia community has registered over 3,000 Julia packages for community use. These include various mathematical libraries, data manipulation tools, and packages for general purpose computing. In addition to these, libraries from Python, R, C/Fortran, C++, and Java can easily be used (JULIA, 2020).

The algorithm was implemented using the modeling language for mathematical programming, JuMP¹ (DUNNING; HUCHETTE; LUBIN, 2017), which offers a high-level interface with solvers like IPOPT (WÄCHTER; BIEGLER, 2006), Gurobi, Artelys KNITRO, IBM CPLEX and many others, for solving optimization problems.

The implementation was made in a computer with an Intel® Xeon™ CPU E5-2665 (2.40GHz and 20MB of cache), with 8 cores and 16 threads, 40 GB of RAM memory, and in a Linux environment. So, according to the need of parallel iterations, the subproblem instances were allocated to be solved at different cores by the algorithm.

For the Bilevel approach, demonstrated in Eq. (4.3) and (4.4), the subproblems are quadratic problems (QP) and the master problem is a non-linear problem (NLP). The Gurobi solver, version 9.0.1, was used to solve the subproblems, whereas for solving the master problem the IPOPT solver, version 3.12.10, was chosen. The tolerance for Gurobi was set at 10^{-6} and for IPOPT at 10^{-4} .

For the Benders Decomposition approach, both the master and the subproblems were solved by Gurobi, and the relative gap, defined in Eq. (4.16), between LB and UB was set to 10^{-2} %.

$$REL_{gap} = \frac{UB - LB}{(UB + 1e^{-20})} \cdot 100 \quad (\%) \quad (4.16)$$

Results for the numerical experiments are shown in the Table 3. The column labeled f^* shows the overall optimal objective. For each instance of the problems, the number of cores,

¹ During the development of this dissertation, a bug on JuMP was found, that did not allow the proper use of the modeling language to handle Bilevel Optimization. The bug report and its solution can be found in <<https://github.com/jump-dev/JuMP.jl/issues/1735>>.

Table 3 – Computational Analysis of Benders and Bilevel Decompositions

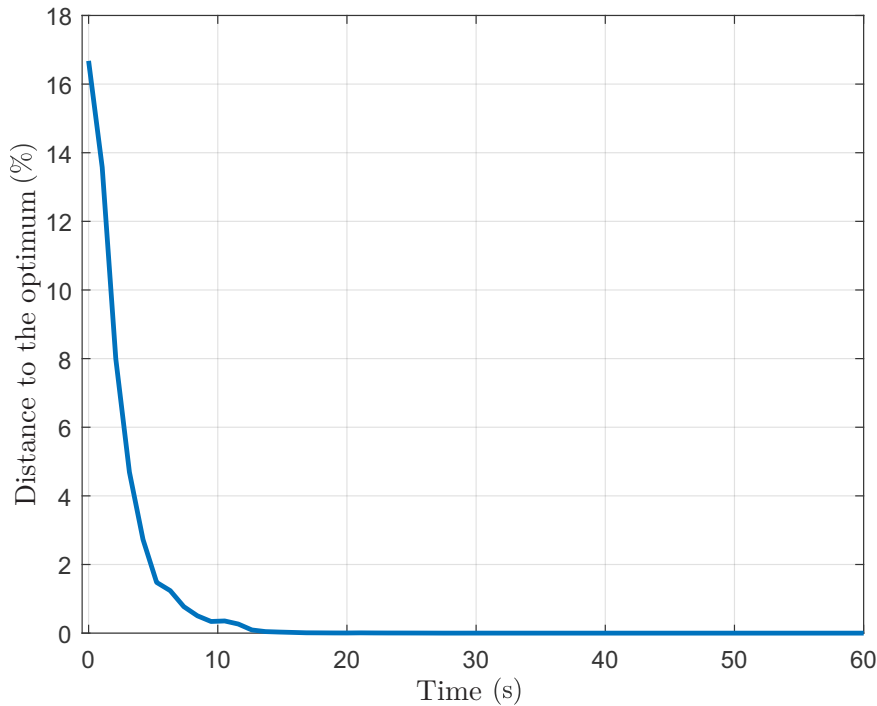
M	T	Cores	f^*	Bilevel		Benders Decomposition	
				f	CPU(s)	f	CPU(s)
20	4	1	0.3504	0.3504	3.46	0.3504	251.57
		2			4.49		211.06
		4			5.20		205.64
		8			4.56		195.03
	6	1	0.7199	0.7199	13.32	0.7211	600
		2			12.33		
		4			10.38		
		8			7.70		
	8	1	4.4881	4.4881	81.60	4.5069	600
		2			59.46		
		4			41.04		
		8			34.25		
40	4	1	1.4774	1.4774	10.86	1.4798	600
		2			9.93		
		4			5.57		
		8			6.57		
	6	1	3.8698	3.8698	133.76	3.9290	600
		2			103.80		
		4			56.20		
		8			47.75		
	8	1	22.4669	22.4669	23.73	22.6710	600
		2			17.10		
		4			8.90		
		8			6.57		
80	4	1	3.9184	3.9184	254.15	3.9667	600
		2			185.87		
		4			143.43		
		8			123.96		
	6	1	6.3187	6.3187	35.80	6.5900	600
		2			21.51		
		4			19.22		
		8			14.73		
	8	1	16.2753	16.2753	82.15	16.9513	600
		2			45.88		
		4			35.48		
		8			26.96		

or workers, is chosen from the set $\{1, 2, 4, 8\}$. The CPU time is shown in order to compare the performance of the proposed approaches and also how much the use of parallel computing offers a reduction in time to find the optimal solution f^* . The time limit for resolving each method was set at 600 s.

Figure 20 provides the percentage distance of the solution to the optimal value as a function of the number of iterations of the Bilevel optimization algorithm. On the other hand, Figure 21 shows the trajectory of the upper and lower bounds generated by the Benders Decomposition, showing the convergence as a function of the number of iterations.

Table 3 presents results and comparisons between the two decomposition techniques, regarding the ability to solve instances with a large number of subsystems. As can be seen, Bilevel optimization is more efficient with respect to both the computational time and the convergence speed to a globally optimal solution.

Figure 20 – Percentage distance of the solution obtained by Bilevel optimization with respect to the global optimum, for the case $M = 40$ and $T = 6$.

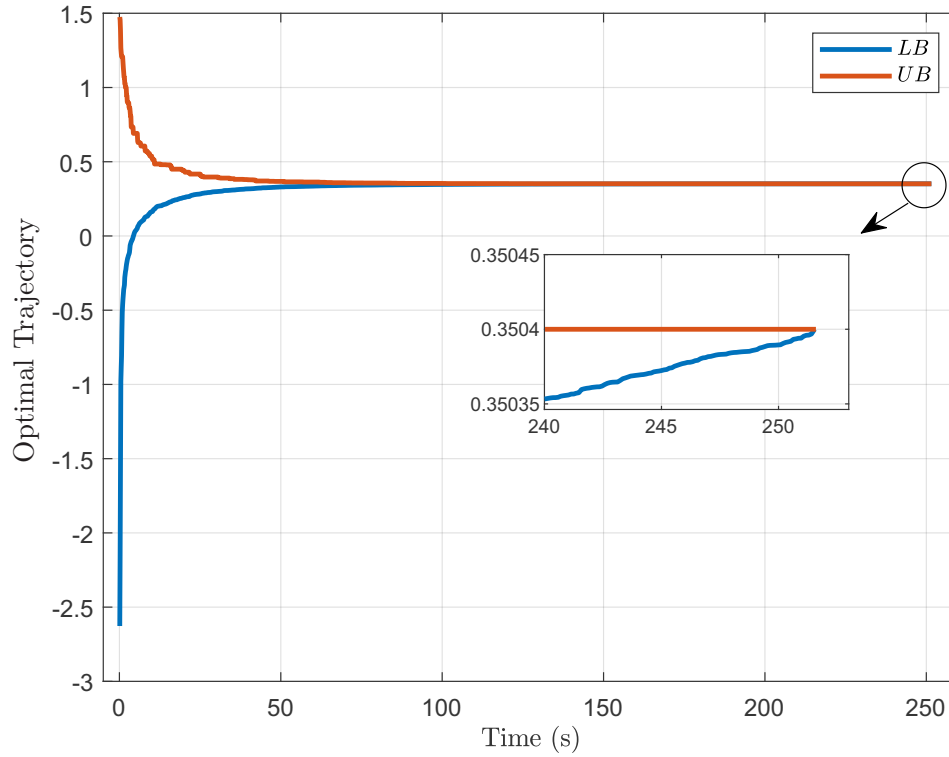


Source: Author (2020)

In addition to the slower convergence, Benders decomposition had difficulties converging to the solution within the defined tolerance and specified timeout, despite the increase in the number of CPU cores. In the instance $M = 20$ and $T = 4$, which was the only instance that Benders decomposition obtained a solution considering the set of problems, it is possible to see an improvement in the solution time as the number of CPU cores increases, as expected.

In the Bilevel optimization, it is possible to observe that in all the problem instances, the global solution was found, and the reductions in computational time were significant with the increase in the number of CPU cores. However, as the number of CPU cores increases, the time

Figure 21 – Trajectory of the lower and upper bounds produced by Benders decomposition, for the case $M = 20$ and $T = 4$.



Source: Author (2020)

to obtain optimal solutions decreases, but not linearly. In the left-hand side of Figure 22, the speed-up ratio is presented, being defined by

$$su_{ratio} = \frac{T(1)}{T(N)}, \quad (4.17)$$

where T is the computational time as a function of the number of cores, and N is the number of CPU cores.

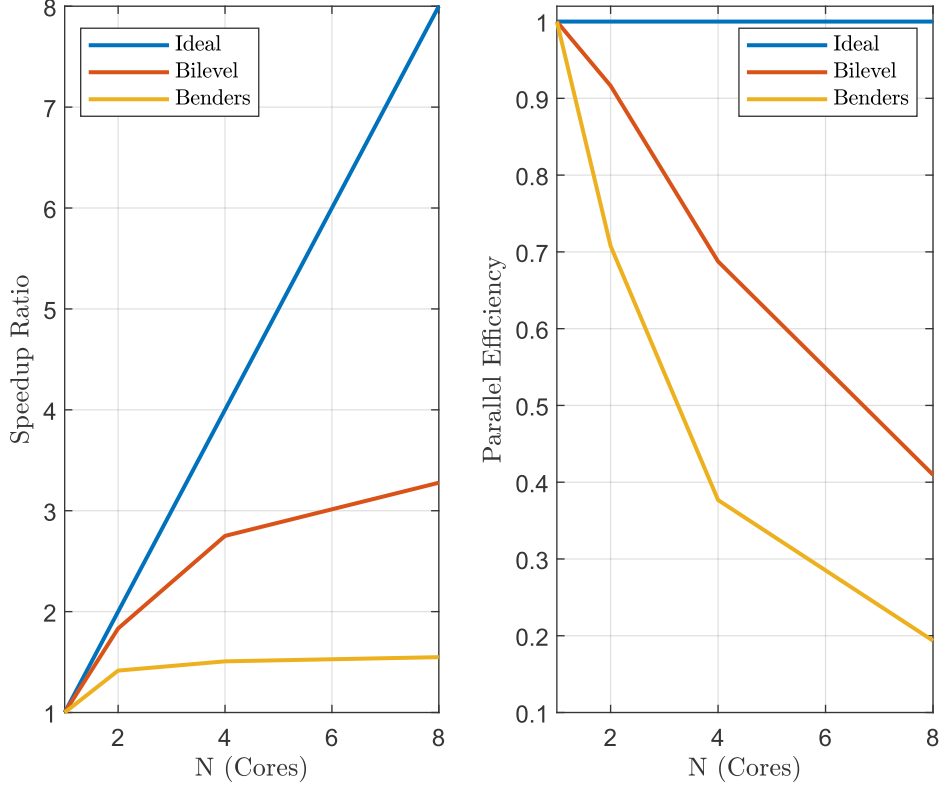
The parallel efficiency that defines the efficiency of the time solution as the cores increases is stated as follows,

$$pe = \left(\frac{T(1)}{T(N)} \right) \frac{1}{N}. \quad (4.18)$$

In the right-hand side of Figure 22, it can be seen that as the number of cores increases, the efficiency decreases. Some reasons related to this behavior are that, with the increase in the time needed to allocate tasks in cores, the gain in parallelizing the problem decreases.

Therefore, it can be observed that the parallelization performance becomes inefficient as the number of cores increases. However, the gain obtained is already relevant for the study of decomposition problems, for the Benders decomposition, the instance that converged, within the determined tolerance, it is possible to see a relevant computational gain.

Figure 22 – Comparison between decomposition techniques according to speed-up (su_{ratio}) and parallel efficiency metrics (pe).



Source: Author (2020)

4.4 BATTERY CHARGING APPLICATION

To illustrate the use of the hierarchical decompositions and to make them less abstract, this section presents an application problem of charging batteries. Consider an electric car charging station, where each car is represented by an independent system (*subproblem*) that reports to the central station (*master*) for the allocation of resources. For each car, there is a state of charge (*SoC*) associated with its battery, which can be defined as in (TUDOROIU et al., 2019):

$$SoC(k+1) = \left(SoC(k) \pm \frac{i(k)\eta}{\Theta} \right), \quad \forall k, \quad (4.19)$$

in which $i(k)$ refers to the current [A] applied at the instant k which, depending on its sign, determines whether the battery is charging or discharging, which also influences the value of η which defines battery charging efficiency; Θ sets the battery charge capacity to [Ah]; and *SoC* varies within the $[0, 100]$ (%) range (from depleted to fully charged).

The MPC problem (4.1) with limited resource constraints can thus be employed in the management of battery charging, using the model given by Eq. (4.19). Problem P , in order to demonstrate a practical application, can be recast as following:

$$\min J = \sum_{m=1}^M \sum_{j=N_1}^{N_2} \|y_m(k+j|k) - w_m(k+j)\|_{Q_m}^2 + \sum_{m=1}^M \sum_{j=0}^{N_u-1} \|\Delta u_m(k+j|k)\|_{W_m}^2 \quad (4.20a)$$

s.t.: For $m \in \mathcal{M}$:

$$SoC_m(k+j+1|k) = SoC_m(k+j|k) + \frac{\eta}{\Theta_m} u_m(k+j|k), \quad j = 0, \dots, N_2 - 1 \quad (4.20b)$$

$$y_m(k+j|k) = SoC_m(k+j|k), \quad j = N_1, \dots, N_2 \quad (4.20c)$$

$$u_m(k+j|k) = \begin{cases} u_m(k+j-1|k) + \Delta u_m(k+j|k), & j = 0, \dots, N_u - 1 \\ u_m(k+j-1|k), & j = N_u, \dots, N_2 - 1 \end{cases} \quad (4.20d)$$

$$SoC_m(k|k) = SoC_m(k) \quad (4.20e)$$

$$u_m(k-1|k) = u_m(k-1) \quad (4.20f)$$

$$y_m^{\min} \leq y_m(k+j|k) \leq y_m^{\max}, \quad j = N_1, \dots, N_2 \quad (4.20g)$$

$$u_m^{\min} \leq u_m(k+j|k) \leq u_m^{\max}, \quad j = 0, \dots, N_u - 1 \quad (4.20h)$$

$$\Delta u_m^{\min} \leq \Delta u_m(k+j|k) \leq \Delta u_m^{\max}, \quad j = 0, \dots, N_u - 1 \quad (4.20i)$$

For $r = 1, \dots, R, j = 0, \dots, N_2 - 1$:

$$\sum_{m=1}^M s_{r,m}^T u_m(k+j|k) \leq s_r^{\max}(k+j), \quad (4.20j)$$

where: SoC_m is the state of charge for each battery in vehicle m ; u_m is the current applied to the battery for charging; and Θ_m is the battery charge capacity of vehicle m given in [Ah]. In this example, we only consider battery charge, so the current signal is non-negative, and therefore the charge efficiency value (η) is constant for all vehicles.

As it can be seen in Eq. (4.20c), the system output y_m is the value of the state SoC_m . Therefore, we consider that the state is observable; otherwise, it would be necessary to apply a state observer or Kalman filter (KALMAN, 1960), for instance.

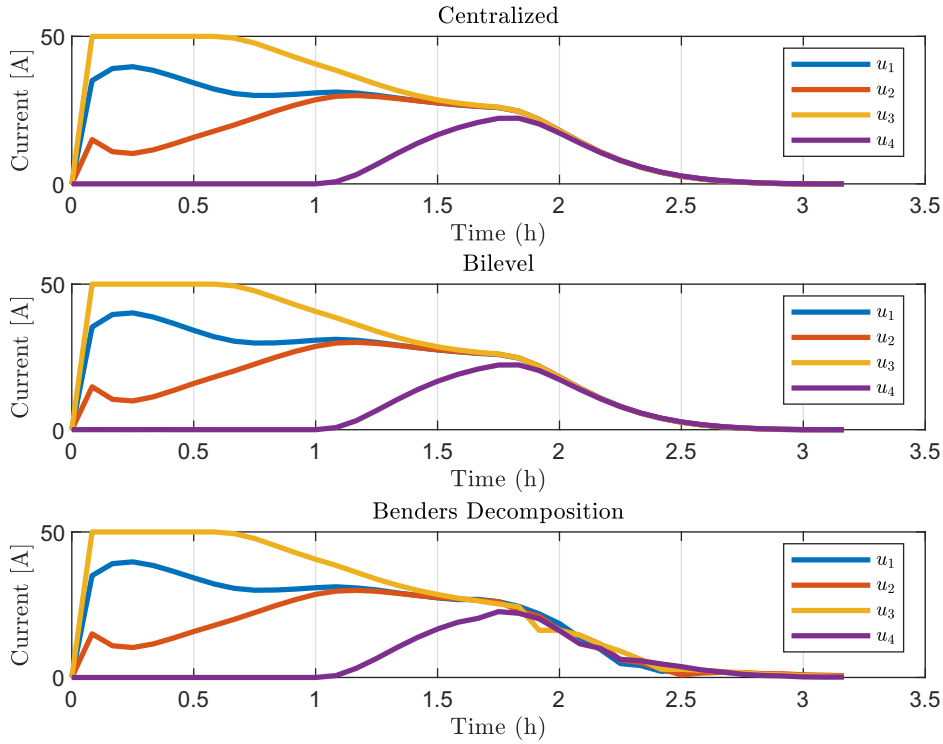
The objective of the problem defined in (4.20) is the charging of batteries and thus, by defining a reference so that all batteries have SoC of 100%, the reference tracking error and the control effort applied to the subsystems are minimized.

4.4.1 Application to an Example Instance

Consider a battery charging problem with the following characteristics: $M = 4$, that is, four vehicles, which have identical batteries with a charging capacity $\Theta_m = 100$ [Ah], and a charging efficiency $\eta = 0.98$. The sampling time is $T_s = 5$ [min], a prediction and control horizon with $T = 12$ samples are considered, the maximum resource is $s^{\max} = 100$ [A] for all instants and cars, and the current injection limit and its variation were defined as $u^{\max} = \Delta u^{\max} = 50$ [A], $u^{\min} = 0$ [A], and $\Delta u^{\min} = -50$ [A].

The results of applying bilevel and Benders decomposition to the battery charging instance are shown in Figure 23. As it can be seen, all subsystems (vehicle batteries) were charged independently, using the distributed formulation, and respecting the maximum available resource and the limits of the defined variables.

Figure 23 – Application of bilevel and Benders decomposition to the battery charging problem.

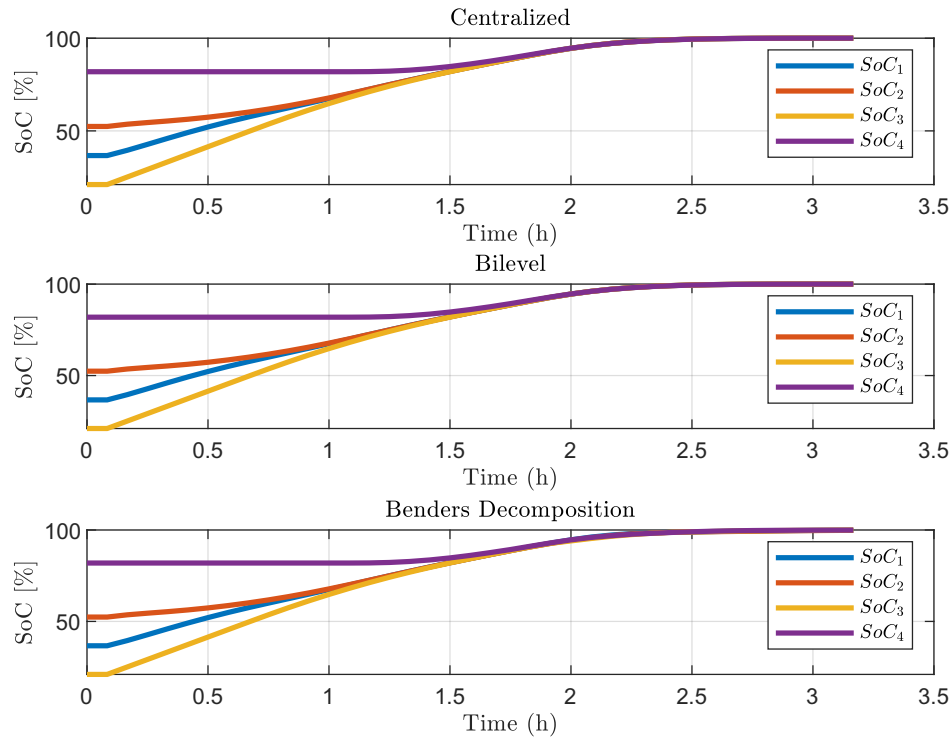


(a) Control signals applied to the subsystems.

The control actions favored the systems with less *SoC*, causing them to recharge first and, only after reaching a higher *SoC*, the control actions direct the available current to vehicles that had a higher *SoC* at the start of charging. The behavior induced by the controller keeps a balance between the resource available and the *SoC* of all vehicles, thereby achieving the primal goal of charging all vehicles fully.

4.5 SUMMARY

In this chapter, the MPC formulation for a class of resource-constrained dynamic systems in its centralized form, the bilevel optimization and Benders decomposition methods applied to the baseline problem, were presented. The problem is reformulated into a hierarchical structure according with both decomposition strategies. Numerical analyses are performed in synthetic instances in order to assess the relative performance of the decompositions, and give insights into the MPC of the considered systems based on the proposed decompositions.



(b) State of charge of the vehicles.

Source: Author (2020)

In a set of test problems, both decompositions were able to obtain nearly optimal and globally optimal solutions, with bilevel achieving a better performance and reaching the optimum faster than the Benders decomposition. A comparison is carried out to assess the benefits of multi-core/distributed computation for the decompositions. It is possible to parallelize the subproblems because of the inherited mathematical structure of bilevel and Benders decomposition.

In this context, in the synthetic problems, parallelization of the decomposition strategies shows that it is possible to reduce the computational time, but not in a linear form. Bilevel optimization was more effective with respect to this feature, because it was the method that solved all the instances to global optimality. Benders decomposition had a poor performance in large problem instances, with several subsystems or higher prediction horizons, but showed that it is possible to converge to the same global optimum of its centralized counterpart.

An application to the charging of electric vehicles is presented using the two decomposition methods. The analysis showed that the electric vehicles can be recharged in a distributed fashion, approaching the optimal behavior that is achieved by a centralized controller.

5 DECOMPOSITIONS FOR MPC OF DYNAMIC SYSTEMS WITH ACTIVATION CONSTRAINTS

In this section, the formulation presented in Chapter 4 for the MPC of resource-constrained dynamic systems is extended by introducing discrete control variables for the activation and deactivation of the control inputs to the subsystems. From the structure of problem (4.1), let $\delta_m(k)$ be binary variables associated with subsystem m at time k , which are used to decide whether or not the subsystem will be influenced by input signals. The MPC problem becomes:

$$P : \min J = \sum_{m=1}^M \sum_{j=N_1}^{N_2} \|y_m(k+j|k) - w_m(k+j)\|_{Q_m}^2 + \sum_{m=1}^M \sum_{j=0}^{N_u-1} \|\Delta u_m(k+j|k)\|_{W_m}^2 \quad (5.1a)$$

while, for $m \in \mathcal{M}$, being subject to:

$$\text{Eqs. (4.1b)-(4.1f)} \quad (5.1b)$$

$$\text{Eqs. (4.1g), (4.1i)} \quad (5.1c)$$

$$\delta_m(k+j|k)u_m^{\min} \leq u_m(k+j|k) \leq \delta_m(k+j|k)u_m^{\max}, \quad j = 0, \dots, N_u - 1 \quad (5.1d)$$

$$\delta_m(k+j|k) \in \{0, 1\}, \quad j = 0, \dots, N_u - 1 \quad (5.1e)$$

and, for $r = 1, \dots, R$, $j = 0, \dots, N_2 - 1$, subject to resource constraint:

$$\sum_{m=1}^M s_{r,m}^T u_m(k+j|k) \leq s_r^{\max}(k+j) \quad (5.1f)$$

in which the constraints remain closely related to problem (4.1), with modification of constraint (4.1h) which becomes (5.1d), i.e., adding variables $\delta_m(k+j|k) \in \{0, 1\}$ to switch on and off the control signals of subsystem m .

5.1 BENDERS DECOMPOSITION AND OUTER APPROXIMATION

Here we begin by introducing the subproblems, like in Chapter 4, but with binary variables for handling the switching of input signals. The Optimality and the Feasibility subproblems are presented in a form that is suitable for use in Benders Decomposition and, the Optimality subproblem for Outer Approximation, as stated in Section 2.4.

5.1.1 Optimality and Feasibility Subproblems

For a feasible resource allocation ss and binary vector β with activation decisions, the optimality subproblem is given by

$$O(ss, \beta) : o(ss, \beta) = \min_z \sum_{m=1}^M f_m(z_m) \quad (5.2a)$$

s.t. : for $m = 1, \dots, M$:

$$\begin{cases} h_m(z_m) = 0 \\ g_m(z_m) \leq 0 \\ P_m z_m + Q_m \beta_m \leq 0 \\ R_m z_m - S_m ss_m \leq 0 \end{cases} \quad (5.2b)$$

where: (i) the same notation of problem (4.2a) is followed; (ii) $\beta = (\beta_1, \dots, \beta_m)$ and $\beta_m = (\delta_m(k+j) : \forall j \in \mathcal{N}_u, \delta_m \in \{0, 1\})$; (iii) P_m and Q_m are suitable matrices that define constraint (5.1d), and R_m and S_m represent (4.5e); and (iv) in the particular problem of concern, S_m is the identity matrix and $R_m z_m$ is effectively $\tilde{R}_m u_m$ since only the terms $s_{r,m}^T u_m(k+j|k)$ are needed in the constraint. Thus, the vector functions h_m and g_m represent the equalities (5.1b) and inequalities (5.1c), respectively.

Notice that $o(ss, \beta)$ induces an upper bound for a feasible (ss, β) , meaning one vector ss that satisfies constraints (4.3b)-(4.3c) and one vector β that satisfies (5.1d), which renders $O(ss, \beta)$ feasible. Clearly $o(ss, \beta)$ can be computed in parallel as follows

$$o(ss, \beta) = \sum_{m=1}^M o_m(ss_m, \beta_m) \quad (5.3)$$

for which

$$O_m(ss_m, \beta_m) : o_m(ss_m, \beta_m) = \min_{z_m} f_m(z_m) \quad (5.4a)$$

$$\text{s.t. : } h_m(z_m) = 0 \quad (5.4b)$$

$$g_m(z_m) \leq 0 \quad (5.4c)$$

$$P_m z_m + Q_m \beta_m \leq 0 \quad (5.4d)$$

$$R_m z_m - S_m ss_m \leq 0. \quad (5.4e)$$

For an infeasible resource allocation and combination of binary variables, the following feasibility subproblem is to be solved:

$$F(ss, \beta) : f(ss, \beta) = \min_{\gamma \geq 0, z} \gamma \quad (5.5a)$$

s.t. : for $m = 1, \dots, M$:

$$\left\{ \begin{array}{l} h_m(z_m) = 0 \\ g_m(z_m) \leq \gamma \cdot e_m \\ P_m z_m + Q_m \beta_m \leq \gamma \cdot e_m \\ R_m z_m - S_m ss_m \leq \gamma \cdot e_m, \end{array} \right. \quad (5.5b)$$

with $e_m = (1, 1, \dots, 1)$ being a vector of suitable dimension and γ a nonnegative scalar. The optimal γ can be obtained by solving an auxiliary subproblem only for the infeasible BO_m . Let $\mathcal{M}_{infeas} = \{m \in \mathcal{M} : O_m(ss_m, \beta_m) \text{ is infeasible}\}$. Then $F(ss, \beta)$ is solved as follows:

$$f(ss, \beta) = \max\{f_m(ss_m, \beta_m) : m \in \mathcal{M}_{infeas}\} \quad (5.6)$$

and, for all $m \in \mathcal{M}_{infeas}$, we solve the problem:

$$F_m(ss_m, \beta_m) : f_m(ss_m, \beta_m) = \min_{\gamma_m \geq 0, z_m} \gamma_m \quad (5.7a)$$

$$\text{s.t.: } h_m(z_m) = 0 \quad (5.7b)$$

$$g_m(z_m) \leq \gamma_m \cdot e_m \quad (5.7c)$$

$$P_m z_m + Q_m \beta_m \leq \gamma_m \cdot e_m \quad (5.7d)$$

$$R_m z_m - S_m ss_m \leq \gamma_m \cdot e_m \quad (5.7e)$$

In fact if (ss_m, β_m) is feasible for O_m , then the corresponding F_m will have an optimal value $f_m(ss_m, \beta_m) = 0$ and the feasibility subproblem does not have to be solved.

5.1.2 Master Problem of the Benders Decomposition

At iteration p of the Benders algorithm, let $ss^{(p)}$ be the resource allocation vector, $\beta^{(p)}$ the vector of binary variables, and assume that $O(ss^{(p)}, \beta^{(p)})$ is feasible. Let $\mu_m^{(p)}$ and $\nu_m^{(p)}$ be the Lagrange multipliers associated with the resource constraint (5.4e) and the activation constraint (5.4d) respectively, at the optimal solution $z_m^{(p)}$. Then the following optimality cut can be introduced to the Benders master problem:

$$\alpha_B \geq \sum_{m \in \mathcal{M}} f_m(z_m^{(p)}) + \sum_{m \in \mathcal{M}} \mu_m^{(p)T} [R_m z_m^{(p)} - S_m ss_m] + \sum_{m \in \mathcal{M}} \nu_m^{(p)T} [P_m z_m^{(p)} + Q_m \beta_m^{(p)}], \quad (5.8)$$

with α_B being the lower bound of the overall objective of P given by Eq. (5.1). Due to the complementarity conditions, the local constraints (5.4b) and (5.4c) do not play a part in the cutting plane. The decision space of the Benders master problem consists of the vector (ss, β) , with resource allocation and binary decisions, and the lower bound α_B .

However, if $(ss^{(p)}, \beta^{(p)})$ is an infeasible vector at iteration p , then let $\mathcal{M}^{(p)} \subset \mathcal{M}$ be the subset of subproblems m such that $\gamma_m = f(ss, \beta)$. Solving $F(ss, \beta)$ renders the following feasibility cut:

$$\sum_{m \in \mathcal{M}^{(p)}} \left\{ \mu_{m,c}^{(p)T} g_m(z_m) + \mu_{m,e}^{(p)T} [R_m z_m^{(p)} - S_m ss_m] + \nu_{m,d}^{(p)T} [P_m z_m^{(p)} + Q_m \beta_m^{(p)}] \right\} \leq 0 \quad (5.9)$$

At iteration p , an optimality cut is obtained by successfully solving $O(ss^{(p)}, \beta^{(p)})$, or else a feasibility cut by solving $F(ss^{(p)}, \beta^{(p)})$. Let $\mathcal{O}^{(p)}$ and $\mathcal{F}^{(p)}$ be the indices of the iterations for which an optimality and feasibility cut was produced respectively. Then, the Benders master problem at iteration p can be stated as follows:

$$BM(p) : \min \alpha_B \quad (5.10a)$$

$$\text{s.t.: } Rss \leq s \quad (5.10b)$$

$$\alpha_B \geq \sum_{m \in \mathcal{M}} \left\{ f_m(z_m^{(i)}) + \mu_m^{(i)T} [R_m z_m^{(i)} - S_m ss_m] + \nu_m^{(i)T} [P_m z_m^{(i)} + Q_m \beta_m] \right\}, \quad i \in \mathcal{O}^{(p)} \quad (5.10c)$$

$$\sum_{m \in \mathcal{M}^{(i)}} \left\{ \mu_{m,c}^{(i)T} g_m(z_m) + \mu_{m,e}^{(i)T} [R_m z_m^{(i)} - S_m ss_m] + \nu_{m,d}^{(i)T} [P_m z_m^{(i)} + Q_m \beta_m] \right\} \leq 0, \quad i \in \mathcal{F}^{(p)} \quad (5.10d)$$

$$\alpha_B \in \mathbf{R}, \quad ss \geq 0, \quad \beta \in \{0, 1\}^{MN_u}. \quad (5.10e)$$

5.1.3 Outer Approximation Master Problem

The problem of concern (5.1) is a MIQP with constraints being all affine and only the objective being nonlinear convex. By applying the OA algorithm, the MPC problem can be solved with a MILP algorithm (for the master) and a set of QPs (for the subproblems). To put in another way, the constraint appearing in the master are the same constraints of the baseline problem (5.1), whereas the objective will be iteratively approximated with linear under estimators. This means that the OA algorithm will not produce infeasible iterates for the subproblems, and only optimality cuts will be generated.

At iteration p , let $ss^{(p)}$ be the resource allocation vector, $\beta^{(p)}$ the vector with binary variables, and assume that $O(ss^{(p)}, \beta^{(p)})$ is feasible. Then using the optimal solution of optimality subproblem, $z^{(p)} = (z_1^{(p)}, \dots, z_M^{(p)})$ it is possible to create the linear approximation for the convex objective f ,

$$\begin{cases} \alpha_{OA} & \geq f(z^{(p)}) + \nabla f(z^{(p)})(z - z^{(p)}) \\ & = \sum_{m \in \mathcal{M}} \left\{ f_m(z_m^{(p)}) + \nabla f_m(z_m^{(p)})(z_m - z_m^{(p)}) \right\}. \end{cases} \quad (5.11)$$

in which ∇f_m is the gradient for the objective function of subsystem m . Since the equality $h(\cdot)$ and inequality $g(\cdot)$ constraints are all affine, there is no need to linearize the corresponding vector functions in the master problem.

Then, OA master problem at iteration p can be stated as follows:

$$OAM(p) : \min \alpha_{OA} \quad (5.12a)$$

$$\text{s.t.: } Rss \leq s \quad (5.12b)$$

$$\alpha_{OA} \geq \sum_{m \in \mathcal{M}} \{f_m(z_m^{(i)}) + \nabla f_m(z_m^{(i)})(z_m - z_m^{(i)})\}, i = 1, \dots, p \quad (5.12c)$$

$$\text{for } m = 1, \dots, M :$$

$$\left\{ \begin{array}{l} h_m(z_m) = 0 \\ g_m(z_m) \leq 0 \\ P_m z_m + Q_m \beta_m \leq 0 \\ R_m z_m - S_m s s_m \leq 0 \end{array} \right. \quad (5.12d)$$

$$\alpha_{OA} \in \mathbf{R}, ss \geq 0, \beta \in \{0, 1\}^{MN_u}. \quad (5.12e)$$

5.2 NUMERICAL EXPERIMENTS

In order to validate the hierarchical approaches, now with new features as the addition of activation/deactivation of the control signals, using binary variables, a set of synthetic problems were obtained at random. The data and procedures followed to generate these problems are detailed in Table 5.

Table 4 reports the size of the proposed problems. For a given number of subsystems and a stipulated horizon, the increase in the number of variables in the OA master problem and the subproblems is significant, compared to Benders decomposition. The number of variables is all added together in each column, and the column Binary vars contains the number of $\{0, 1\}$ variables in each problem.

As before, the Julia language was used to implement the algorithm together with modeling language for mathematical programming, JuMP, for the reasons already discussed in the previous chapter. The numerical experiments were performed in computer with an Intel® Xeon™ CPU E5-2665 (2.40GHz and 20MB of cache), with 8 cores and 16 threads, 40 GB of RAM, and in a Linux environment.

The Gurobi solver, version 9.0.1, was used with a tolerance of 10^{-6} for the master and subproblems of both decomposition strategies, Benders and Outer Approximation. The relative gap between LB and UB was set to $10^{-2} \%$ as defined below:

$$REL_{gap} = \frac{UB - LB}{(UB + 1e^{-20})} \cdot 100 \quad (\%) \quad (5.13)$$

According to Grossmann (2002), the Outer Approximation method yields a tighter lower bound than the one produced by the master problem of Benders Decomposition. However, in

Table 4 – Number of variables of synthetic problems with binary variables, for analysis of decomposition strategies.

M	T	Outer Approximation			Benders Decomposition		
		MP	Binary vars	SP	MP	Binary vars	SP
4	4	109	12	96	13	12	96
	6	173	20	152	21	20	152
	8	237	28	208	29	28	208
	10	301	36	264	37	36	264
8	4	217	24	192	25	24	192
	6	345	40	304	41	40	304
	8	473	56	416	57	56	416
	10	601	72	528	73	72	528
10	4	271	30	240	31	30	240
	6	431	50	380	51	50	380
	8	591	70	520	71	70	520
	10	751	90	660	91	90	660
20	4	541	60	480	61	60	480
	6	861	100	760	101	100	760
	8	1181	140	1040	141	140	1040
	10	1501	180	1320	181	180	1320

Table 5 – Problem data related to Table 6

M	T	seed	y_{min}/y_{max}	u_{min}/u_{max}	$\Delta u_{min}/\Delta u_{max}$	r_{max}
4	{4, 6, 8, 10}	2	{-50, 50}	{1, 3}	{-3, 3}	2
8	{4, 6, 8, 10}	2	{-50, 50}	{1, 3}	{-3, 3}	2
10	{4, 6, 8, 10}	2	{-50, 50}	{1, 3}	{-3, 3}	2
20	{4, 6, 8, 10}	2	{-50, 50}	{1, 3}	{-3, 3}	2

our experiments, the tighter bounding procedures did not slow down the OA algorithm, which actually converged faster than the Benders decomposition.

For problems that the algorithms did not reach the tolerance defined by the relative gap, the Outer Approximation method was closer to the tolerance than the Benders decomposition when the CPU limit was reached, after 3600s of computation – this confirms the assumption that the OA algorithm produces tighter bounds in comparison to the Benders decomposition.

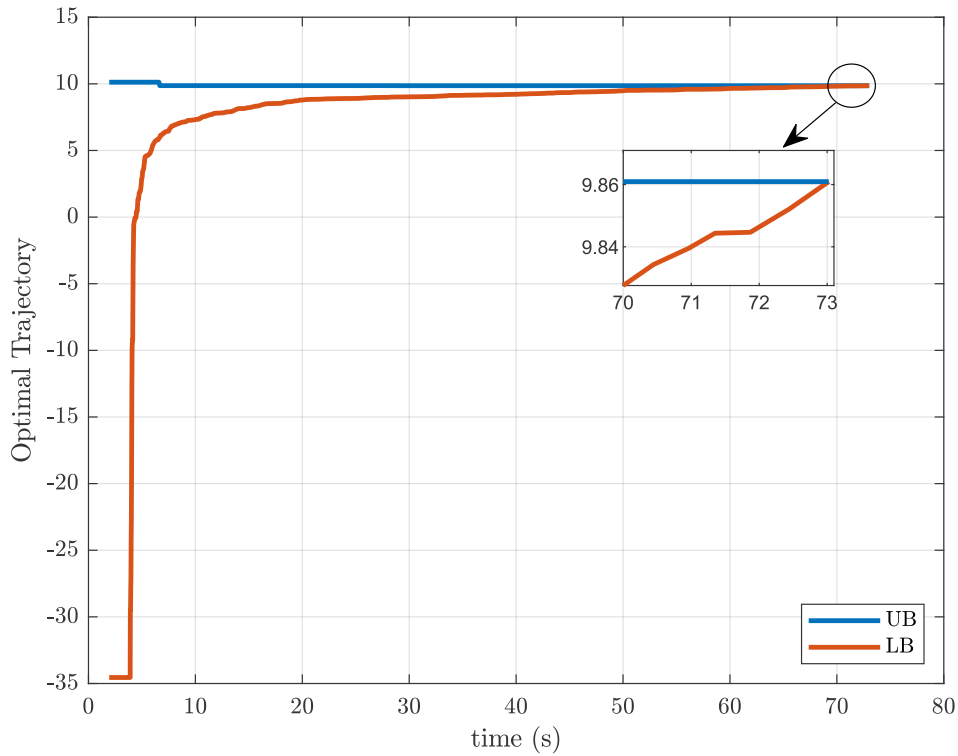
If one compares the size of the problem, the number of binary variables in Table 4 and the solution times in Table 6, it can be noticed that the number of binary variables makes the solution of the problems more challenging.

Figure 24 and Figure 25 show the optimal trajectory of the upper bound UB and lower bound LB generated by decomposition strategies, when they are applied to the MPC problem

Table 6 – Computational analysis of the Outer Approximation and Benders decomposition.

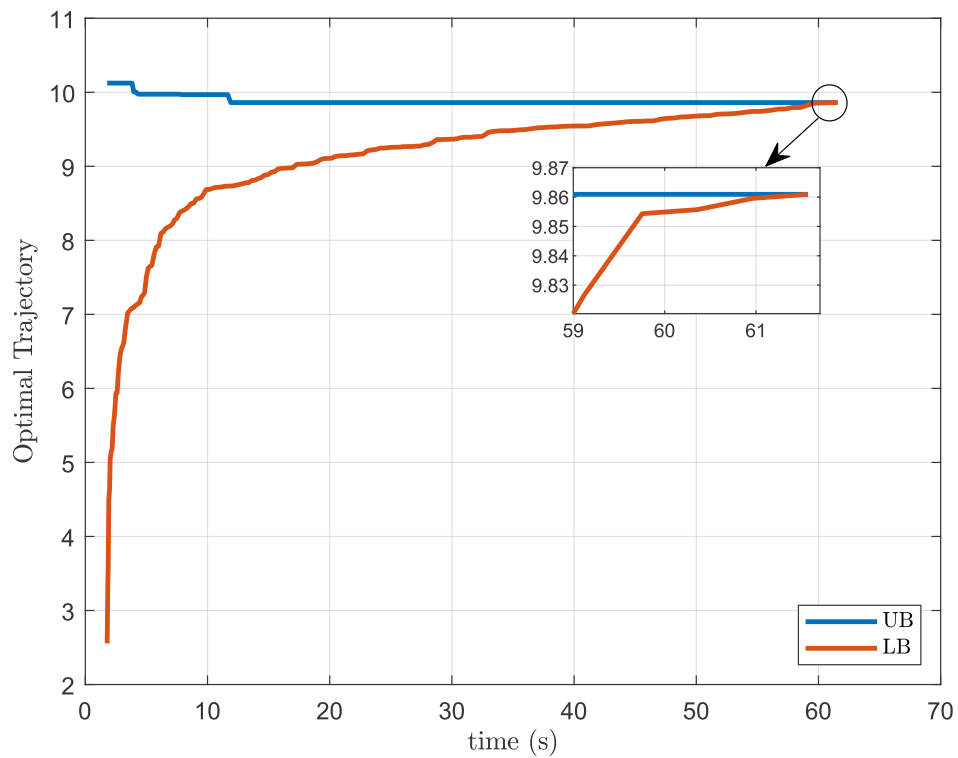
M	T	f^*	Outer Approximation			Benders Decomposition		
			f	CPU(s)	GAP(%)	f	CPU(s)	GAP(%)
4	4	1.3911	1.3911	2.0634	0	1.3911	3.4560	0
	6	3.7478	3.7478	5.1494	0	3.7478	10.3064	0
	8	11.1778	11.1778	38.6868	0	11.1778	95.8767	0
	10	34.5252	34.5222	1331.4079	0.0086	34.5229	2618.3762	0.0066
8	4	3.6063	3.6063	3.0238	0	3.6063	4.7467	0
	6	9.8609	9.8609	61.5691	0	9.8609	73.0234	0
	8	28.2525	28.2525	3600	2.5148	28.2525	3600	1.0521
	10	80.9463	81.0472	3600	4.1118	81.0075	3600	5.6016
10	4	7.8773	7.8773	54.7207	0	7.8773	226.4206	0
	6	27.8807	27.9743	3600	5.9458	28.1876	3600	17.8776
	8	96.3721	97.1651	3600	7.4725	96.4674	3600	14.3194
	10	328.5944	331.1783	3600	4.5508	330.1090	3600	7.7865
20	4	15.0514	15.0514	577.7808	0	15.0514	3388.1925	0
	6	85.4176	85.8142	3600	4.3164	85.6710	3600	9.8120
	8	554.2668	556.8338	3600	2.0377	556.5846	3600	4.1113
	10	3635.5532	3639.0866	3600	0.5187	3638.8462	3600	1.1087

Figure 24 – Trajectory of the bounds in Benders decomposition solution, for problem $M = 8$ and $T = 6$.



Source: Author (2020)

Figure 25 – Trajectory of the bounds in Outer Approximation solution, for problem $M = 8$ and $T = 6$.



Source: Author (2020)

with resource and activation constraints, for the case with $M = 8$ subsystems and prediction and control horizon of length $T = 6$. Again it is possible to see that the bounds of the Outer Approximation are much closer and converge in a significantly faster pace than the Benders decomposition.

Finally, the results reported in the experiments show that both decomposition strategies can be effective. They allowed the baseline MPC problem to be decomposed into a set of subproblems, whereby the master handles the binary variables (for the activation of control signals) and resource allocations, whereas the subproblems consist of small quadratic programs. The decompositions enabled a distributed solution of the MPC problem that can achieve nearly optimal solutions for a given small tolerance.

5.3 BATTERIES CHARGING APPLICATION WITH ACTIVATION CONSTRAINTS

In order to illustrate the use of the presented decompositions and the modification on the activation constraints, this session presents another example problem of charging batteries. Considering the MPC problem (5.1) with limited resource constraints and activation constraints can thus be employed in the management of battery charging, using the model given by Eq. (4.19). Problem P , in order to demonstrate a practical application, can be recast like the following:

$$\min J = \sum_{m=1}^M \sum_{j=N_1}^{N_2} \|y_m(k+j|k) - w_m(k+j)\|_{Q_m}^2 + \sum_{m=1}^M \sum_{j=0}^{N_u-1} \|\Delta u_m(k+j|k)\|_{W_m}^2 \quad (5.14a)$$

s.t. : for $m \in \mathcal{M}$:

$$\left\{ \begin{array}{l} SoC_m(k+j+1|k) = SoC_m(k+j|k) + \frac{\eta}{\Theta_m} u_m(k+j|k), j = 0, \dots, N_2 - 1 \\ y_m(k+j|k) = SoC_m(k+j|k), j = N_1, \dots, N_2 \\ u_m(k+j|k) = \begin{cases} u_m(k+j-1|k) + \Delta u_m(k+j|k), j = 0, \dots, N_u - 1 \\ u_m(k+j-1|k), j = N_u, \dots, N_2 - 1 \end{cases} \\ SoC_m(k|k) = SoC_m(k) \\ u_m(k-1|k) = u_m(k-1) \\ y_m^{\min} \leq y_m(k+j|k) \leq y_m^{\max}, j = N_1, \dots, N_2 \\ \delta_m(k+j|k) u_m^{\min} \leq u_m(k+j|k) \leq \delta_m(k+j|k) u_m^{\max}, j = 0, \dots, N_u - 1 \\ \Delta u_m^{\min} \leq \Delta u_m(k+j|k) \leq \Delta u_m^{\max}, j = 0, \dots, N_u - 1 \end{array} \right. \quad (5.14b)$$

For $r = 1, \dots, R, j = 0, \dots, N_2 - 1$:

$$\left\{ \begin{array}{l} \sum_{m=1}^M s_{r,m}^T u_m(k+j|k) \leq s_r^{\max}(k+j), \end{array} \right. \quad (5.14c)$$

where: SoC_m is the state of charge for the battery of vehicle m ; u_m is the current applied to the battery for charging; Θ_m is the battery charge capacity of vehicle m given in [Ah]; and

$\delta_m(k + j|k) \in \{0, 1\}$. In this context, we only consider battery charge, so the current signal is non-negative and therefore the charge efficiency value (η) is constant for all vehicles.

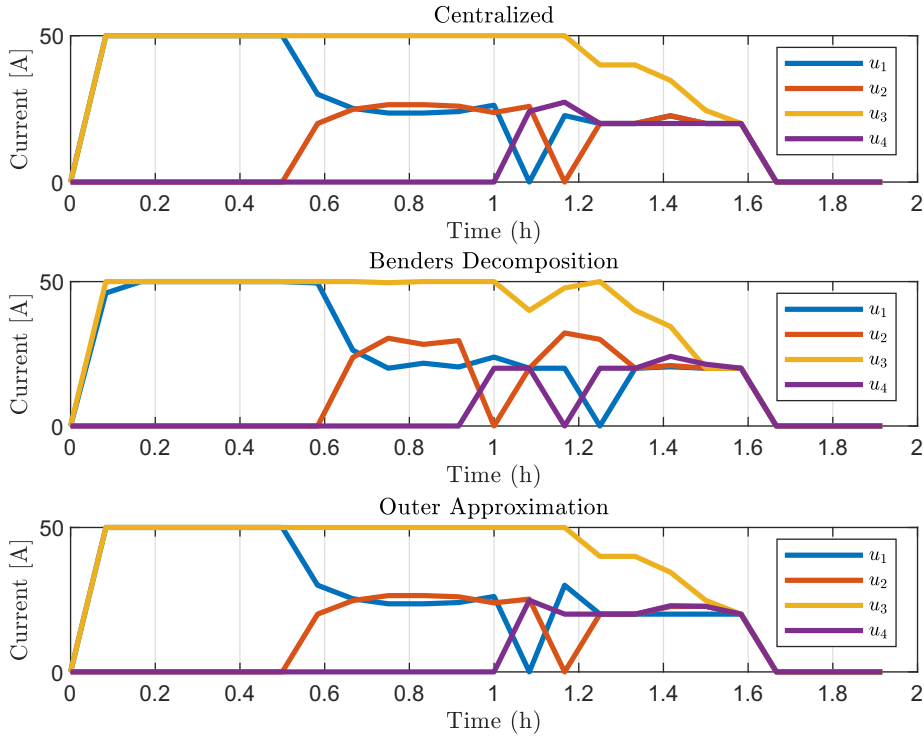
As it can be seen in Eq. (5.14b), the system output y_m is the value of the state SoC_m . Again, we assume that the state is observable; otherwise, it would be necessary to apply a state observer, or Kalman filter (KALMAN, 1960).

5.3.1 Application to a Sample Instance

Consider a battery charging problem with the following characteristics, $M = 4$, that is, four vehicles, which have identical batteries with a charging capacity $\Theta_m = 100 [Ah]$, a charging efficiency $\eta = 0.98$. The sampling time chosen was $T_s = 5 [\text{min}]$, a prediction and control horizon of $T = 5$ samples were considered, the maximum resource was $s^{\max} = 100 [A]$ for all instants and vehicles, and the current injection limit and its variation were defined as $u^{\max} = \Delta u^{\max} = 50 [A]$, $u^{\min} = 20 [A]$, and $\Delta u^{\min} = -50 [A]$.

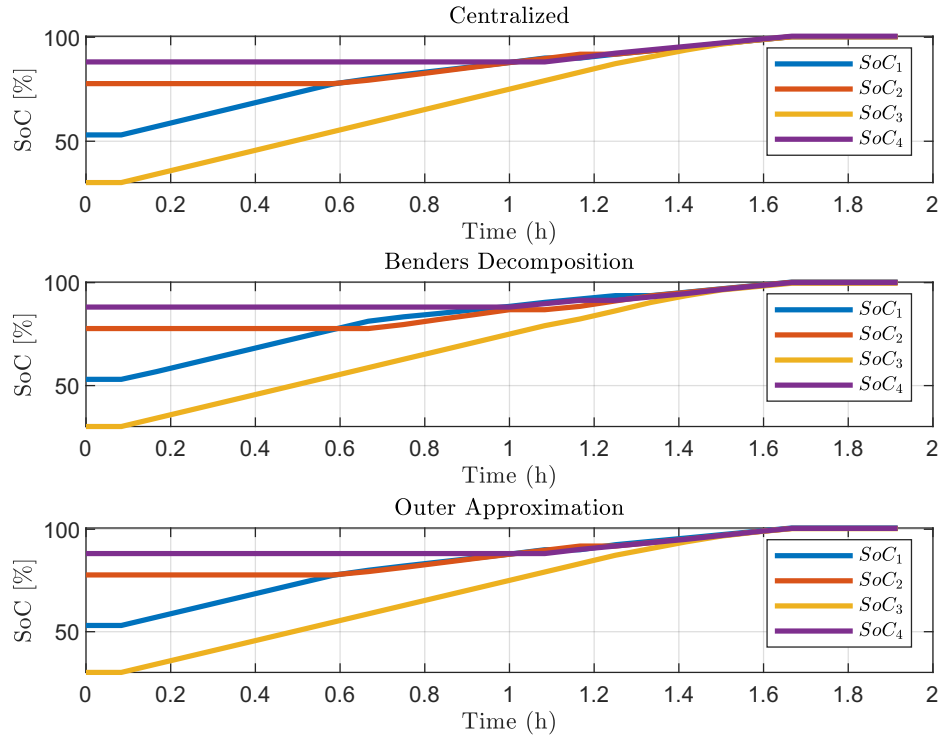
The results of applying Benders Decomposition and Outer Approximation to the battery charging instance are shown in Figure 26.

Figure 26 – Application of Benders decomposition and Outer Approximation to the battery charging problem, with activation constraints.



(a) Control signals applied to the systems.

The plots in Figure 26 show that all subsystems (vehicle batteries) were charged independently, using the distributed formulation, and respecting the maximum available resource and



(b) State of charge of the vehicles

Source: Author (2020)

the limits imposed on the defined variables.

Again, the control actions favored the systems with less SoC , causing them to recharge first. Only then, the control actions are directly available to vehicles with a higher SoC at the start of charging. The behavior induced by the controller keeps a balance between the resource available and the SoC of all vehicles, thereby achieving the primal goal of charging all vehicles fully.

5.4 SUMMARY

In this chapter the MPC of resource constraint formulation with the addition of activation constraints on the control variables, in its centralized form, then Outer Approximation and Benders Decomposition methods applied to the baseline problem were presented. The problem is reformulated into a hierarchical structure according with both decomposition strategies. Numerical analyses were performed, in synthetic instances, in order to give exemplification and insights on possibilities of solutions to these problems in a distributed fashion.

In a set of test problems, both decompositions were able to obtain nearly optimal and globally optimal solutions, with Outer Approximation having a better performance and achieving the global optimal faster than Benders Decomposition for the synthetic problems proposed.

Benders decomposition had a lower performance with problems with more subsystems

or higher horizons compared to Outer Approximation but showed that its possible to converge to the same global optimal of its centralized counterpart.

An application to charging electric vehicles, taking use of activation constraints, is presented using this decompositions methods, and is possible to see that was possible to charge to vehicles in a distributed fashion, approaching the optimal behavior that is achieved by a centralized controller.

6 FINAL REMARKS

6.1 CONCLUSION

This dissertation addresses the use of hierarchical decomposition on the problem of MPC of resource constraints, including the use of activation constraints that arises from the interconnection of dynamic systems that share limited resources. Different decompositions based solutions are presented, taking into account the characteristic of this class of system.

Control methods such as MPC, which is widely addressed in the literature, were presented to give a glance on model-based control strategies. Among its numerous characteristics, the use of linear models in state space and optimization methods to obtain optimal control laws based on these models and measurements of the system.

So taking advantage of the MPC structure, the formulation proposed in this work, handles linear dynamic systems that share limited resources, a system that has a resource constraint between its subsystems that couples all of their controls. Latter is discussed that decompositions strategies were realized in order to reformulate the original problem into a distributed structure.

Such a distributed structure was harnessed by hierarchical approaches that split the computations between a master problem, responsible for the coordination of resources, and a set of subproblems that can be solved concurrently or in parallel. To that end, hierarchical approaches were derived from Bilevel optimization, Benders decomposition, and Outer Approximation in order to take advantage of the system structure and the mathematical structure that arises on the decomposition of the problem, with this was possible to decoupling the subsystems and still been able to manage to achieve the optimal global solution.

Computational problems were created in order to validate the decomposition strategies used. First, only considering continuous problems, Bilevel optimization had a better overall performance. It was the easiest to implement in a programming point of view, and the derivatives/sensitivities were the only communication needed between the master and subproblems.

Benders were more challenging to implement because of the need for the implementation of two kinds of subproblems, then the communications with the master were also relatively simple, with only the necessity of passing the coefficients of the cuts. Its performance was the least efficient, reaching the maximum CPU limit without reaching the tolerance for the primal-dual gap, and often not reaching the optimum. A possibility to reduce computational time would be to solve the dual of the master, which could be warm started since cuts render the incumbent dual solution sub-optimal rather than primal infeasible.

An extension of the original problem was proposed, with the inclusion of discrete or integer variables in the control action. Here Outer Approximation and Benders decomposition

took place in order to validate the hierarchical approach to the problem for being strategies that initially handle MINLP. Outer Approximation has a master problem with more constraints and variables than Benders decomposition, but it was shown to have better performance regarding the lower bound. This tight lower bound could manage to provide faster computation and a better primal-dual gap when not achieving the optimal solution in the CPU limit time.

Practical examples of charging vehicles were presented to give a less abstract view of using these hierarchical decomposition strategies, with only continuous variables and then with activation constraints. Using these methods it is possible to achieve good results, Bilevel had the best approximation of the centralized case, and Benders, even with some minor differences on the control action, could achieve the goal of fully charging vehicles.

For the second case with activation constraints, Outer Approximation renders results that are close to the centralized formulation, and again, Benders, with some differences in the control action, could achieve the goal of fully charging vehicles.

It is possible to conclude that the benefit of the hierarchical decomposition is rather organizational than outperforming its centralized counterpart, as these decompositions enable the control system to be reconfigured locally and expanded with reduced coordination. Notice that the signals communicated between the master problem and subproblems are relatively simple, consisting of resource allocations (from master to subsystems), and sensitivities, cuts, (from subsystems to master). The master problem does not need to have detailed information about the dynamics of the subproblems.

All code of this work is available on GitHub repository: <<https://github.com/pedrohvbsilva/HDS-MPC-RC>>, since the problems generation through the simulation of MPC of charging electric vehicles, so that it is possible a better reproduction of the work presented in this dissertation.

6.2 FUTURE WORK PROPOSAL

- **Use of Multi-cuts**

The generation of multiple Benders cuts as proposed by You and Grossmann (2013), where it has one cut for each subsystem can bring significant savings in CPU time can be achieved by using multi-cut algorithms.

- **Regularized Benders Decomposition**

For a large number of subsystems, it was observed that Benders decomposition presents a low performance in terms of convergence to a global optimum. This behavior motivates a regularization strategy that can assist in the convergence speed of Benders decomposition.

A variation of the regularization method is known as Level Regularization (LR) (ACKOOIJ; OLIVEIRA; SONG, 2019; COLONETTI; FINARDI, 2020). Operating at the level of master problem, the principle that guides the operation of LR is relatively simple: choose a candidate solution from a set of levels in order to minimize the distance between the candidate solution and a reference solution, while imposing a maximum value for the objective according to the current level.

- **Application of Lazy-Constraints**

Some optimization solvers allow users to interact with them during the solution process by providing user-defined functions which are triggered under certain conditions. The generic term for these functions is callback. In integer optimization, the main callback types are lazy constraints, user cuts and heuristic solutions.

Some callbacks define a new constraint which is only activated when necessary, i.e. when a current solution does not respect them. It can avoid building an optimization model with too many constraints up-front. This new constraint is called Lazy-Constraints, this interactions direct trough the solver can perform better than the original cuts provided by Benders decomposition or Outer Approximation.

6.3 CONTRIBUTIONS

As scientific contributions, this research led to the development of two scientific papers, one of which was presented in *Simpósio Brasileiro de Automação Inteligente - SBAI*, that took place in Ouro Preto, Minas Gerais, Brazil in 2019:

- SILVA, P. H. V. B., Brinhosa, A. F., Camponogara, E., & Seman, L. O. A Computational Analysis of a Bilevel Decomposition for MPC of Resource Constrained Dynamic Systems. 2019. 143–147. <<https://doi.org/10.17648/sbai-2019-111120>>

And the second article was accepted for publication on IEEE Latin America Transactions:

- SILVA, P. H. V. B., Seman, L. O., Camponogara, E. A Computational Analysis Of Decomposition Strategies For Model Predictive Control Of Resource-Constrained Dynamic Systems. 2020. IEEE-LAT.

BIBLIOGRAPHY

ACKOOIJ, W. V.; OLIVEIRA, W. de; SONG, Y. On level regularization with normal solutions in decomposition methods for multistage stochastic programming problems. *Computational Optimization and Applications*, Springer, v. 74, n. 1, p. 1–42, 2019.

BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. *Nonlinear Programming*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2006. Available from Internet: <<http://doi.wiley.com/10.1002/0471787779>>.

BENDERS, J. F. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, v. 4, n. 1, p. 238–252, dec 1962. Available from Internet: <<http://link.springer.com/10.1007/BF01386316>>.

BEZANSON, J. et al. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*, v. 59, n. 1, p. 65–98, jan 2017. Available from Internet: <<https://epubs.siam.org/doi/10.1137/141000671>>.

BOYD, S.; VANDENBERGHE, L. *Convex Optimization*. 10. ed. Cambridge University Press, 2012. 715 p. Available from Internet: <[https://web.stanford.edu/{~}boyd/cvxbook/bv{_{_}cvxboo}](https://web.stanford.edu/~boyd/cvxbook/bv_{_}cvxboo)>.

BRACKEN, J.; MCGILL, J. Production and marketing decisions with multiple objectives in a competitive environment. *Journal of Optimization Theory and Applications*, Springer, v. 24, n. 3, p. 449–458, 1978.

BRACKEN, J.; MCGILL, J. T. Defense applications of mathematical programs with optimization problems in the constraints. *Operations Research*, INFORMS, v. 22, n. 5, p. 1086–1096, 1974.

BRANDÃO, A. S. M. *CONTROLE PREDITIVO COM GERAÇÃO DE CÓDIGO: UM ESTUDO COMPARATIVO*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, 2018.

CAMACHO, E. F.; BORDONS, C. *Model Predictive Control*. 2. ed. [S.l.]: Springer Science & Business Media, 2007.

CAMPONOGARA, E. et al. Distributed model predictive control. *IEEE Control Systems*, IEEE, v. 22, n. 1, p. 44–52, feb 2002. Available from Internet: <<https://ieeexplore.ieee.org/document/980246/>>.

CAMPONOGARA, E. et al. Hierarchical decompositions for MPC of resource constrained control systems: applications to building energy management. *Optimization and Engineering*, Springer US, n. 0123456789, p. 1–26, may 2020. Available from Internet: <<https://doi.org/10.1007/s11081-020-09506-x>>.

CHING, J.; BECK, J. L.; PORTER, K. A. Bayesian state and parameter estimation of uncertain dynamical systems. *Probabilistic Engineering Mechanics*, Elsevier, v. 21, n. 1, p. 81–96, jan 2006. Available from Internet: <<https://www.sciencedirect.com/science/article/abs/pii/S0266892005000597>>.

CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control—part i. the basic algorithm. *Automatica*, Elsevier, v. 23, n. 2, p. 137–148, 1987.

CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control—part ii extensions and interpretations. *Automatica*, Elsevier, v. 23, n. 2, p. 149–160, 1987.

COLONETTI, B.; FINARDI, E. Combining Lagrangian relaxation, Benders decomposition, and the level bundle method in the stochastic hydrothermal unit-commitment problem. *International Transactions on Electrical Energy Systems*, Wiley Online Library, v. 1, p. 1–10, 2020.

COLSON, B.; MARCOTTE, P.; SAVARD, G. An overview of bilevel optimization. *Annals of Operations Research*, v. 153, n. 1, p. 235–256, 2007.

CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control?? a computer control algorithm. n. 17, p. 72, 1980.

DEMPE, S. et al. *Bilevel Programming Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015. 205–231 p. (Energy Systems). Available from Internet: <<http://link.springer.com/10.1007/978-3-662-45827-3>>.

DUNNING, I.; HUCHETTE, J.; LUBIN, M. JuMP: A Modeling Language for Mathematical Optimization. *SIAM Review*, v. 59, n. 2, p. 295–320, jan 2017. Available from Internet: <<https://epubs.siam.org/doi/10.1137/15M1020575>>.

DURAN, M. A.; GROSSMANN, I. E. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical programming*, Springer, v. 36, n. 3, p. 307–339, 1986.

FADALI, M. S.; VISIOLI, A. *Digital Control Engineering: Analysis and Design*. [S.l.]: Academic Press, 2013.

FLOUDAS, C. A. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. [S.l.]: Oxford University Press, USA, 1995. 450 p.

GEOFFRION, A. M. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, v. 10, n. 4, p. 237–260, 1972.

GROSSMANN, I. Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques. *Optimization and Engineering*, v. 3, n. 3, p. 227–252, 2002.

GUIGNARD, M.; KIM, S. Lagrangean decomposition: A model yielding stronger Lagrangean bounds. *Mathematical Programming*, v. 39, n. 2, p. 215–228, jun 1987. Available from Internet: <<http://link.springer.com/10.1007/BF02592954>>.

GÜMÜŞ, Z. H.; FLOUDAS, C. A. Global optimization of nonlinear bilevel programming problems. *Journal of Global Optimization*, v. 20, n. 1, p. 1–31, 2001.

JULIA, L. *Julia Language Webpage*. [S.l.], 2020. Accesed June 22, 2020. Available from Internet: <<https://julialang.org/>>.

KALASHNIKOV, V. V. et al. Bilevel programming and applications. *Mathematical Problems in Engineering*, v. 2015, 2015.

KALMAN, R. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering*, v. 82, p. 35–45, 1960.

KARMARKAR, N. A new polynomial-time algorithm for linear programming. In: *Proceedings of the sixteenth annual ACM symposium on Theory of computing - STOC '84*. New York, New York, USA: ACM Press, 1984. v. 4, n. April, p. 302–311. Available from Internet: <<http://portal.acm.org/citation.cfm?doid=800057.808695>>.

KOEHLER, L. A. et al. Real-Time Integrated Holding and Priority Control Strategy for Transit Systems. *IEEE Transactions on Intelligent Transportation Systems*, IEEE, v. 20, n. 9, p. 3459–3469, 2018.

MENDES, P. R. et al. Energy management of an experimental microgrid coupled to a v2g system. *Journal of Power Sources*, Elsevier, v. 327, p. 702–713, 2016.

MENDES, P. R. D. C. *Predictive Control for Energy Management of Renewable Energy Based Microgrids*. 156 p. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2016.

MENDES, P. R. D. C.; NORMEY-RICO, J. E.; ALBA, C. B. Economic energy management of a microgrid including electric vehicles. *2015 IEEE PES Innovative Smart Grid Technologies Latin America, ISGT LATAM 2015*, IEEE, p. 869–874, 2016.

NOCEDAL, J.; WRIGHT, S. J. *Numerical Optimization*. 2. ed. New York, NY, USA: Springer, 2006.

NORMEY-RICO, J. E.; CAMACHO, E. F. *Control of Dead-time Processes*. London: Springer London, 2007. (Advanced Textbooks in Control and Signal Processing).

QUESADA, I.; GROSSMANN, I. E. An lp/nlp based branch and bound algorithm for convex minlp optimization problems. *Computers & chemical engineering*, Elsevier, v. 16, n. 10-11, p. 937–947, 1992.

RAHMANIANI, R. et al. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, Elsevier B.V., v. 259, n. 3, p. 801–817, 2017. Available from Internet: <<http://dx.doi.org/10.1016/j.ejor.2016.12.005>>.

SANDELL N., J. et al. Survey of decentralized control methods for large scale systems. *IEEE Transactions on Automatic Control*, v. 23, n. 2, p. 108–128, 1978.

SCHERER, H. et al. Distributed MPC for resource-constrained control systems. *Optimal Control Applications and Methods*, v. 36, n. 3, p. 272–291, may 2015. Available from Internet: <<http://doi.wiley.com/10.1002/oca.2151>>.

SCHERER, H. et al. Process Control Efficient Building Energy Management Using Distributed Model Predictive Control. *Journal of Process Control*, v. 24, p. 740–749, 2013.

SINHA, A.; MALO, P.; DEB, K. A Review on Bilevel Optimization: From Classical to Evolutionary Approaches and Applications. *IEEE Transactions on Evolutionary Computation*, IEEE, v. 22, n. 2, p. 276–295, 2018.

TUDOROIU, R.-E. et al. Estimation Techniques for State of Charge in Battery Management Systems on Board of Hybrid Electric Vehicles Implemented in a Real-Time MATLAB/SIMULINK Environment. In: *New Trends in Electrical Vehicle Powertrains*. IntechOpen, 2019. cap. 4, p. 28. Available from Internet: <[https://www.intechopen.com/books/new-trends-in-electrical-vehicle-powertrains/estimation-techniques-for-state-of-charge-in-battery-management-systems-on-board-of-hybrid-electric->](https://www.intechopen.com/books/new-trends-in-electrical-vehicle-powertrains/estimation-techniques-for-state-of-charge-in-battery-management-systems-on-board-of-hybrid-electric-)>.

VANDERBEI, R. J. *Linear Programming*. 4. ed. New York: Springer Science & Business Media, 2014. (International Series in Operations Research & Management Science, v. 196). Available from Internet: <<http://link.springer.com/10.1007/978-1-4614-7630-6>>.

WÄCHTER, A.; BIEGLER, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, Springer-Verlag, v. 106, n. 1, p. 25–57, mar 2006. Available from Internet: <<http://link.springer.com/10.1007/s10107-004-0559-y>>.

WESTERLUND, T.; PETTERSSON, F. An extended cutting plane method for solving convex minlp problems. *Computers & Chemical Engineering*, Pergamon, v. 19, p. 131–136, 1995.

YOU, F.; GROSSMANN, I. E. Multicut benders decomposition algorithm for process supply chain planning under uncertainty. *Annals of Operations Research*, Springer, v. 210, n. 1, p. 191–211, 2013.