SEVENTH FRAMEWORK PROGRAMME
THEME – ICT
[Information and Communication Technologies]

Contract Number:	223854
Project Title:	Hierarchical and Distributed Model Predictive Control of Large-Scale Systems
Project Acronym:	HD-MPC

Deliverable Number:	D4.1.1
Deliverable Type:	Report
Contractual Date of Delivery:	01/03/2009
Actual Date of Delivery:	**28/08/2009**
Title of Deliverable:	**Report of literature survey, analysis, and comparison of on-line optimization methods for hierarchical and distributed MPC**
Dissemination level:	Public
Workpackage contributing to the Deliverable:	WP4
WP Leader:	Moritz Diehl
Partners:	TUD, KUL, USE, UWM
Author(s):	Carlo Savorgnan, Moritz Diehl

Table of contents

Executive Summary **3**

1 Introduction **4**
1.1 Model predictive control 4
1.2 Distributed model predictive control 5

2 Methods for distributed optimization **7**
2.1 The Jacobi and Gauss-Seidel algorithms 8
2.2 Primal decomposition – resource allocation 9
2.3 Dual decomposition – price coordination 10
2.4 Remarks 11

3 Discussion and conclusions **12**
3.1 Number of iterations and feasibility issues 12
3.2 Smoothness of the master problem 13
3.2.1 Augmented Lagrangian Method 13
3.2.2 Proximal minimization algorithm 13
3.2.3 Proximal center algorithm 13

Bibliography **15**

Project co-ordinator

Name: Bart De Schutter
Address: Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 Delft, The Netherlands
Phone Number: +31-15-2785113
Fax Number: +31-15-2786679
E-mail: b.deschutter@dcsc.tudelft.nl

Project web site: `http://www.ict-hd-mpc.eu`

Executive summary

This report gives an overview of the optimization methods that can be applied to hierarchical and distributed MPC. The last chapter contains some discussion and conclusions, which are relevant to MPC and which suggest some possible research directions which can be investigated in the HD-MPC project.

Chapter 1

Introduction

This report gives an overview of optimization methods for hierarchical and distributed MPC. In this chapter we briefly recall the model predictive control methodology and define the notation that we shall use in the rest of the report.

1.1 Model predictive control

Consider a system described by the difference equation

$$x_{t+1} = \phi(x_t, u_t), \quad t = 0, 1, 2, \ldots \tag{1.1}$$

where t represents the time and the vectors $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ represent the state and input, respectively. Given the stage cost

$$\ell_t(x_t, u_t), \tag{1.2}$$

the constraints

$$x_t \in \mathscr{X} \subseteq \mathbb{R}^n \quad \text{and} \quad u_t \in \mathscr{U} \subseteq \mathbb{R}^m, \quad t = 0, 1, 2, \ldots, \tag{1.3}$$

and the initial condition $\bar{x}_0$, we consider the optimal control problem (OCP) of finding a control sequence $\{u_t\}_{t=0}^{\infty}$, such that the cost

$$\sum_{t=0}^{\infty} \ell_t(x_t, u_t), \tag{1.4}$$

is minimized.

This problem is difficult to solve in general due to the fact that we have an infinite number of variables. The MPC strategy tackles this difficulty by repetitively solving a finite horizon approximation of the original infinite horizon OCP. The method can be summarized in the following steps

1. Choose a prediction horizon N;
2. Measure the value of the state at the current time $\bar{t}$ and denote it by $\bar{x}$;
3. Solve the optimization problem

$$\begin{array}{cl} \min\limits_{\substack{x_0,\ldots,x_N \\ u_0,\ldots,u_{N-1}}} & \sum_{t=0}^{N} \ell_t(x_t, u_t) \\ \text{s.t.} & x_0 = \bar{x} \\ & x_{t+1} = \phi(x_t, u_t) \;\; \forall t = 0, \ldots, N-1 \\ & x_t \in \mathscr{X} \;\; \forall t = 0, \ldots, N \\ & u_t \in \mathscr{U} \;\; \forall t = 0, \ldots, N-1 \end{array} \tag{1.5}$$

4. Apply the computed value of the input u_0;

5. Go to step (2).

Remark 1 *The considered OCP and the corresponding MPC method here are basic. Nevertheless, the optimization methods discussed in this report can be applied to more sophisticated MPC applications (see, e.g., the books [4, 14]).*

1.2 Distributed model predictive control

Consider a networked system composed of M interconnected subsystems, each one described by a difference equation of the form

$$x_{t+1}^i = \phi^i(x_t^j, u_t^j; j \in \mathcal{N}_i), \quad t = 0, 1, 2, \dots \tag{1.6}$$

where $x_t^i \in \mathbb{R}^{n_{x_i}}$, $u_t^i \in \mathbb{R}^{n_{u_i}}$ represent the state and the input of the subsystem i at time t. The index set $\mathcal{N}^i$ contains index i and all indices of the subsystems that interact with the subsystem i. Consider for example the networked system in Figure 1.1, where the arrows indicate interaction between the subsystems $\Sigma_1, \Sigma_2, \Sigma_3, \Sigma_4$. If we consider Σ_4 we have $\mathcal{N}^4 = \{3, 4\}$ and therefore

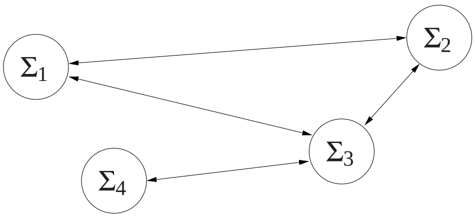

Figure 1.1: An example of networked systems.

$$x_{t+1}^4 = \phi^4(x_t^3, x_t^4, u_t^3, u_t^4).$$

When we consider a networked systems, the optimization problem to be solved at every MPC iteration has usually a particular structure which can be exploited to obtain a distributed MPC scheme. This will be the subject of the remainder of this report.

To simplify the exposition in the following chapters we introduce the following notation:

$$\mathbf{x}^i = \left[x_0^{i\,T} u_0^{i\,T} \dots x_{N-1}^{i\,T} u_{N-1}^{i\,T} x_N^{i\,T} \right]^T, \; \mathbf{x} = \left[\mathbf{x}^{1T} \dots \mathbf{x}^{MT} \right]^T. \tag{1.7}$$

Using this notation problem (1.5) can be now written in the form

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & h(\mathbf{x}) = 0 \\ & \mathbf{x} \in \mathscr{X} \end{array} \tag{1.8}$$

where

$$f(\mathbf{x}) = \sum_{t=0}^{N} \ell_t(x_t, u_t), \tag{1.9}$$

the function $h(\mathbf{x})$ is obtained by stacking all the equalities constraints coming from the dynamics $x^i_{t+1} = \phi^i(x^j_t, u^j_t; j \in \mathcal{N}_i)$ and $\mathbf{x} \in \mathscr{X}$ represent all the other constraints.

Depending on the nature of the MPC problem considered, the functions f and h and the set $\mathscr{X}$ can have some separability properties with respect to the variables $\mathbf{x}^i$. In the next chapter we will define these separability properties and we will show how we can exploit them to obtain a distributed MPC scheme. In particular, we will show how problem (1.8) can be decomposed to obtain a method where every subsystem solves a simple subproblem, possibly under the coordination of a central control unit.

Chapter 2

Methods for distributed optimization

The literature on distributed optimization is wide, since for decades, researches have studied different techniques to solve computationally demanding problems exploiting distributed and parallel computing. In this chapter we survey the main methods to solve optimization problems of the form (1.8), which arise in the distributed MPC framework.

For illustrative convenience, let us rewrite problem (1.8)

$$\begin{array}{ll} \min\limits_{\mathbf{x}} & f(\mathbf{x}) \\ \text{s.t.} & h(\mathbf{x}) = 0 \\ & \mathbf{x} \in \mathscr{X} \end{array} \tag{2.1}$$

To effectively distribute the computation of the solution of problem (2.1) and obtain a distributed MPC scheme we need to define the following separability properties (the $\mathbf{x}^i$ used in the definitions are the local vectors defined in (1.7)).

- The cost function f is separable if there exist functions $f_1, \dots, f_M$ such that

$$f(\mathbf{x}) = \sum_{i=1}^{M} f^i(\mathbf{x}^i). \tag{2.2}$$

- The constraint set $\mathscr{X}$ is separable if it can be written as

$$\mathscr{X} = \mathcal{X}^1 \times \cdots \times \mathcal{X}^M \tag{2.3}$$

  and the constraint $\mathbf{x} \in \mathscr{X}$ is equivalent to $\mathbf{x}^i \in \mathcal{X}^i \ \ \forall i = 1, \dots, M$.

- The constraint $h(\mathbf{x})$ is separable if it can be written as

$$h^i(\mathbf{x}^i) = 0 \quad \forall i = 1, \dots, M. \tag{2.4}$$

When the optimization problem (2.1) arises from a distributed MPC problem it usually presents one or more of the separability properties above. Clearly, when all the separability properties hold, problem (2.1) is constituted of M independent problems of the form

$$\begin{array}{ll} \min\limits_{\mathbf{x}^i} & f^i(\mathbf{x}^i) \\ \text{s.t.} & h^i(\mathbf{x}^i) = 0 \\ & \mathbf{x}^i \in \mathcal{X}^i \end{array} \tag{2.5}$$

which can be solved separately. In the reminder of this chapter we shall illustrate the methods that can be used to solve the central problem (2.1) in a distributed manner when it cannot be trivially written as M independent problems.

In the following we will often refer to subsystems as nodes. The two terms will be used interchangeably.

2.1 The Jacobi and Gauss-Seidel algorithms

Consider an optimization problems where h and $\mathscr{X}$ are separable

$$\begin{array}{ll} \min\limits_{\mathbf{x}^1,\dots,\mathbf{x}^M} & f(\mathbf{x}^1,\dots,\mathbf{x}^M) \\ \text{s.t.} & h^i(\mathbf{x}^i) = 0 \qquad \forall i = 1,\dots,M \\ & \mathbf{x}^i \in \mathscr{X}^i \qquad \forall i = 1,\dots,M \end{array} \tag{2.6}$$

This kind of problems may not be interesting from the distributed MPC perspective, since it would correspond to a control problem where all the subsystem dynamics and constraints are completely decoupled. However, we discuss this case for completeness and because the Gauss-Seidel and Jacobi algorithms can be used as a building block for other algorithms.

The algorithms illustrated in this section are based on the idea of iteratively solving problem (2.6). Every node computes a new value of the local variable $\mathbf{x}^i$, while keeping all the other variables fixed. In each iteration, the optimization phase is followed by a broadcasting phase, where the results are communicated to the other nodes. The two methods differ in the way the two phases are organized.

In the Jacobi algorithm all nodes i updates the value of $\mathbf{x}^i$ simultaneously. The optimization problem solved at each node reads

$$\begin{array}{lll} \mathbf{x}^i_{k+1} = \arg & \min\limits_{\mathbf{x}^i} & f(\mathbf{x}^1_k,\dots,\mathbf{x}^{i-1}_k,\mathbf{x}^i,\mathbf{x}^{i+1}_k\dots,\mathbf{x}^M_k) \\ & \text{s.t.} & h^i(\mathbf{x}^i) = 0 \\ & & \mathbf{x}^i \in \mathscr{X}^i \end{array} \tag{2.7}$$

After problem (2.7) has been solved, the variables $\mathbf{x}^i_{k+1}$ are sent to the other nodes.

In the Gauss-Seidel algorithm (known also as block-coordinate descent [2]) the variables $\mathbf{x}^i$ are updated sequentially. The local optimization problem reads

$$\begin{array}{lll} \mathbf{x}^i_{k+1} = \arg & \min\limits_{\mathbf{x}^i} & f(\mathbf{x}^1_{k+1},\dots,\mathbf{x}^{i-1}_{k+1},\mathbf{x}^i,\mathbf{x}^{i+1}_k\dots,\mathbf{x}^M_k) \\ & \text{s.t.} & h^i(\mathbf{x}^i) = 0 \\ & & \mathbf{x}^i \in \mathscr{X}^i \end{array} \tag{2.8}$$

After solving problem (2.8) the value of $\mathbf{x}^i_{k+1}$ is sent to all the other nodes.

Remark 2 *In general, the local optimization problems in the Gauss-Seidel algorithm cannot be solved simultaneously. When a coloring scheme can be applied, however, part of the computation can be carried out in parallel [2, Subsection 1.2.4].*

More details on this methods, including a proof of convergence, can be found in [2].

In the next section, we shall discuss decomposition methods that can be applied to problems where there are constraints that couple two or more vector variables $\mathbf{x}^i$. These kind of problems are interesting for distributed MPC, since they occur when the subsystems have coupled dynamics or when different subsystems share a common resource.

2.2 Primal decomposition – resource allocation

In this section ,we illustrate how a problem can be decomposed by resource allocation. This method can be applied to problems that become separable when one or more variables are fixed. The method will be explained by means of an example.

Consider the problem

$$\begin{array}{lll} \min\limits_{\mathbf{x}^1,\mathbf{x}^2} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2) & \\ \text{s.t.} & h^1(\mathbf{x}^1)=0 & \mathbf{x}^1 \in \mathscr{X}^1 \\ & h^2(\mathbf{x}^2)=0 & \mathbf{x}^2 \in \mathscr{X}^2 \\ & a^T\mathbf{x}^1+b^T\mathbf{x}^2 \le c & \end{array} \tag{2.9}$$

where a and b are vectors of appropriate dimension and c is a scalar. The set $\mathscr{X}$ in the centralized problem can be written for this problem as $\mathscr{X} = \{\mathscr{X}^1 \times \mathscr{X}^2\} \cap \{(\mathbf{x}^1,\mathbf{x}^2)|a^T\mathbf{x}^1+b^T\mathbf{x}^2 \le c\}$. Therefore, the only coupling contraint in problem (2.9) is given by the inequality constraint. This kind of problems can easily arise in distributed MPC when two subsystems share a limited resource.

To reformulate problem (2.9) in a way that allows resource allocation, we introduce a variable γ to split the coupling inequality constraint into two inequalities

$$\begin{array}{lll} \min\limits_{\mathbf{x}^1,\mathbf{x}^2,\gamma} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2) & \\ \text{s.t.} & h^1(\mathbf{x}^1)=0 & \mathbf{x}^1 \in \mathscr{X}^1 \\ & h^2(\mathbf{x}^2)=0 & \mathbf{x}^2 \in \mathscr{X}^2 \\ & a^T\mathbf{x}^1 \le \gamma & \\ & b^T\mathbf{x}^2 \le c-\gamma & \end{array} \tag{2.10}$$

It is clear that if we fix γ we obtain two decoupled subproblems

$$\begin{array}{llll} \phi^1(\gamma) = & \min\limits_{\mathbf{x}^1} & f^1(\mathbf{x}^1) \\ & \text{s.t.} & h^1(\mathbf{x}^1)=0 \\ & & \mathbf{x}^1 \in \mathscr{X}^1 \\ & & a^T\mathbf{x}^1 \le \gamma \end{array} \qquad \begin{array}{llll} \phi^2(\gamma) = & \min\limits_{\mathbf{x}^2} & f^2(\mathbf{x}^2) \\ & \text{s.t.} & h^2(\mathbf{x}^2)=0 \\ & & \mathbf{x}^2 \in \mathscr{X}^2 \\ & & b^T\mathbf{x}^2 \le c-\gamma \end{array} \tag{2.11}$$

Problem (2.9) is then equivalent to the master program

$$\min_{\gamma} \ \phi^1(\gamma)+\phi^2(\gamma) \tag{2.12}$$

This problem can be solved iteratively by choosing a value of γ_k and then minimizing the functions $\phi^1(\gamma_k)$ and $\phi^2(\gamma_k)$ in parallel. The update of γ_k can be done using a subgradient scheme

$$\gamma_{k+1} = \gamma_k - \alpha_k g_k \tag{2.13}$$

where α_k is the step size and g_k is the subgradient of $\phi^1(\gamma_k)+\phi^2(\gamma_k)$. Notice that g_k can be written as $g_k = -\lambda^1(\gamma_k)+\lambda^2(\gamma_k)$ where $\lambda^1(\gamma_k)$ and $\lambda^2(\gamma_k)$ are Lagrange multipliers associated to the inequalities $a^T\mathbf{x}^1 \le \gamma$ and $b^T\mathbf{x}^2 \le c-\gamma$ in the two subproblems defined in (2.11). (see [1, Chapter 4] or [3, Chapter 5] for more information about Lagrange multipliers and their sensitivity interpretation). When the master problem is solved using this scheme, the method has an interesting economic interpretation. At every iteration the master program allocates the resources (by choosing γ_k) and the nodes return the prices associated with this choice ($\lambda^1(\gamma_k)$ and $\lambda^2(\gamma_k)$). The iterations continue until the prices have reached the equilibrium $\lambda^1(\gamma_k)+\lambda^2(\gamma_k)=0$. In the next section, we illustrate a decomposition method in which this pricing scheme is reversed: the master program decides the prices whereas the nodes allocate the resources.

An interesting fact concerning the the primal decomposition method via resource allocation is that under suitable conditions this method can be implemented such that the current solutions $\mathbf{x}^1$ and $\mathbf{x}^2$ of the subproblems is feasible at every iteration [15, 16].

2.3 Dual decomposition – price coordination

When the cost function is separable a problem can be decomposed exploiting Lagrangian duality. This method is illustrated at length in [19, 7, 2]. Following the approach of the previous section we will explain the method by means of an example.

Consider the problem

$$\begin{array}{lll} \min\limits_{\mathbf{x}^1,\mathbf{x}^2} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2) & \\ \text{s.t.} & h^1(\mathbf{x}^1)=0 & \mathbf{x}^1\in\mathscr{X}^1 \\ & h^2(\mathbf{x}^2)=0 & \mathbf{x}^2\in\mathscr{X}^2 \\ & a^T\mathbf{x}^1+b^T\mathbf{x}^2=c & \end{array} \tag{2.14}$$

where a and b are vectors of suitable dimension and c is a scalar. In problem (2.14), the only coupling constraint is given by the last equality. Problems of this form frequently arise in distributed MPC when the dynamics of one subsystem depends on one or more state and input variables of other subsystems.

To decompose the problem, we consider the partial Lagrangian

$$\mathscr{L}(\mathbf{x}^1,\mathbf{x}^2,\lambda)=f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2)+\lambda(a^T\mathbf{x}^1+b^T\mathbf{x}^2-c) \tag{2.15}$$

where we have introduced a Lagrange multiplier for the coupling constraints only. The dual corresponding problem, which will be referred to as the master problem, reads

$$\max_{\lambda}\ g^1(\lambda)+g^2(\lambda)-\lambda c \tag{2.16}$$

where

$$g^1(\lambda)=\begin{array}{ll} \min\limits_{\mathbf{x}^1} & f^1(\mathbf{x}^1)+\lambda a^T\mathbf{x}^1 \\ \text{s.t.} & h^1(\mathbf{x}^1)=0 \\ & \mathbf{x}^1\in\mathscr{X}^1 \end{array} \qquad g^2(\lambda)=\begin{array}{ll} \min\limits_{\mathbf{x}^2} & f^2(\mathbf{x}^2)+\lambda b^T\mathbf{x}^2 \\ \text{s.t.} & h^2(\mathbf{x}^2)=0 \\ & \mathbf{x}^2\in\mathscr{X}^2 \end{array} \tag{2.17}$$

Similarly to resource allocation, price coordination also uses an iterative algorithm. The method alternates an update of λ with the solution of the subproblems which can be carried out in parallel. The update of λ can be done using a subgradient method or any other suitable method.

This decomposition method also has an interesting economic interpretation. At every iteration the master problem decides the price associated to the coupling constraint (the value of λ) and the nodes accordingly decide the value of the resources ($\mathbf{x}^1$ and $\mathbf{x}^2$). The iterations continue until the equilibrium $a^T\mathbf{x}^1+b^T\mathbf{x}^2-c$ has been reached.

Remark 3 *At every iteration there is no guarantee that the primal variables are feasible. Only when the method converged to a solution we have this guarantee.*

Remark 4 *Dual decomposition should be used only when strong duality hold for problem* (2.14). *If there is no strong duality, the solutions of problems* (2.14) *and* (2.16) *may not be equivalent.*

2.4 Remarks

In this chapter we introduced the basic optimization techniques that can be exploited to designed hierarchical and distributed MPC algorithms. To make this methods applicable to a particular problem we assumed specific separability properties. However, it is important to notice that by suitable transformation, we can often rewrite the problem under consideration such that the new formulation has the required properties.

Chapter 3

Discussion and conclusions

In this chapter we discuss some important aspects that should be taken into account when designing a distributed MPC algorithm.

3.1 Number of iterations and feasibility issues

In the previous chapter we presented some basic methods for distributed optimization. However, we did not discuss the efficiency of this methods, which is an important aspect when we consider distributed MPC. In fact, MPC requires the online solution of optimization problem (2.1) in a limited amount of time. This is especially important when the dynamics of the system are fast. Since complete discussion of the efficiency of the illustrated methods is out of the scope of this report, we decided to point out an aspect which we believe is relevant for distributed MPC.

When we decompose a centralized optimization problem, we may need a large number of iterations before convergence is achieved (the numerical examples in [13, 9] indicate that we may need more than 100 iterations even for simple problems). Therefore, the time required by the distributed optimization algorithm used may be too big for the requirements dictated by MPC. This observation suggests that instead of solving the optimization problem completely we may perform only a limited number of iterations, if using a suboptimal solution is acceptable for our application. If we decide to adopt such kind of strategy it is essential to stop the algorithm only when the current solution is feasible for the system. In this regard, the methods illustrated in the previous chapter perform differently.

- The Jacobi and Gauss-Seidel algorithms produce feasible solutions for all the iterations. This is made easy by the fact the this methods can be applied when we have no coupling constraints. However, as we already noticed coupling constraints occur in practice for distributed MPC applications and this fact make these methods less interesting.

- As already pointed out in section 2.2, it is possible to produce feasible solutions for every iterations also when we use primal decomposition [15, 16], but in general this is difficult. In fact, to obtain a feasible solution from the nodes, it is essential that the master problem chooses a feasible resource allocation.

- When dual decomposition is used, it is difficult to produce feasible solutions at each iteration. The reason is that the satisfaction of the coupling constraints is achieved by the master problem only by choosing a suitable value of the Lagrange multipliers.

It is interesting to notice that the scientific literature has given little attention to the study of methods which produce feasible solutions at every iteration. This is probably due to the fact that many applications for which distributed optimization was studied, have no hard time constraints and one can wait for algorithm convergence. In view of this fact, one possible research direction which could be further investigated in the HD-MPC project is the study of decomposition methods which are designed to quickly achieve feasible suboptimal solutions for distributed optimization problems with coupling constraints.

3.2 Smoothness of the master problem

In Sections 2.2 and 2.3, we have seen that the master problem in the primal and dual decomposition can be solved by a subgradient method. Although it is commonly used, the subgradient method works poorly for some problems. For this reason several techniques have been studied to obtain a differentiable master problem in the dual decomposition method. This methods are based on the fact that when the cost function of the primal problem is strictly convex, then the cost function of the master problem is differentiable [17, 2, 18, 5, 8, 10].

3.2.1 Augmented Lagrangian Method

Consider the problem

$$\begin{array}{ll} \min\limits_{\mathbf{x}^1,\mathbf{x}^2} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2) \\ \text{s.t.} & a^T\mathbf{x}^1+b^T\mathbf{x}^2=c \end{array} . \tag{3.1}$$

The idea of the *augmented Lagrangian method* is to add a term to the primal cost function such that the new problem has the same minimizers and strict convexity is improved:

$$\begin{array}{ll} \min\limits_{\mathbf{x}^1,\mathbf{x}^2} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2)+\frac{\gamma}{2}\parallel a^T\mathbf{x}^1+b^T\mathbf{x}^2-c\parallel^2 \\ \text{s.t.} & a^T\mathbf{x}^1+b^T\mathbf{x}^2=c \end{array} . \tag{3.2}$$

Unfortunately, the master problem obtained by applying the dual decomposition to (3.2), is not separable in $\mathbf{x}^1$ and $\mathbf{x}^2$ and one must use the Jacobi or Gauss-Seidel algorithm for its solution [2].

3.2.2 Proximal minimization algorithm

Another method to obtain a differentiable master problem is adding a prox term to the cost function of the primal problem to obtain strict convexity. Consider problem (3.1). The *proximal minimization algorithm* iteratively solves the problem

$$\begin{array}{ll} (\mathbf{x}^1_{k+1},\mathbf{x}^2_{k+1})=\arg\min\limits_{\mathbf{x}^1,\mathbf{x}^2} & f^1(\mathbf{x}^1)+f^2(\mathbf{x}^2_{)}+\frac{\gamma}{2}\left\|\left[\mathbf{x}^{1^T}\ \mathbf{x}^{2^T}\right]^T-\left[\mathbf{x}_k^{1^T}\ \mathbf{x}_k^{2^T}\right]^T\right\|^2 \\ \text{s.t.} & a^T\mathbf{x}^1+b^T\mathbf{x}^2=c \end{array} . \tag{3.3}$$

Problem (3.3) is then solved using dual decomposition [2].

3.2.3 Proximal center algorithm

A common problem in the augmented Lagrangian method and the proximal minimization algorithm is that the parameter γ, which strongly influences the convergence, is difficult to tune. Some heuristics to solve this problem can be found in [6, 5]. Another limitation of this methods is that they use a gradient method which was proved to be less efficient than Nesterov's optimal method [11].

In [9, 10], Nesterov's optimal method is combined with smoothing techniques inspired by [12] to obtain a method whose efficiency is proved to be of one order of magnitude better than methods based on gradient updates. For a complete description of the algorithm, see [9].

Bibliography

[1] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] D.P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation*. Prentice Hall, 1989.

[3] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[4] E.F. Camacho and C. Bordons. *Model Predictive Control*. Springer, London, 2004.

[5] G. Chen and M. Teboulle. A proximal-based decomposition method for convex minimization problems. *Mathemathical Programming A*, 64:81–101, 1994.

[6] S. Kontogiorgis, R. De Leone, and R. Meyer. Alternating direction splitings for block angular parallel optimization. *J. Opt. Theory and Appl.*, 90:1–29, 1996.

[7] L.S. Lasdon. *Optimization Theory for Large Systems*. MacMillan, 1970.

[8] P. Mahey, S. Oualibouch, and P. Tao. Proximal decomposition on the graph of a maximal monotone operator. *SIAM Journal of Optimization*, 5:454–466, 1995.

[9] I. Necoara and J.A.K. Suykens. Application of a Smoothing Technique to Decomposition in Convex Optimization. *IEEE Trans. Automat. Contr.*, 53:2674–2679, 2008.

[10] I. Necoara and J.A.K. Suykens. An interior-point Lagrangian decomposition method for separable convex optimization. *Journal of Optimization Theory and Applications*, 2009.

[11] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Springer, 2004.

[12] Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Programming*, 103(1):127–152, 2005.

[13] D.P. Palomar and M. Chiang. Alternative Distributed Algorithms for network Utility Maximization: Framework and Applications. *IEEE Trans. Automat. Contr.*, 52:2254–2269, 2007.

[14] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin, 2009.

[15] G.J. Silverman. Primal decomposition of mathematical programs by resource allocation. I. Basic theory and a direction-finding procedure. *Operations Research*, 20:58–74, 1972.

[16] G.J. Silverman. Primal decomposition of mathematical programs by resource allocation. II. Computational algorithm with an application to the modular design problem. *Operations Research*, 20:75–93, 1972.

[17] J. E. Spingarn. Applications of the method of partial inverses to convex programming: decomposition. *Mathematical Programming (A)*, 32:199–223, 1985.

[18] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Contr. Optim.*, 29:119–138, 1991.

[19] H. Uzawa. Iterative methods for concave programming. In K. Arrow, L. Hurwicz, and H. Uzawa, editors, *Studies in Linear and Nonlinear Programming*, 1958.