Scalability study for a hierarchical NMPC scheme for resource sharing problems

Peter Pflaum[1] and Mazen Alamir[2] and Mohamed Yacine Lamoudi[1]

***Abstract*—This paper deals with the computational efficiency evaluation of a hierarchical DMPC (distributed model predictive control) framework for resource sharing problems. The provided DMPC framework is based on a dual decomposition of the centralized open-loop controller which is decomposed into several subproblems and one coordinator problem. At coordinator level the bundle method is used in order to recover the globally optimal solution through an iterative process.**

The main focus of this paper is a detailed discussion of the impact of the bundle method's parametrization on the computational performance of the whole scheme. Additionally a qualitative comparison with a similar scheme based on primal decomposition is provided and some rules of thumb for determining an effective parametrization of the bundle method are established. In the provided simulations the scheme is applied to a large-scale problem of the smart district context. More precisely the centralized optimization problem of a district composed of 1000 buildings sharing a globally limited power resource is able be solved to optimality using our proposed framework in around 3 seconds.

***Index Terms*—Dual decomposition, bundle method, smart district, DMPC, large-scale optimization**

I. Introduction

Resource sharing problems are playing a crucial role in today's modern society where continuous economic growth needs to be achieved while facing more and more limited resources. The most promising way to solve this dilemma today is to improve the efficiency of using those resources. For instance in transportation systems important savings could be possible if transportation capacities were more optimally coordinated or in electrical power grids environmentally harmful power stations using limited fossil energy carriers could be avoided if the energy usage was better coordinated between producers and consumers.

One promising technique which is able to achieve these objectives is Distributed Model Predictive Control (DMPC). In DMPC a large-scale optimization problem is divided into many sub-problems which are then solved individually. Through an iterative communication scheme the local controllers are able to recover the optimal solution of the centralized problem or at least to find a relevant sub-optimal solution. The main reason why DMPC approaches are more suited to the considered class of resource sharing problems than classical centralized MPC controllers is due to the problems' large-scale character which often makes a centralized controller intractable.

Since a detailed discussion of available DMPC techniques is beyond the scope of this article, for more general information on DMPC, the reader is referred to [1] where a collection of state-of-the art DMPC-techniques is provided. The following literature review directly focuses on the class of resource sharing problems considered for this paper where several subsystems with decoupled and possibly nonlinear dynamics are subject to one or more shared and limited resources. In [6] different algorithms for optimal file allocation problems in distributed computer systems are discussed and similarities with micro-economic systems are emphasized. In [8] a DMPC scheme based on a primal decomposition and using a disaggregated bundle method to solve the coordinator problem is shown to require remarkably few iterations in receding-horizon mode in order to achieve global optimality. A discussion of different decomposition methods is provided in [9] where amongst others primal and dual decomposition techniques are discussed with an emphasis on resource sharing problems. Finally, in [10], a comparison of two DMPC frameworks which are based on a primal and on a dual decomposition is performed in the context of a power allocation problem in smart districts.

In this work we consider a general nonlinear DMPC scheme for resource sharing problems. An algorithm is proposed which is based on a dual decomposition of the centralized problem and it is solved using different versions of the bundle method, namely disaggregated, aggregated and partially aggregated bundle method. The central contribution in this work is the discussion of the cost in terms of computation time that occurs when the resource sharing problem becomes very large and how these difficulties can be overcome. In the scope of this discussion we also consider the very similar scheme proposed in [8] which is based on a primal decomposition and mention its advantages and drawbacks compared to our proposed approach.

In a simulation section we provide simulation results where the proposed scheme is applied to a problem of the smart district context where a limited amount of energy is the shared resource between 1000 buildings. The required time to converge to the global optimum is around 3 seconds for a centralized problem having a total number of decision variables of about 800 000.

The paper is organized as follows: In section II the considered centralized MPC problem is presented. Section III introduces the dual decomposition and the principle of the bundle method. In section IV the impact of the bundle methods'

[1] Schneider-Electric Industries. 37, quai Paul Louis Merlin, 38000 Grenoble, France. peter.pflaum@schneider-electric.com mohamed-yacine.lamoudi@schneider-electric.com

[2] CNRS-University of Grenoble, Gipsa-lab, BP 46, Domaine Universitaire, 38400 Saint Martin d'Hères , France. mazen.alamir@gipsa-lab.grenoble-inp.fr

parametrization on the computational efficiency is discussed in detail. In section V the proposed algorithm is applied in the context of smart districts where a limited amount of available energy is the shared resource in a district composed of 1000 buildings. Finally section VI concludes the paper.

II. Problem description

In this section the class of problems targeted by the proposed scheme is given. Namely the class of resource sharing problems is considered where several subsystems are coupled through a shared and limited resource.

A. Subsystem MPC problem

Consider a set of N_S dynamically uncoupled subsystems where each subsystem $l \in \mathscr{S} := \{1, ..., N_S\}$ obeys the general nonlinear dynamic equation:

$$x_{l,k+1} = f(x_{l,k}, u_{l,k}) \tag{1}$$

where $x_{l,k}$ and $u_{l,k}$ are the state and input vector of the subsystem l at instant k. In the sequel, given a vector quantity $v_l \in \mathbb{R}^{n_v}$ related to subsystem l, the boldfaced vector $\mathbf{v}_{l,k}$ represents the future profile of v_l over the prediction horizon of length N_p beginning at instant k, namely $\mathbf{v}_{l,k} := \left[v_{l,k}^T, ..., v_{l,k+N_p-1}^T\right]^T$. Note that when no ambiguity occurs the time index k is dropped.

For each subsystem $l \in \mathscr{S}$ the vector $r_{l,k} \in \mathbb{R}^{n_r}$ represents the vector of consumed resources where n_r stands for the number of different resources. The relation between the dynamics of system l and its consumed resources r_l over the prediction horizon is expressed through the following local equality constraint:

$$\mathbf{r}_{l,k} = \mathbf{h}_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k}) \tag{2}$$

Each subsystem $l \in \mathscr{S}$ is controlled by a local model predictive controller which is denoted hereafter by MPC_l. This is done by solving an optimization problem at each sampling instant k which is given by:

$$\text{MPC}_l : \underset{\mathbf{x}_{l,k} \in \mathscr{X}_{l,k}, \mathbf{u}_{l,k} \in \mathscr{U}_{l,k}}{\text{Minimize}} \quad L_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k}) \tag{3}$$

where $L_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k})$ is the objective function of subsystem l and $\mathscr{X}_{l,k}, \mathscr{U}_{l,k}$ denote the domains of the state and input constraints respectively.

B. Centralized MPC problem

Consider now a global limitation on the shared resource. It is expressed through the following inequality:

$$\mathbf{H}(\mathbf{r}_{1,k}, ..., \mathbf{r}_{N_S,k}) \le \mathbf{R}_{lim} \tag{4}$$

where $\mathbf{R}_{lim} \in \mathbb{R}^{n_r \cdot N_p}$ is the vector of the global resource limit over the prediction horizon and $\mathbf{H}(\mathbf{r}_{1,k}, ..., \mathbf{r}_{N_S,k})$ being linear in order to be decomposable as shown in the next section. For notational conciseness we will assume $\mathbf{H}(\mathbf{r}_{1,k}, ..., \mathbf{r}_{N_S,k}) = \sum\limits_{l \in \mathscr{S}} \mathbf{r}_{l,k}$ in the following.

The centralized MPC problem is then given by:

$$\begin{aligned} &\underset{\{\mathbf{x}_{l,k} \in \mathscr{X}_{l,k}, \mathbf{u}_{l,k} \in \mathscr{U}_{l,k}\}_{l \in \mathscr{S}}}{\text{Minimize}} && \sum_{l \in \mathscr{S}} L_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k}) \\ &\text{Subject to:} && \sum_{l \in \mathscr{S}} \mathbf{r}_{l,k} \le \mathbf{R}_{lim} \end{aligned} \tag{5}$$

For high numbers of subsystems N_S the centralized optimization problem becomes very large which may cause difficulties in handling the problem for the following reasons:

- non-scalability
- high computation times
- high communication requirements

To deal with these difficulties, hierarchical decomposition methods where the centralized problem is decomposed into several subproblems and one coordinator problem are very well suited. Two methods for decomposing the centralized problem stand out, namely primal and dual decomposition. In primal decomposition methods the coordinator problem is to directly determine the optimal distribution of the limited resource amongst the subsystems. In dual decomposition methods, the coordinator determines an optimal virtual price on the resource of interest such that the global limitation on the shared resource is respected.

A DMPC framework addressing resource sharing problems based on primal decomposition has been stated in [8]. In this paper we propose an approach based on dual decomposition. Common to both approaches is the bundle method which is used to efficiently solve the resulting coordinator problem based on the sub-gradient information of the subproblems.

III. The approach: Dual decomposition using bundle methods

A. Dual decomposition

In this section problem (5) is decomposed using the dual decomposition method. In the following the notation $\overline{v} := \{v_1, ..., v_{N_S}\}$ is used.

The dual problem is given by:

$$\begin{aligned} &\underset{\boldsymbol{\lambda}_k}{\text{Maximize}} \left[\inf_{\{\mathbf{x}_{l,k} \in \mathscr{X}_{l,k}, \mathbf{u}_{l,k} \in \mathscr{U}_{l,k}\}_{l \in \mathscr{S}}} \mathscr{L}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k, \boldsymbol{\lambda}_k) \right] \\ &\text{Subject to:} \quad \boldsymbol{\lambda}_k \ge 0 \end{aligned} \tag{6}$$

where $\mathscr{L}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k, \boldsymbol{\lambda}_k)$ is the Lagrangian given by:

$$\mathscr{L}(\overline{\mathbf{x}}_k, \overline{\mathbf{u}}_k, \boldsymbol{\lambda}_k) = \sum_{l \in \mathscr{S}} L_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k}) + \boldsymbol{\lambda}_k \cdot (\sum_{l \in \mathscr{S}} \mathbf{r}_{l,k} - \mathbf{R}_{lim}) \tag{7}$$

The vector $\boldsymbol{\lambda}_k \in \mathbb{R}^{n_r \cdot N_p}$ represents the so-called Lagrangian multipliers or dual variables.

Since the subsystems' dynamics are decoupled, the dual problem (6) can be decomposed into N_s subproblems (eq. (8)) which will be denoted as $\text{MPCdual}_l(\boldsymbol{\lambda}_k)$ and one coordinator problem (eq. (9)):

$$\text{MPCdual}_l(\boldsymbol{\lambda}_k) : \underset{\mathbf{x}_{l,k} \in \mathscr{X}_{l,k}, \mathbf{u}_{l,k} \in \mathscr{U}_{l,k}}{\text{Minimize}} \quad L_l(\mathbf{x}_{l,k}, \mathbf{u}_{l,k}) + \boldsymbol{\lambda}_k \cdot \mathbf{r}_{l,k} \tag{8}$$

Let $J_{l,k}(\boldsymbol{\lambda}_k) := L_l(\mathbf{x}^{\star}_{l,k}, \mathbf{u}^{\star}_{l,k}) + \boldsymbol{\lambda}_k \cdot \mathbf{r}^{\star}_{l,k}$ denote the achieved optimal value for a given dual variable $\boldsymbol{\lambda}_k$.

The coordinator problem becomes then:

$$\begin{aligned} &\underset{\boldsymbol{\lambda}_k}{\text{Maximize}} \sum_{l \in \mathscr{S}} J_{l,k}(\boldsymbol{\lambda}_k) - \boldsymbol{\lambda}_k \cdot \mathbf{R}_{lim} \\ &\text{Subject to:} \quad \boldsymbol{\lambda}_k \geq 0 \end{aligned} \tag{9}$$

Solving the coordinator problem (9) is not a trivial task, because the functions $J_{l,k}(\cdot)$ are not known at coordinator level and can only be evaluated point-wise for a given value of $\boldsymbol{\lambda}_k$ and at the computationally expensive price of solving all the subproblems and communicating their results to the coordinator. For this reason the efficiency of the solution strategy of the coordinator problem plays a crucial role. In the following subsection the bundle method which fulfills these requirements is introduced.

B. Solving the master problem - Bundle method

As mentioned above, the master solver does not have any information about the state and the shape of the subproblems objective functions $J_{l,k}(\cdot)$. In order to solve the master problem, i.e. to determine the optiamal virtual price $\boldsymbol{\lambda}_k$, the idea is to approximate the functions $\{J_{l,k}(\cdot)\}_{l \in \mathscr{S}}$ successively through an iterative process. More precisely, at each iteration the master affects a value of the dual variable $\boldsymbol{\lambda}_k$ to the subproblems, the subproblems are solved and their objective values $J_{l,k}(\boldsymbol{\lambda}_k)$ as well as their consumed resource vectors $\mathbf{r}_{l,k}$ are communicated to the master. The interesting point about the returned information is that according to eq. (8) the resource vector $\mathbf{r}_{l,k}$ is in fact the sub-gradient of the local objective function with respect to the dual variable: $\mathbf{r}_{l,k}(\boldsymbol{\lambda}_k) := \partial J_{l,k}(\boldsymbol{\lambda}_k)$. This sub-gradient interpretation of the resource vector $\mathbf{r}_{l,k}$ is the reason why bundle methods appear to be an appealing way to solve the master problem. The bundle method is based on iteratively approximating the cost function $J = \sum_{l \in \mathscr{S}} J_{l,k}(\boldsymbol{\lambda}_k) - \boldsymbol{\lambda}_k \cdot \mathbf{R}_{lim}$ by a so called *cutting plane model*. For more information regarding bundle methods the reader is referred to [4] or to [8] where this technique is applied to a similar DMPC problem based on a primal decomposition. In order to keep the notations as concise as possible the index k is dropped in this section since the iterative process takes place at a given instant k.

In the following the bundle method is described for its disaggregated version where for each subsystem's objective function $J_l(\boldsymbol{\lambda})$ an individual cutting plane approximation $\check{J}_l(\boldsymbol{\lambda})$ is stored at the master level. However it is also possible to approximate the sum of all subsystems $\sum_{l \in \mathscr{S}} J_l$ in a single cutting plane model (aggregated bundle) or even to divide the subsystems into groups of N_g subsystems where each group is represented by one cutting plane model (partially aggregated bundle). A detailed discussion of the impact of this choice is provided in the following section.

For each subsystem a cutting plane approximation of its objective function $\{J_l\}_{l \in \mathscr{S}}$ is stored in a memory $\mathfrak{B}_l^{(s)}$ which is updated at each iteration s and defined as follows:

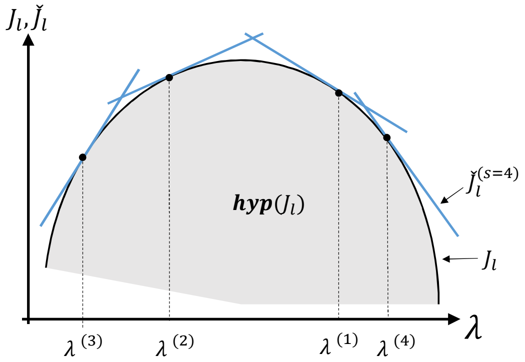

Fig. 1: Objective function J_l and its piece-wise linear approximation $\check{J}_l(\boldsymbol{\lambda})$ after 4 iterations.

$$\mathfrak{B}_l^{(s)} := \{\mathbf{g}_l^{(i)}, \epsilon_l^{(i)}\}_{i=1,\ldots,n_{\mathfrak{B}}} \tag{10}$$

The bundle $\mathfrak{B}_l^{(s)}$ behaves like a FIFO register that keeps only the last $n_{\mathfrak{B}}$ elements in memory. At every iterate s the last element ($i = n_{\mathfrak{B}}$) in the bundle is forgotten and the first element ($i = 1$) is updated by:

$$\begin{aligned} \mathbf{g}_l^{(1)} &\leftarrow \mathbf{r}_l \\ \epsilon_l^{(1)} &\leftarrow J_l(\boldsymbol{\lambda}^{(s)}) - \langle \mathbf{r}_l^{(s)}, \boldsymbol{\lambda}^{(s)} \rangle \end{aligned} \tag{11}$$

where $J_l(\boldsymbol{\lambda}^{(s)})$ and $\mathbf{r}_l^{(s)}$ are respectively the objective function value and the optimal resource profile computed by subsystem l for the given dual variable $\boldsymbol{\lambda}^{(s)}$ at iteration s. Based on the bundles $\mathfrak{B}_l^{(s)}$ the cutting plane approximations $\check{J}_l^{(s)}(\cdot)$ are defined as:

$$\check{J}_l^{(s)}(\cdot) = \min_{i=1,\ldots,n_{\mathfrak{B}}} \langle \mathbf{g}_l^{(i)}, \cdot \rangle + \epsilon_l^{(i)} \tag{12}$$

Every linear piece $\langle \mathbf{g}_l^{(i)}, \cdot \rangle + \epsilon_l^{(i)}$ defines a half space and is a supporting hyperplane of the hypograph $\mathbf{hyp}(J_l)$ of the function J_l. Since J_l is concave each hyperplane stored in the bundle $\mathfrak{B}_l^{(s)}$ constitutes a global over-estimator of J_l as illustrated in figure 1. Replacing the objective function terms of the subsystems J_l by their approximations $\check{J}_l^{(s)}$ and adding a regularization term consisting of a weighted trust region around the current best solution $\bar{\boldsymbol{\lambda}}^{(s)}$ in problem (9), the master problem that has to be solved at each iteration s finally becomes:

$$\begin{aligned} \boldsymbol{\lambda}^{(s+1)} := \underset{\boldsymbol{\lambda}}{\text{Maximize}} &\sum_{l \in \mathscr{S}} \check{J}_l^{(s)}(\boldsymbol{\lambda}) - \boldsymbol{\lambda} \cdot \mathbf{R}_{lim} + \mu \left\| \boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}^{(s)} \right\|_{L2} \\ \text{Subject to:} \quad &|\boldsymbol{\lambda} - \bar{\boldsymbol{\lambda}}^{(s)}| \leq \rho \\ &\geq 0 \end{aligned} \tag{13}$$

where the variable parameter ρ is the trust region size around the current best solution $\bar{\boldsymbol{\lambda}}^{(s)}$ (the so-called central point) and μ is the weight on the L_2-norm of the distance from the central point $\bar{\boldsymbol{\lambda}}^{(s)}$.

The updating rule for the central point $\bar{\boldsymbol{\lambda}}^{(s)}$ and for the trust region size ρ from one iteration to another is pretty straightforward: If the objective value $J(\boldsymbol{\lambda}^{(s+1)})$ computed by equation (9) is greater than the objective value at the central point $J(\bar{\boldsymbol{\lambda}}^{(s)})$ then the central point is updated as $\bar{\boldsymbol{\lambda}}^{(s+1)} \leftarrow \boldsymbol{\lambda}^{(s+1)}$, otherwise it remains unchanged. The trust region parameter ρ is increased in case the objective value was improved and decreased otherwise. Note that also if no improvement was achieved, the precision of the approximation is still improved due to the new elements that were added to the bundles.

IV. Algorithm efficiency evaluation

In this section we discuss difficulties arising when the proposed method is applied to large-scale problems. An additional aspect in the scope of this section is a comparison with the DMPC framework proposed in [8] which is based on a primal decomposition of the centralized problem and solved by a disaggregated version of the bundle method.

A. Difference between primal and dual master problem

In the following the main structural differences between the primal and the dual decomposition approaches are pointed out.

- The number of decision variables at the coordinator level are increasing much stronger with the number of subsystems N_S in the primal decomposition approach. This results in a potentially higher computation time for the primal master problem for high values of N_S:
$$\begin{aligned} \text{primal:} \quad & N_{variables} = N_S \cdot N_p + N_S \\ \text{dual:} \quad & N_{variables} = N_p + N_S \end{aligned} \tag{14}$$
- While in the dual decomposition approach the consumed resources $\mathbf{r}_{l,k}$ of the subproblems are at the same time the gradients of the subproblems with respect to the virtual price vector, the subproblems' gradients $\mathbf{g}_{l,k}(\mathbf{r}_{l,k}) \in \partial J_{l,k}(\mathbf{r}_{l,k})$ for the primal decomposition approach are not available in such a natural way unless the problem is a LP or a QP (Quadratic Program).
- One important advantage of the primal decomposition approach is that at each iteration, the algorithm provides a feasible solution, since the shared resource constraint $\mathbf{H}(\mathbf{r}_{1,k}, ..., \mathbf{r}_{N_s,k}) \leq \mathbf{R}_{lim}$ is explicitly guaranteed by the master problem.

The time to solve the master problem has a crucial impact on the efficiency in solving the global optimization problem. In order to understand the computational complexity of the master problem, the following paragraph describes how the cutting plane approximations are implemented and solved as a Linear Program (LP). The following equation shows the LP structure representing the disaggregated cutting plane approximations of a problem with $N_S = 2$ or more precisely: $\underset{\lambda}{\text{Maximize}} \ \left[\check{J}_1(\lambda) + \check{J}_2(\lambda)\right]$.

$$\begin{aligned} &\underset{\lambda}{\text{Maximize}} \quad \eta_1 + \eta_2 \\ &\text{Subject to:} \begin{cases} \mathbf{g}_1^{(1)}\lambda & +\epsilon_1^{(1)} \geq \eta_1 \\ \vdots & \quad \vdots \\ \mathbf{g}_1^{(n_{\mathfrak{B}})}\lambda & +\epsilon_1^{(n_{\mathfrak{B}})} \geq \eta_1 \\ \mathbf{g}_2^{(1)}\lambda & +\epsilon_2^{(1)} \geq \eta_2 \\ \vdots & \quad \vdots \\ \mathbf{g}_2^{(n_{\mathfrak{B}})}\lambda & +\epsilon_2^{(n_{\mathfrak{B}})} \geq \eta_2 \end{cases} \end{aligned} \tag{15}$$

From this equation it becomes obvious that the number of constraints N_{ctr} in the master problem (and with it the computational effort to solve it) depends on the amount of subsystems N_S and the amount of cuts $n_{\mathfrak{B}}$ per cutting plane model. More precisely: $N_{ctr} = n_{\mathfrak{B}} \cdot N_S$. Since the time to solve the master problem increases quasi-exponentially with the amount of constraints (as confirmed through simulations which are shown in figure 2), the computational efficiency of the disaggregated bundle method becomes very bad for high numbers of subsystems N_S due to the increasing time t_{Master} needed to solve the master problem. However this limitation can be overcome when using aggregated or partially aggregated versions of the bundle method as detailed in the following section where the bundle method's impact on the scheme's computational efficiency is investigated.

B. Aggregated vs. disaggregated bundle method

In this section aggregated and partially aggregated bundle versions are recalled. Compared to the disaggregated bundle method they can strongly increase the whole scheme's computational efficiency, especially for high N_S. In the aggregated bundle method, one cutting plane model is used to approximate the sum of all subsystems' objective functions. This is obviously a less precise approximation and it usually results in a higher number of iterations as it is demonstrated in [7]. However the extremely reduced time t_{master} to solve the master problem at a given iteration makes it the more effective solution in terms of total computation time when N_S exceeds a certain number. The partially aggregated bundle method can be seen as a kind of trade-off between the fully aggregated and the disaggregated bundle versions. The idea is to build groups of several subsystems whose objective functions are then approximated by a common cutting plane model. This way a higher precision in the approximation of the subsystems' objective functions is maintained and consequently the necessary amount of iterations reduced, while still strongly decreasing the computational effort of the master problem. The following equation provides the resulting number of constraints N_{ctr} in the master problem:

$$N_{ctr} = n_{\mathfrak{B}} \cdot \frac{N_S}{N_g} \tag{16}$$

where N_g is the number of subsystems whose objective functions are approximated in one cutting plane model.

C. Convergence time

DMPC approaches require an important number of iterations compared to a centralized solution as discussed in [3]. Obviously the necessary number of iterations is one important factor to evaluate the efficiency of an algorithm. However when aiming to assess the real-time implementability of a DMPC framework a focus should not only be on the necessary iterations but also on computation time savings and on the impact of the communication infrastructure. Indeed in many situations it may be more advantageous to accept a higher number of iterations if the computational time per iteration can be reduced sufficiently. The following equation provides a way to estimate the total time a DMPC framework requires to converge:

$$t_{total} = n_{iter} \cdot (t_{Master} + t_{comm} + t_{Subsys}) \qquad (17)$$

where n_{iter} is the necessary number of iterations to converge to the global optimum, t_{Master} is the time to solve the master problem, t_{Subsys} is the time to solve one subproblem (note that they can be solved in parallel) and t_{comm} is the delay occurring at each iteration due to the communication time between the master and the subsystems.

It can be assumed that in equation (17) t_{comm} and t_{Subsys} are constant parameters. n_{iter} and t_{Master} however depend both on the two parameters $n_{\mathfrak{B}}$ and N_g of the bundle method, which makes it not obvious to determine the optimal trade-off for the bundle parametrization such that the total convergence time t_{total} of the scheme is minimal. However the qualitative impact of $n_{\mathfrak{B}}$ and N_g on the number of iterations n_{iter} and on the time to solve the master problem t_{Master} can be summarized as follows:

- n_{iter} becomes minimal when the approximation of the subsystems' objective functions in the master problem are sufficiently detailed. More precisely, a sufficient number of cuts $n_{\mathfrak{B}}$ memorized in each bundle and a small enough number N_g, meaning few subsystems being approximated in a common cutting plane model, result in a small n_{iter}.
- High values of $n_{\mathfrak{B}}$ and small values of N_g however result in an important number of constraints according to $N_{ctr} = n_{\mathfrak{B}} \cdot \frac{N_S}{N_g}$ in the master problem and consequently in a high t_{Master} (see figure 2). This means, there is a trade-off to be made between reducing n_{iter} and increasing t_{Master}.

Figure 3 qualitatively illustrates how the approximations' precision, i.e. the number of constraints N_{ctr}, affects the whole scheme's computational efficiency. Finding the optimal trade-off between $n_{\mathfrak{B}}$ and N_g for a specific problem however is problem-dependent and needs to be done through offline-simulations.

V. Application to smart district scenario

In this section application results of the proposed DMPC scheme in the context of a smart district are provided. We consider a district composed of 1000 buildings. At a given instant in time, the buildings' optimization problem is to guarantee the occupants' comfort over a prediction horizon of 24 hours while minimizing their energy bill. The building's optimization problems are subject to a coupling constraint, namely a limited total power supply in the district. In [10] the scenario is described in greater detail. A crucial point which is worth being mentioned here is that the comfort constraints requiring the indoor temperature to lie in a certain temperature envelop, is a relaxed constraint. More precisely, violations of the comfort bounds are penalized with a quadratic term in the buildings' objective functions. This way it is guaranteed that there is always a feasible solution to the global optimization problem, since in case of a severe power limit the indoor comfort would be deteriorated.

Figure 4 shows results of the optimal solution of the district

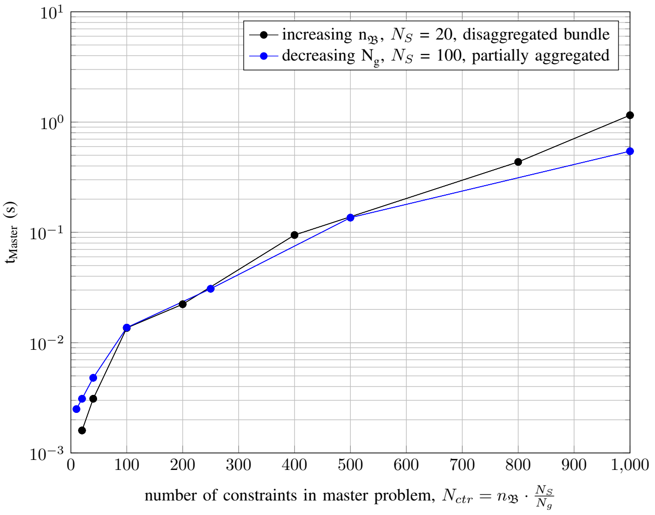

Fig. 2: This figure shows that the time t_{Master} to solve the master problem increases quasi exponentially with the number of constraints. It also becomes obvious that it hardly matters whether the number of constraints increases due to the number of cuts $n_{\mathfrak{B}}$ in each bundle or due to the number of subsystems' objective functions N_g being aggregated in one bundle.

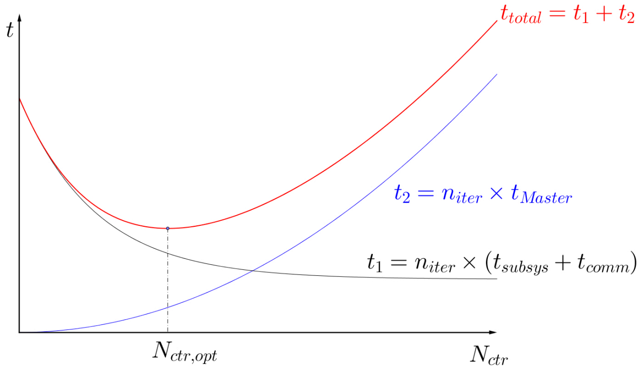

Fig. 3: Qualitative impact of the number of constraints $N_{ctr} = n_{\mathfrak{B}} \cdot \frac{N_S}{N_g}$ on the total efficiency of the DMPC framework. N_{ctr} corresponds to the precision of the cutting plane approximations of the subsystems' objective functions which is tuned through the two parameters $n_{\mathfrak{B}}$ and N_g.

problem for a 24 hour horizon and at a sampling period of 15 minutes. In order to achieve the best possible comfort under the restrictive power limit certain buildings are storing thermal energy during the night such that their consumption is reduced during the peak hour. Note that the corresponding centralized Quadratic Programming problem would have had around 800 000 variables. Solving such a centralized problem would clearly not be competitive compared to the distributed version.

Assuming a negligible communication time $t_{Comm} = 0$ and supposing that all subproblems can be solved in parallel, an estimation of the resulting computation time of the distributed optimization problem is around 3 seconds as computed by equation (18). The provided timings are obtained from simulations on an Intel(R) Core(TM) i7-3540M @ 3.00GHz using the gurobi solver (see [5]). Note however that in a realistic implementation of such a solution the communication time can become quite important, especially if the computation of the buildings' optimization problems would be performed on local controllers which are physically installed in the buildings. A more efficient solution in terms of computational efficiency might be one that relies on cloud computing.

$$\begin{aligned} t_{total} &= n_{iter} \cdot (t_{Master} + t_{comm} + t_{Subsys}) \\ &= 30 \cdot (25ms + 0 + 80ms) = 3150ms \end{aligned} \tag{18}$$

A. Assessing the scheme's efficiency in open loop

In this section the impact of variations in the parameters $n_{\mathfrak{B}}$ (number of cuts per bundle) and N_g (number of subsystems approximated in one bundle) on the global efficiency of the dual decomposition scheme in open loop is measured in simulations.

Figure 5 shows the convergence speed in terms of number of iterations for different numbers N_g of subsystems' objective functions being approximated in a single cutting plane model. The number of cuts per bundle is fixed to $n_{\mathfrak{B}} = 6$. From the results one can see that the expected effect of an increasing number of iterations for an increasing number N_g can not be observed for this specific problem. In fact, it does not matter in this example whether a disaggregated or a fully aggregated bundle version is used. Thus the best choice is simply the aggregated bundle method since the computation time of the master problem t_M is smallest in this case as it can be seen in the figure's legend.

Table I shows the impact of a varying number of cuts kept in the memory for each cutting plane model. For the different simulations the number of subsystems grouped in one bundle was fixed to $N_g = 50$. The results show that as long as more than one cut is stored in the bundle's memory (i.e. $n_{\mathfrak{B}} \geq 2$) the convergence speed does not show a clear dependency on $n_{\mathfrak{B}}$ for this specific problem. Thus the good choice is simply a small number of cuts $n_{\mathfrak{B}}$ per bundle, since this leads to the smallest computation time t_M for the master problem.

VI. Conclusion

In this paper a DMPC framework using a dual decomposition approach for solving large-scale resource sharing

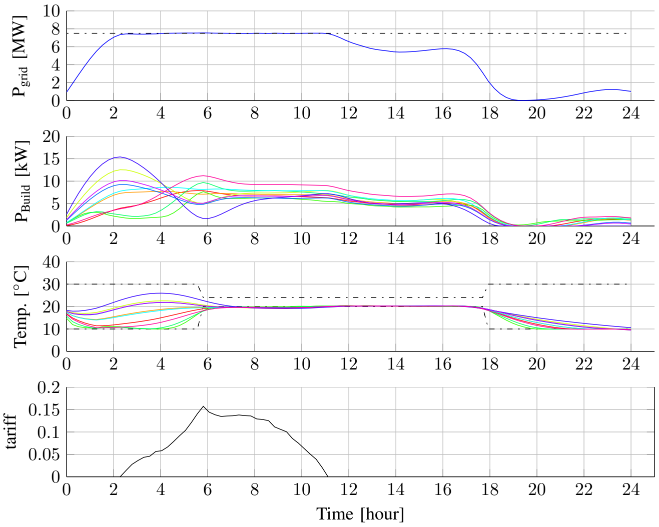

Fig. 4: Resulting optimal solution of a district composed of 1000 buildings for a 24 hour horizon and a sampling period of 15 minutes. The optimal power and temperature profiles of 10 randomly chosen buildings are provided in the second and third sub-figures. The first sub-figure shows the global power consumption of the district and that the global power limit is respected. Finally in the fourth sub-figure the dual variable having the nice property to act as an additional (virtual) tariff on the consumed energy is plotted.

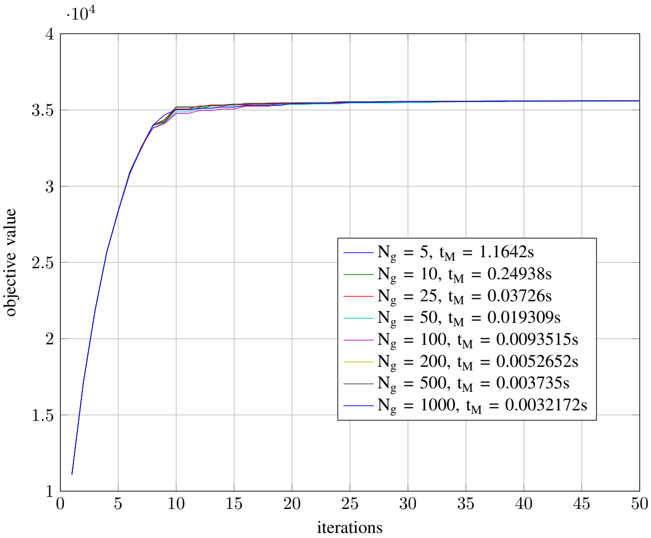

Fig. 5: Convergence speed in terms of number of iterations for different numbers of subsystems N_g aggregated in one cutting plane model. The fact that there is no visible difference for different N_g means that choosing a fully aggregated bundle method for this specific problem is always the best choice since in this case the time to solve the master problem t_M is shortest.

$n_{\mathfrak{B}}$	n_{iter}(99.5 %)	n_{iter}(99.75 %)	t_M (s)	N_{ctr}
1	41	49	0.0023	20
2	20	22	0.0037	40
4	18	22	0.0108	80
6	22	22	0.0150	120
8	27	27	0.0168	160
10	23	29	0.0233	200
30	23	30	0.1459	600

TABLE I: Convergence speed for different values of cuts $n_{\mathfrak{B}}$ kept in the memory of each bundle. N_g is fixed to 50. It turns out that as long as $n_{\mathfrak{B}} > 1$ there is no significant difference in the results. Consequently for this specific problem choosing a small value for $n_{\mathfrak{B}}$ is the best choice, since the time to solve the master problem t_M is smallest in this case.

problems has been developed. The core of the proposed scheme is the solution strategy of the master problem which relies on bundle methods.

The central contribution of this paper is a detailed discussion of the resulting efficiency of the scheme with respect to the parametrization of the bundle method. More precisely, the impact of using an aggregated, a partially aggregated or a disaggregated bundle version on the total convergence time of the scheme has been discussed.

In this context it has been shown that the efficiency of a DMPC framework should not only be assessed in terms of necessary iterations as it can often be observed in the literature, but in terms of total computation time taking also into account the communication infrastructure and the computation time at each iteration.

References

[1] *Distributed Model Predictive Control Made Easy*. Springer Netherlands, 2014, vol. 69.

[2] O. Briant, C. Lemarechal, P. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck, "Comparison of bundle and classical column generation," *Mathematical programming*, vol. 113, no. 2, pp. 299–344, 2008.

[3] M. Diehl, "Report of literature survey, analysis, and comparison of on-line optimization methods for hierarchical and distributed MPC," *Hierarchical and Distributed Model Predictive Control of Large-Scale Systems-Deliverable Number: 4.1.1/ Seventh Framework programme theme -ICT.*, 2009.

[4] A. Frangioni, "Generalized bundle methods," *SIAM Journal on Optimization*, vol. 13, no. 1, pp. 117–156, 2002.

[5] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2014. [Online]. Available: http://www.gurobi.com

[6] J. Kurose and R. Simha, "A microeconomic approach to optimal resource allocation in distributed computer systems," *Computers, IEEE Transactions on*, vol. 38, no. 5, pp. 705–717, May 1989.

[7] M. Y. Lamoudi, "Distributed model predictive control for energy management in buildings, p. 153 - 155," Ph.D. dissertation, University of Grenoble, 2012.

[8] M. Y. Lamoudi, M. Alamir, and P. Béguery, "A distributed-in-time NMPC-based coordination mechanism for resource sharing problems," in *Distributed Model Predictive Control Made Easy*, ser. Intelligent Systems, Control and Automation: Science and Engineering, J. M. Maestre and R. R. Negenborn, Eds. Springer Netherlands, 2014, vol. 69, pp. 147–162.

[9] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 8, pp. 1439–1451, 2006.

[10] P. Pflaum, M. Alamir, and M. Lamoudi, "Comparison of a primal and a dual decomposition for distributed MPC in smart districts," in *IEEE International Conference on Smart Grid Communications*, Nov. 2014.

[11] H. Scheu, J. Calderón, D. Doan, J. García, R. Negenborn, A. Tarău, F. Arroyave, B. De Schutter, J. Espinosa, and W. Marquardt, "Report on assessment of existing coordination mechanisms for simple case studies, and on possible options for improving and extending these coordination mechanisms," European FP7 STREP project HD-MPC, Deliverable D3.3.1, Dec. 2009.