RATE ANALYSIS OF INEXACT DUAL FIRST ORDER METHODS: APPLICATION TO DISTRIBUTED MPC FOR NETWORK SYSTEMS

ION NECOARA AND VALENTIN NEDELCU *

Abstract. In this paper we propose and analyze two dual methods based on inexact gradient information and averaging that generate approximate primal solutions for smooth convex optimization problems. The complicating constraints are moved into the cost using the Lagrange multipliers. The dual problem is solved by inexact first order methods based on approximate gradients and we prove sublinear rate of convergence for these methods. In particular, we provide, for the first time, estimates on the primal feasibility violation and primal and dual suboptimality of the generated approximate primal and dual solutions. Moreover, we solve approximately the inner problems with a parallel coordinate descent algorithm and we show that it has linear convergence rate. In our analysis we rely on the Lipschitz property of the dual function and inexact dual gradients. Further, we apply these methods to distributed model predictive control for network systems. By tightening the complicating constraints we are also able to ensure the primal feasibility of the approximate solutions generated by the proposed algorithms. We obtain a distributed control strategy that has the following features: state and input constraints are satisfied, stability of the plant is guaranteed, whilst the number of iterations for the suboptimal solution can be precisely determined.

Key words. Inexact dual gradient algorithms, parallel coordinate descent algorithm, rate of convergence, dual decomposition, estimates on suboptimality and infeasibility, distributed model predictive control.

1. Introduction.

Different problems from control and estimation can be addressed within the framework of network systems [17]. In particular, model predictive control (MPC) has become a popular advanced control technology implemented in network systems due to its ability to handle hard input and state constraints. Network systems are complex and large in dimension, whose structure may be hierarchical, multistage or dynamical and they have multiple decision-makers. Such systems can be broken down into smaller, more malleable subsystems called decompositions. How to consider the relationships between these various decompositions has led to much of the recent work within the general subject of the study of network systems.

Decomposition methods represent a powerful tool for solving distributed control, estimation and other engineering problems. The basic idea of these methods is to decompose the original large optimization problem into smaller subproblems which are then coordinated by a master problem. Decomposition methods can be divided into two main classes: primal and dual decomposition methods. In primal decomposition the optimization problem is solved using the original formulation and variables, while the complicating constraints are handled via methods such as interior point, penalty functions, feasible directions, Jacobi [4, 7, 10, 17, 28]. In dual decomposition the original problem is rewritten using Lagrangian relaxation and then solve the dual problem [1–3, 8, 18]. When the original problem is characterized by both simple and complicating constraints, dual decomposition may represent an appropriate choice since the complicating constraints can be moved into the cost using Lagrange multipliers and then the inner problems, that have simple constraints, are solved and the dual variables are updated with a Newton or (sub)gradient algorithm. Dual fast gradient methods based on exact first order information with provable guarantees on suboptimality are given in [18] for general convex problems and [23] for QP's. Dual

*The authors are with Automation and Systems Engineering Department, University Politehnica Bucharest, 060042 Bucharest, Romania. Corresponding author: I. Necoara, Tel. +40-21-4029195, Fax +40-21-4029195, Email ion.necoara@acse.pub.ro.

methods based on subgradient iteration and averaging, that produce primal solutions in the limit, can be found e.g. in [11,13,27]. Converge rate analysis for the dual subgradient method has been studied e.g. in [19], where the authors provide estimates of order $\mathcal{O}(1/\sqrt{k})$ for suboptimality and feasibility violation of the approximate solutions. Thus, an important drawback of the dual methods is that feasibility of the primal variables can be ensured only at optimality, which is usually impossible to attain in practice. However, in many applications, e.g. from control and estimation, the constraints can represent different requirements on physical limitation of actuators, safety limits and operating conditions of the controlled plant. Neglecting these constraints can reduce economic profit and cause damage to the environment or equipments. Therefore, any control or estimation scheme must ensure feasibility. Further, there is no convergence rate analysis in any of the existing literature for inexact dual (fast) gradient schemes. Thus, our goal is to develop inexact dual gradient algorithms which provide approximate primal solutions that are suboptimal and close to feasibility.

There are many ways to ensure feasibility of the primal variables in distributed MPC, e.g. through constraint tightening [4, 8, 12, 24] or distributed implementations of some classical methods such as the method of feasible directions, penalty functions, Jacobi and others [5, 7, 10, 15, 28]. In [8], a dual distributed algorithm for solving the MPC problem for systems with coupled dynamics and constraints is presented. The algorithm generates a primal feasible solution using primal averaging and constraint tightening. The Jacobi algorithm from [2] is used to update the primal variables, while the dual variables are updated using the subgradient method in [19]. The authors prove the convergence of the algorithm using the analysis of the dual subgradient method from [19] which has very slow convergence rate. In [12], the authors propose a decentralized MPC algorithm that uses the constraint tightening technique to achieve robustness while guaranteeing robust feasibility of the entire system. In [10, 24], distributed MPC algorithms for systems with coupled constraints is discussed. The approach divides the single large planning optimization into smaller subproblems, each planning only for the controls of a particular subsystem. Relevant plan data is communicated between subproblems to ensure that all decisions satisfy the coupled constraints. In [16, 28] cooperative based distributed MPC algorithms are proposed that converge to the centralized solution. In [15] a distributed MPC algorithm is proposed based on agent negotiation. In [4,5] distributed algorithms based on interior point or feasible directions are proposed that also converge to the centralized solution and guarantees primal feasibility. An iterative distributed model predictive control of large-scale nonlinear systems subject to asynchronous and delayed state feedback is discussed in [14]. See also [6, 17, 25] for recent surveys of distributed and hierarchical MPC methods. While most of the work cited above focuses on a primal approach, our work develops for the first time efficient dual methods that ensure constraint feasibility, tackles more general problems and more complex constraints and provides much better estimates on suboptimality.

Contribution. The contributions of the paper are as follows:

1. We propose and analyze novel dual algorithms with low complexity and fast rate of convergence that generate approximate primal solutions for large smooth convex problems.
2. We introduce a general framework for inexact first order information and then propose two inexact gradient methods for solving the dual (outer) problem:
   - an inexact dual gradient method, with rate of convergence of order

$\mathcal{O}(1/k)$.
- an inexact dual fast gradient method, with convergence rate of order $\mathcal{O}(1/k^2)$.

3. For both methods we provide for the first time a complete rate analysis and estimates on primal/dual suboptimality and feasibility violation of the generated approximate solutions.
4. In our schemes we solve the inner problems only up to a certain accuracy ϵ_{in} by means of a parallel coordinate descent method for which we prove linear rate of convergence.
5. For convex optimization models arising from distributed MPC problems, we adapt our algorithms using a tightening constraints approach, such that the convergence rates of the methods are preserved but in addition we are also able to ensure the primal feasibility.
6. To certify the complexity of the proposed methods, we apply the new algorithms on several linear distributed MPC problems with state and input constraints.

Paper outline. The paper is organized as follows. In Section 2 we introduce the dual problem of our original optimization problem formulated in Section 1.1. In Sections 2.2 and 2.3 we develop inexact dual gradient and fast gradient schemes for solving the outer problem and analyze their convergence rates. In Section 3 we propose a parallel coordinate descent method for solving the inner problems and prove its convergence rate. In Section 4 we first show how the distributed MPC problem for a network system can be recast in the form of our optimization model. Then, we combine the new dual algorithms with constraint tightening in order to ensure primal feasibility and stability. Finally, in Section 5 we provide extensive simulations in order to certify the efficiency of the newly developed algorithms.

Notation: We work in the space $\mathbb{R}^n$ composed by column vectors. For $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ we denote the standard Euclidean inner product $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^n \mathbf{u}_i \mathbf{v}_i$, norm $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ and projection onto non-negative orthant $\mathbb{R}^n_+$ as $[\mathbf{u}]_+$. We use $\langle \cdot, \cdot \rangle$, $\|\cdot\|$ and $[\cdot]_+$ for spaces of different dimension. For a real number α, $\lfloor \alpha \rfloor$ denotes the largest integer which is less than or equal to α. For any $\varepsilon \in [0, 1]$ we say that a quantity q is of order $\mathcal{O}(p(\varepsilon))$ if there exists $c > 0$ such that $q \leq cp(\varepsilon)$. Further, for a convex set $\mathbf{U}$, relint$(\mathbf{U})$ denotes the relative interior and $D_{\mathbf{U}}$ its diameter $D_{\mathbf{U}} = \max_{\mathbf{u},\mathbf{v} \in \mathbf{U}} \|\mathbf{u} - \mathbf{v}\|$. For a matrix $G \in \mathbb{R}^{p \times n}$, $\|G\|$ and $\|G\|_F$ denote the 2-norm and Frobenius norm, respectively.

1.1. Problem formulation. We are interested in solving the following large-scale smooth convex optimization problem:

$$F^* = \min_{\mathbf{u} \in \mathbf{U}} \{F(\mathbf{u}) : \quad h(\mathbf{u}) \leq 0\}, \tag{1.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ and the components of $h : \mathbb{R}^n \to \mathbb{R}^p$ are convex functions, and $\mathbf{U} \subseteq \mathbb{R}^n$ is a compact, convex set. Further, we assume that F and the components of h are twice differentiable. We also assume that the projection on the set defined by the complicating constraints (called also *coupling constraints*) $h(\mathbf{u}) \leq 0$ is hard to compute, but the set $\mathbf{U}$ is simple, i.e. the projection on this set can be computed very efficiently (e.g. hyperbox, Euclidean ball, etc).

In this paper we consider the following assumptions:

Assumption 1.1. (*i*) *Function F is σ_F-strongly convex w.r.t. $\|\cdot\|$ (see [21, Definition 2.1.2]).*

(*ii*) *The Jacobian of* h *is bounded on* $\mathbf{U}$*, i.e. there exists a constant* $c_h > 0$ *such that:*

$$\|\nabla h(\mathbf{u})\|_F \leq c_h \ \ \forall \mathbf{u} \in \mathbf{U}.$$

ASSUMPTION 1.2. *Slater condition holds for* (1.1)*, i.e. exists* $\tilde{\mathbf{u}} \in relint(\mathbf{U})$ *with* $h(\tilde{\mathbf{u}}) < 0$.

Note that as a consequence of Assumption (1.2), we have that strong duality holds for (1.1).

2. Solving the dual problem using inexact first order methods. Our goal is to solve the optimization problem (1.1) using dual gradient based methods. In order to update the dual variables we use inexact dual gradient methods (Sections 2.2 and 2.3), while the inner problems are solved up to a certain accuracy by means of a parallel coordinate descent algorithm (Section 3). An important feature of our algorithms consists of the fact that even if we use the inexact gradient of the dual function, after a certain number k_{out} of outer iterations, we are still able to compute a sequence of primal variables $\hat{\mathbf{u}}^{k_{\text{out}}}$ which are ϵ_{out}-optimal and their feasibility violation is also less than $\mathcal{O}(\epsilon_{\text{out}})$, i.e.:

$$\hat{\mathbf{u}}^{k_{\text{out}}} \in \mathbf{U}, \ \|[h(\hat{\mathbf{u}}^{k_{\text{out}}})]^+\| \leq \mathcal{O}(\epsilon_{\text{out}}) \ \text{ and } \ -\mathcal{O}(\epsilon_{\text{out}}) \leq F(\hat{\mathbf{u}}^{k_{\text{out}}}) - F^* \leq \mathcal{O}(\epsilon_{\text{out}}). \tag{2.1}$$

2.1. A framework for inexact first order information. We assume that the projection on $\mathbf{U}$ is simple but the projection on the set defined by the coupling constraints $h(\mathbf{u}) \leq 0$ is hard to compute. Therefore, we move the complicating constraints into the cost via Lagrange multipliers and define the dual function:

$$d(\lambda) = \min_{\mathbf{u} \in \mathbf{U}} \mathcal{L}(\mathbf{u}, \lambda), \tag{2.2}$$

where $\mathcal{L}(\mathbf{u}, \lambda) = F(\mathbf{u}) + \langle \lambda, h(\mathbf{u}) \rangle$ denotes the partial Lagrangian w.r.t. the complicating constraints $h(\mathbf{u}) \leq 0$. We also denote by $\mathbf{u}(\lambda)$ an optimal solution of the *inner problem*:

$$\mathbf{u}(\lambda) \in \arg\min_{\mathbf{u} \in \mathbf{U}} \mathcal{L}(\mathbf{u}, \lambda). \tag{2.3}$$

Based on Assumption 1.1 the gradient of the dual function $d(\lambda)$ is given by [2, Appendix A]:

$$\nabla d(\lambda) = h(\mathbf{u}(\lambda)).$$

The following lemma gives a characterization of the Lipschitz property for the gradient $\nabla d(\lambda)$:

LEMMA 2.1 (see Appendix). *Let the function* F *and the components of* h *be twice differentiable and Assumption 1.1 holds. Then, the gradient* $\nabla d(\lambda)$ *is Lipschitz continuous with constant:*

$$L_d = \frac{c_h^2}{\sigma_F}.$$

Under strong duality (see Assumption 1.2) we have for the *outer problem*:

$$F^* = \max_{\lambda \geq 0} d(\lambda), \tag{2.4}$$

for which we denote an optimal solution by λ^*. Since we cannot usually solve the inner optimization problem (2.3) exactly, but with some inner accuracy obtaining an approximate optimal solution $\bar{\mathbf{u}}(\lambda)$, we have to use inexact gradients and approximate values of the dual function d. Thus, we introduce the following two notions:

$$\bar{d}(\lambda) = \mathcal{L}(\bar{\mathbf{u}}(\lambda), \lambda) \text{ and } \bar{\nabla} d(\lambda) = h(\bar{\mathbf{u}}(\lambda)).$$

If we assume that $\bar{\mathbf{u}}(\lambda)$ is computed such that the following inner ϵ_{in}-optimality holds:

$$\bar{\mathbf{u}}(\lambda) \in \mathbf{U}, \;\; \mathcal{L}(\bar{\mathbf{u}}(\lambda), \lambda) - \mathcal{L}(\mathbf{u}(\lambda), \lambda) \le \frac{\epsilon_{\text{in}}}{3}, \tag{2.5}$$

then the next lemma provides bounds for the dual function $d(\lambda)$ in terms of a linear and a quadratic model which use only approximate information of the dual function and of its gradient.

LEMMA 2.2. *[9, Section 3.2] Let Assumptions 1.1 and 1.2 hold and for a given λ let $\bar{\mathbf{u}}(\lambda)$ be computed such that* (2.5) *is satisfied. Then, the following inequalities are valid:*

$$0 \ge d(\mu) - [\bar{d}(\lambda) + \langle \bar{\nabla} d(\lambda), \mu - \lambda \rangle] \ge -L_d \|\mu - \lambda\|^2 - \epsilon_{in} \quad \forall \mu \in \mathbb{R}^p_+. \tag{2.6}$$

Proof. For linear functions h, this lemma is proved in [9, Section 3.2] with stopping criterion $\epsilon_{\text{in}}/2$ in (2.5). For general convex functions h satisfying Assumption 1.1 (ii) we can easily show that $\|h(\mathbf{u}) - h(\mathbf{v})\| \le \sqrt{2} c_{\text{h}} \|\mathbf{u} - \mathbf{v}\|$ and then following exactlty the same steps as in [9] we get the result in (2.6). ∎

Remark 2.3 Relation (2.5) represents the stopping criterion for solving the inner problem (2.3). Many optimization methods offer direct control of this criterion (see e.g. the method of Section 3). For affine functions h (see e.g. MPC problems in Section 4), the stopping criterion in (2.5) can be taken as [9]: $\bar{\mathbf{u}}(\lambda) \in \mathbf{U}$ and $\mathcal{L}(\bar{\mathbf{u}}(\lambda), \lambda) - \mathcal{L}(\mathbf{u}(\lambda), \lambda) \le \frac{\epsilon_{\text{in}}}{2}$.

2.2. Inexact dual gradient method for solving the dual (outer) problem. In this section we analyze the convergence properties of an inexact dual projected gradient algorithm for solving approximately the dual problem (2.4). Let $\{\alpha^j\}_{j \ge 0}$ be a sequence of positive numbers and $S^k = \sum_{j=0}^k \alpha^j$. We consider the following inexact dual gradient algorithm:

Algorithm (IDG)(λ^0)
Given $\lambda^0 \in \mathbb{R}^p_+$, for $k \ge 0$ compute:
1. $\bar{\mathbf{u}}^k \approx \arg\min_{\mathbf{u} \in \mathbf{U}} \mathcal{L}(\mathbf{u}, \lambda^k)$ such that (2.5) holds
2. $\lambda^{k+1} = \left[\lambda^k + \alpha^k \bar{\nabla} d(\lambda^k)\right]_+$.

Recall that inexact gradient $\bar{\nabla} d(\lambda^k) = h(\bar{\mathbf{u}}^k)$ and $\alpha^k \in \left[\frac{1}{2\underline{L}}, \frac{1}{2L_{\text{d}}}\right]$ is a given step size with $\underline{L} \ge L_{\text{d}}$. The following theorem provides an estimate on the dual suboptimality for algorithm **(IDG)**:

THEOREM 2.4. *Let Assumptions 1.1 and 1.2 hold and the sequences $\left(\bar{\mathbf{u}}^k, \lambda^k\right)_{k \ge 0}$ be generated by algorithm* **(IDG)** *and define the average sequence of dual variables $\hat{\lambda}^k = \frac{1}{S^k} \sum_{j=0}^k \alpha^j \lambda^{j+1}$. Then, the following estimate on dual suboptimality can be derived for dual problem* (2.4)*:*

$$F^* - d(\hat{\lambda}^k) \le \frac{\underline{L} R_{\text{d}}^2}{k+1} + \epsilon_{in}, \tag{2.7}$$

where we define:

$$R_{\mathrm{d}} = \|\lambda^* - \lambda^0\|.$$

Proof. Let us first notice that the update of the dual variables can be equivalently written as $\lambda^{k+1} = \arg\min_{\lambda \geq 0} \left[\frac{1}{2\alpha^k} \left\|\lambda - \lambda^k\right\|^2 - \left\langle \bar{\nabla} d(\lambda^k), \lambda - \lambda^k \right\rangle\right]$, for which the optimality condition reads:

$$\left\langle \lambda^{k+1} - \lambda^k - \alpha^k \bar{\nabla} d(\lambda^k), \lambda - \lambda^{k+1} \right\rangle \geq 0 \quad \forall \lambda \geq 0. \tag{2.8}$$

If we now define $r_\lambda^j = \|\lambda^j - \lambda\|^2$ for any $\lambda \geq 0$, then we have:

$$\begin{aligned} r_\lambda^{j+1} &= \|\lambda^{j+1} - \lambda^j + \lambda^j - \lambda\|^2 = r_\lambda^j + 2\langle \lambda^{j+1} - \lambda^j, \lambda^j - \lambda^{j+1} + \lambda^{j+1} - \lambda\rangle + \|\lambda^{j+1} - \lambda^j\|^2 \\ &= r_\lambda^j + 2\langle \lambda^{j+1} - \lambda^j, \lambda^{j+1} - \lambda\rangle - \|\lambda^{j+1} - \lambda^j\|^2 \\ &\overset{(2.8)}{\leq} r_\lambda^j - 2\alpha^j \langle \bar{\nabla} d(\lambda^j), \lambda - \lambda^j\rangle + 2\alpha^j \left[\langle \bar{\nabla} d(\lambda^j), \lambda^{j+1} - \lambda^j\rangle - L_{\mathrm{d}} \|\lambda^{j+1} - \lambda^j\|^2\right] \\ &\overset{(2.6)}{\leq} r_\lambda^j + 2\alpha^j \left[\bar{d}(\lambda^j) - d(\lambda)\right] + 2\alpha^j \left[d(\lambda^{j+1}) - \bar{d}(\lambda^j) + \epsilon_{\mathrm{in}}\right] \\ &= r_\lambda^j + 2\alpha^j [d(\lambda^{j+1}) - d(\lambda) + \epsilon_{\mathrm{in}}] \quad \forall \lambda \geq 0, \end{aligned} \tag{2.9}$$

where in the first inequality we use the fact that $\alpha^j \leq \frac{1}{2L_{\mathrm{d}}}$. Summing up these inequalities for $j = 0, \dots, k$ and using the definition of $\hat{\lambda}^k$ we can write:

$$2S^k \left[d(\lambda) - d(\hat{\lambda}^k)\right] \leq r_\lambda^0 + 2S^k \epsilon_{\mathrm{in}} \quad \forall \lambda \geq 0.$$

Letting now $\lambda = \lambda^*$, dividing both sides of the previous inequality by $2S^k$ and taking into account that $S^k \geq \frac{k+1}{2\underline{L}}$ we obtain (2.7). ☐

We can observe that the first term in the estimate (2.7) represents the standard rate of convergence of the gradient method for the class of smooth functions [21]. Also, the second term ϵ_{in} is the error induced by the fact that the gradient is computed only approximately and shows that algorithm **(IDG)** does not accumulate errors.

However, we are now interested in finding estimates for primal feasibility violation for original problem (1.1). Let us introduce the following average primal sequence:

$$\hat{\mathbf{u}}^k = \frac{1}{S^k} \sum_{j=0}^{k} \alpha^j \bar{\mathbf{u}}^j. \tag{2.10}$$

The following theorem provides an estimate on primal feasibility violation for problem (1.1):

THEOREM 2.5. *Under the assumptions of Theorem 2.4 and with* $\hat{\mathbf{u}}^k$ *defined in* (2.10)*, the following estimate on primal feasibility violation can be derived for the original problem* (1.1)*:*

$$\|[h(\hat{\mathbf{u}}^k)]_+\| \leq v(k, \epsilon_{in}) \ \ \forall k \geq 0, \tag{2.11}$$

where $v(k, \epsilon_{in}) = \frac{4\underline{L} R_{\mathrm{d}}}{k+1} + \frac{6\underline{L}\|\lambda^0\|}{k+1} + 2\sqrt{\frac{\underline{L}}{k+1}\epsilon_{in}}$.

Proof. Using the definition of λ^{j+1} we have that the following component-wise inequalities hold: $\lambda^j + \alpha^j \bar{\nabla} d(\lambda^j) \leq \lambda^{j+1}$ for all $j \geq 0$. Summing up these inequalities for $j = 0, \dots, k$ and taking into account that $\bar{\nabla} d(\lambda^j) = h(\bar{\mathbf{u}}^j)$ we obtain: $\sum_{j=0}^{k} \alpha^j h(\bar{\mathbf{u}}^j) \leq \lambda^{k+1} - \lambda^0 \leq \lambda^{k+1}$, which together with the convexity of h gives: $h(\hat{\mathbf{u}}^k) \leq \frac{\lambda^{k+1}}{S^k}$. Since $\lambda^{k+1} \geq 0$ we also have that $0 \leq \left[h(\hat{\mathbf{u}}^k)\right]_+ \leq \frac{\lambda^{k+1}}{S^k}$ and thus we can further write:

$$\left\| \left[h(\hat{\mathbf{u}}^k)\right]_+ \right\| \leq \frac{\|\lambda^{k+1}\|}{S^k}. \tag{2.12}$$

Thus, in order to find an estimate on primal feasibility violation, we have to upper bound the norm of the dual sequence λ^{k+1}. For this purpose we can use (2.9) with $\lambda = \lambda^*$:

$$\|\lambda^{j+1} - \lambda^*\|^2 \leq \|\lambda^j - \lambda^*\|^2 + 2\alpha^j \left[d(\lambda^{j+1}) - d(\lambda^*) + \epsilon_{\text{in}}\right].$$

Summing up these inequalities for $j = 0, \dots, k$, using $\langle \lambda^0, \lambda^* \rangle \geq 0$ and $d(\lambda^{j+1}) \leq d(\lambda^*)$, we get:

$$\|\lambda^{k+1} - \lambda^*\|^2 \leq \|\lambda^*\|^2 + \|\lambda^0\|^2 + 2S^k \epsilon_{\text{in}},$$

Now, using the Cauchy-Schwartz inequality we get the second order inequality in $\|\lambda^{k+1}\|$:

$$\|\lambda^{k+1}\|^2 - 2\|\lambda^*\|\|\lambda^{k+1}\| - \|\lambda^0\|^2 - 2S^k \epsilon_{\text{in}} \leq 0.$$

Therefore, $\|\lambda^{k+1}\|$ must be less than the largest root of the corresponding second-order equation:

$$\begin{aligned} \|\lambda^{k+1}\| &\leq \frac{2\|\lambda^*\| + \left[4\|\lambda^*\|^2 + 4\|\lambda^0\|^2 + 8S^k \epsilon_{\text{in}}\right]^{1/2}}{2} \leq 2\|\lambda^*\| + \|\lambda^0\| + \sqrt{2S^k \epsilon_{\text{in}}} \\ &\leq 2\|\lambda^* - \lambda^0\| + 3\|\lambda^0\| + \sqrt{2S^k \epsilon_{\text{in}}}, \end{aligned}$$

where in the second inequality we used that $\sqrt{\zeta_1 + \zeta_2} \leq \sqrt{\zeta_1} + \sqrt{\zeta_2}$. Introducing this inequality in (2.12) and taking into account that $S^k \geq \frac{k+1}{2L}$ we obtain (2.11). □

Theorem 2.6. *Let the assumptions of Theorem 2.5 hold. Then, the following estimates on primal suboptimality can be derived for the original problem* (1.1)*:*

$$-\left(R_{\text{d}} + \|\lambda^0\|\right) v(k, \epsilon_{in}) \leq F(\hat{\mathbf{u}}^k) - F^* \leq \frac{L\|\lambda^0\|^2}{k+1} + \epsilon_{in}. \tag{2.13}$$

Proof. In order to prove the left-hand side inequality we can write:

$$\begin{aligned} F^* = d(\lambda^*) &= \min_{\mathbf{u} \in \mathbf{U}} F(\mathbf{u}) + \langle \lambda^*, h(\mathbf{u}) \rangle \leq F(\hat{\mathbf{u}}^k) + \langle \lambda^*, h(\hat{\mathbf{u}}^k) \rangle \\ &\leq F(\hat{\mathbf{u}}^k) + \left\langle \lambda^*, [h(\hat{\mathbf{u}}^k)]_+ \right\rangle \leq F(\hat{\mathbf{u}}^k) + \|\lambda^*\| \|[h(\hat{\mathbf{u}}^k)]_+\| \\ &= F(\hat{\mathbf{u}}^k) + \|\lambda^* - \lambda^0 + \lambda^0\| \|[h(\hat{\mathbf{u}}^k)]_+\| \leq F(\hat{\mathbf{u}}^k) + (R_{\text{d}} + \|\lambda^0\|) \|[h(\hat{\mathbf{u}}^k)]_+\|, \end{aligned}$$

which together with (2.11) lead to the result.

Now, we prove the right-hand side inequality. Taking $\lambda = 0$ in the first inequality of (2.9) we get:

$$\|\lambda^{j+1}\|^2 - 2\alpha^j \langle \bar{\nabla} d(\lambda^j), \lambda^j \rangle \leq \|\lambda^j\|^2 + 2\alpha^j \left[\langle \bar{\nabla} d(\lambda^j), \lambda^{j+1} - \lambda^j \rangle - L_{\mathrm{d}} \|\lambda^{j+1} - \lambda^j\|^2 \right]$$
$$\stackrel{(2.6)}{\leq} \|\lambda^j\|^2 + 2\alpha^j \left[d(\lambda^{j+1}) - \bar{d}(\lambda^j) + \epsilon_{\mathrm{in}} \right].$$

Taking into account that $\bar{\nabla} d(\lambda^j) = h(\bar{\mathbf{u}}^j)$ and using the definition of $\bar{d}(\lambda^j)$ we have: $-\langle \bar{\nabla} d(\lambda^j), \lambda^j \rangle = F(\bar{\mathbf{u}}^j) - \bar{d}(\lambda^j)$. Using this relation in the previous inequality we get:

$$\|\lambda^{j+1}\|^2 + 2\alpha^j \left[F(\bar{\mathbf{u}}^j) - \bar{d}(\lambda^j) \right] \leq \|\lambda^j\|^2 + 2\alpha^j \left[d(\lambda^{j+1}) - \bar{d}(\lambda^j) \right] + 2\alpha^j \epsilon_{\mathrm{in}}.$$

Summing up these inequalities for $j = 0, \dots, k$ and taking into account that F is convex and d concave, we obtain the following inequality:

$$2S^k \left[F(\hat{\mathbf{u}}^k) - d(\hat{\lambda}^k) \right] \leq \|\lambda^0\|^2 + 2S^k \epsilon_{\mathrm{in}}.$$

Dividing both sides of the previous inequality by S^k and using that $S^k \geq \frac{k+1}{2L}$ and $d(\hat{\lambda}^k) \leq F^*$, we obtain (2.13). ❑

Now, for a desired accuracy ϵ_{out} for solving problem (1.1), we are interested in finding the number of outer iterations k_{out} and a relation between ϵ_{out} and ϵ_{in} such that primal feasibility violation and suboptimality satisfy (2.1) and, moreover, the dual suboptimality will be also less than $\mathcal{O}(\epsilon_{\mathrm{out}})$. For simplicity, we consider the initial iterate $\lambda^0 = 0$ and thus $R_{\mathrm{d}} = \|\lambda^*\|$. Further, we consider a constant step size $\alpha^j = \frac{1}{2L_{\mathrm{d}}}$. Using Theorems 2.4, 2.5 and 2.6 we can take:

$$k_{\mathrm{out}} = \left\lfloor \frac{4L_{\mathrm{d}} R_{\mathrm{d}}^2}{\epsilon_{\mathrm{out}}} \right\rfloor \quad \text{and} \quad \epsilon_{\mathrm{in}} = \epsilon_{\mathrm{out}},$$

for which we obtain the following estimates for primal feasibility violation and suboptimality:

$$\hat{\mathbf{u}}^{k_{\mathrm{out}}} \in \mathbf{U}, \ \|[h(\hat{\mathbf{u}}^{k_{\mathrm{out}}})]_+\| \leq \frac{2}{R_{\mathrm{d}}} \epsilon_{\mathrm{out}},$$
$$-2\epsilon_{\mathrm{out}} \leq F(\hat{\mathbf{u}}^{k_{\mathrm{out}}}) - F^* \leq \epsilon_{\mathrm{out}} \quad \text{and} \quad F^* - d(\hat{\lambda}^{k_{\mathrm{out}}}) \leq \frac{5}{4} \epsilon_{\mathrm{out}}.$$

From the previous discussion it follows that in the algorithm **(IDG)** the inner problems (2.3) need to be solved with about the same accuracy as the desired accuracy of the outer problem, i.e. $\epsilon_{\mathrm{in}} = \epsilon_{\mathrm{out}}$ in the stopping criterion (2.5).

2.3. Inexact dual fast gradient method for solving the dual (outer) problem. In this section we discuss an inexact dual fast gradient scheme for updating the dual variable λ. A similar algorithm was proposed by Nesterov in [22] and applied further in [18] for solving dual problems with exact gradient information. An inexact version of the algorithm can be also found in [9]. The scheme defines two sequences $\left(\hat{\lambda}^k, \lambda^k \right)_{k \geq 0}$ for the dual variables:

Algorithm (IDFG)(λ^0)

Given $\lambda^0 \in \mathbb{R}^p_+$, for $k \geq 0$ compute:

1. $\bar{\mathbf{u}}^k \approx \arg\min_{\mathbf{u}\in\mathbf{U}} \mathcal{L}(\mathbf{u}, \lambda^k)$ such that (2.5) holds
2. $\hat{\lambda}^k = \left[\lambda^k + \frac{1}{2L_{\mathrm{d}}}\bar{\nabla} d(\lambda^k)\right]_+$
3. $\lambda^{k+1} = \frac{k+1}{k+3}\hat{\lambda}^k + \frac{2}{k+3}\left[\lambda^0 + \frac{1}{2L_{\mathrm{d}}}\sum_{s=0}^{k}\frac{s+1}{2}\bar{\nabla} d(\lambda^s)\right]_+$,

where we recall that $\bar{\nabla} d(\lambda^k) = h(\bar{\mathbf{u}}^k)$. Based on Theorem 4 in [9], which is an extension of the results in [18, 22] to the inexact case, we have the following result which will help us to establish upper bounds on primal and dual suboptimality and feasibility violation for our method.

Lemma 2.7. *[9, Theorem 4] If Assumptions 1.1 and 1.2 hold and the sequences* $\left(\bar{\mathbf{u}}^k, \hat{\lambda}^k, \lambda^k\right)_{k\geq 0}$ *are generated by algorithm* **(IDFG)***, then for all* $k \geq 0$ *we have:*

$$\frac{(k+1)(k+2)}{4} d(\hat{\lambda}^k) \geq \max_{\lambda\geq 0} -L_d\|\lambda - \lambda^0\|^2 + \sum_{s=0}^{k}\frac{s+1}{2}\left[\bar{d}(\lambda^s) + \langle\bar{\nabla} d(\lambda^s), \lambda - \lambda^s\rangle\right]$$

$$- \frac{(k+1)(k+2)(k+3)}{12}\epsilon_{in} \quad \forall \lambda \in \mathbb{R}^p_+. \tag{2.14}$$

The following theorem provides an estimate on the dual suboptimality for algorithm **(IDFG)**:

Theorem 2.8. *Let Assumptions 1.1 and 1.2 hold and the sequences* $\left(\bar{\mathbf{u}}^k, \hat{\lambda}^k, \lambda^k\right)_{k\geq 0}$ *be generated by algorithm* **(IDFG)***. Then, an estimate on dual suboptimality for* (2.4) *is given by:*

$$F^* - d(\hat{\lambda}^k) \leq \frac{4L_d R_{\mathrm{d}}^2}{(k+1)^2} + (k+1)\epsilon_{in}, \tag{2.15}$$

with R_{d} *defined as in Theorem 2.4.*

Proof. Using the first inequality from (2.6) in (2.14) we get:

$$\frac{(k+1)(k+2)}{4} d(\hat{\lambda}^k) \geq - L_{\mathrm{d}}\|\lambda^0 - \lambda^*\|^2 + \sum_{s=0}^{k}\frac{s+1}{2} d(\lambda^*) - \frac{(k+1)(k+2)(k+3)}{12}\epsilon_{\mathrm{in}}.$$

Dividing now both sides by $\frac{(k+1)(k+2)}{4}$, rearranging the terms and taking into account that $d(\lambda^*) = F^*$, $(k+1)^2 \leq (k+1)(k+2)$ and $(k+3)/3 \leq k+1$ we obtain (2.15). ☐

We can observe that the first term in the estimate (2.15) represents the standard rate of convergence of the fast gradient method for the class of smooth functions [21]. Also, the second term $(k+1)\epsilon_{\mathrm{in}}$ is the error induced by the fact that the gradient is computed only approximately and shows that algorithm **(IDFG)** accumulates the errors.

Further, we are interested now in finding estimates on primal feasibility violation and primal suboptimality for our original problem (1.1). For this purpose we define the following average sequence for the primal variables:

$$\hat{\mathbf{u}}^k = \sum_{s=0}^{k}\frac{2(s+1)}{(k+1)(k+2)}\bar{\mathbf{u}}^s. \tag{2.16}$$

The next result gives an estimate on primal feasibility violation.

THEOREM 2.9. *Under the assumptions of Theorem 2.8 and* $\hat{\mathbf{u}}^k$ *generated by* (2.16)*, an estimate on primal feasibility violation for original problem* (1.1) *is given by:*

$$\|[h(\hat{\mathbf{u}}^k)]_+\| \leq v(k, \epsilon_{in}), \tag{2.17}$$

where $v(k, \epsilon_{in}) = \frac{16 L_d R_\mathrm{d}}{(k+1)^2} + \frac{8 L_d \|\lambda^0\|}{(k+1)^2} + 4\sqrt{\frac{L_\mathrm{d}}{k+1}\epsilon_{in}}$.

Proof. Using (2.14), the convexity of F and h and taking into account that $(k+3)/3 \leq k+1$, we can write for any $\lambda \in \mathbb{R}^p_+$:

$$\max_{\lambda \geq 0} -\frac{4L_\mathrm{d}}{(k+1)^2}\|\lambda - \lambda^0\|^2 + \langle \lambda, h(\hat{\mathbf{u}}^k)\rangle \leq (k+1)\epsilon_\mathrm{in} + d(\hat{\lambda}^k) - F(\hat{\mathbf{u}}^k). \tag{2.18}$$

For the second term of the right-hand side we have:

$$\begin{aligned} d(\hat{\lambda}^k) - F(\hat{\mathbf{u}}^k) &\leq d(\lambda^*) - F(\hat{\mathbf{u}}^k) = \min_{\mathbf{u}\in\mathbf{U}} F(\mathbf{u}) + \langle \lambda^*, h(\mathbf{u})\rangle - F(\hat{\mathbf{u}}^k) \\ &\leq F(\hat{\mathbf{u}}^k) + \langle \lambda^*, h(\hat{\mathbf{u}}^k)\rangle - F(\hat{\mathbf{u}}^k) = \langle \lambda^*, h(\hat{\mathbf{u}}^k)\rangle \leq \langle \lambda^*, [h(\hat{\mathbf{u}}^k)]_+\rangle, \end{aligned} \tag{2.19}$$

where in the last inequality we used that $\lambda^* \geq 0$. By evaluating the left-hand side term in (2.18) at $\lambda = \frac{(k+1)^2}{8L_\mathrm{d}}\left[h(\hat{\mathbf{u}}^k)\right]_+$ and taking into account that $\langle [h(\hat{\mathbf{u}}^k)]_+, h(\hat{\mathbf{u}}^k) - [h(\hat{\mathbf{u}}^k)]_+\rangle = 0$ we obtain the following inequality:

$$\begin{aligned} \max_{\lambda\geq 0} -\frac{4L_\mathrm{d}}{(k+1)^2}\|\lambda - \lambda^0\|^2 + \langle \lambda, h(\hat{\mathbf{u}}^k)\rangle \geq& \frac{(k+1)^2}{16L_\mathrm{d}}\|[h(\hat{\mathbf{u}}^k)]_+\|^2 \\ &- \frac{4L_\mathrm{d}\|\lambda^0\|^2}{(k+1)^2} + \langle \lambda^0, [h(\hat{\mathbf{u}}^k)]_+\rangle. \end{aligned} \tag{2.20}$$

Combining now (2.19) and (2.20) with (2.18), using the Cauchy-Schwartz inequality and introducing the notation $\alpha = \|\left[h(\hat{\mathbf{u}}^k)\right]_+\|$, we obtain the following second order inequality in α:

$$\frac{(k+1)^2}{16L_\mathrm{d}}\alpha^2 - \|\lambda^* - \lambda^0\|\alpha - (k+1)\epsilon_\mathrm{in} - \frac{4L_\mathrm{d}\|\lambda^0\|^2}{(k+1)^2} \leq 0.$$

Therefore, α must be less than the largest root of the second-order equation, from which together with the definition of R_d and the identity $\sqrt{\zeta_1 + \zeta_2} \leq \sqrt{\zeta_1} + \sqrt{\zeta_2}$, we get the result. ☐

THEOREM 2.10. *Assume that the conditions in Theorem 2.9 are satisfied and let* $\hat{\mathbf{u}}^k$ *be given by* (2.16)*. Then, the following estimate on primal suboptimality for problem* (1.1) *can be derived:*

$$-\left(R_\mathrm{d} + \|\lambda^0\|\right) v(k, \epsilon_{in}) \leq F(\hat{\mathbf{u}}^k) - F^* \leq \frac{4L_d\|\lambda^0\|^2}{(k+1)^2} + (k+1)\epsilon_{in}. \tag{2.21}$$

Proof. The left-hand side inequality can be derived similarly as in the previous section (see the proof of Theorem 2.6). In order to prove the right-hand side inequality we use (2.18):

$$F(\hat{\mathbf{u}}^k) - d(\hat{\lambda}^k) \leq -\max_{\lambda \geq 0} -\frac{4L_{\mathrm{d}}}{(k+1)^2}\|\lambda - \lambda^0\|^2 + \langle \lambda, h(\hat{\mathbf{u}}^k)\rangle + \frac{k+3}{3}\epsilon_{\mathrm{in}}$$
$$\overset{\lambda=0}{\leq} \frac{4L_{\mathrm{d}}\|\lambda^0\|^2}{(k+1)^2} + \frac{k+3}{3}\epsilon_{\mathrm{in}}.$$

Taking now into account that $d(\hat{\lambda}^k) \leq F^*$ and $(k+3)/3 \leq k+1$ we get the result. ❑

Similar to the previous section, assume that we fix the outer accuracy to a desired value ϵ_{out}. We are interested in finding the number of outer iterations k_{out} and a relation between ϵ_{out} and ϵ_{in} such that primal feasibility violation and suboptimality satisfy (2.1). For simplicity, we again consider $\lambda^0 = 0$ and thus $R_{\mathrm{d}} = \|\lambda^*\|$. Using now Theorems 2.8, 2.9 and 2.10 we can take:

$$k_{\mathrm{out}} = \left\lfloor 2R_{\mathrm{d}}\sqrt{\frac{L_{\mathrm{d}}}{\epsilon_{\mathrm{out}}}}\right\rfloor \quad \text{and} \quad \epsilon_{\mathrm{in}} = \frac{\epsilon_{\mathrm{out}}\sqrt{\epsilon_{\mathrm{out}}}}{2R_{\mathrm{d}}\sqrt{L_{\mathrm{d}}}},$$

for which we obtain:

$$\hat{\mathbf{u}}^{k_{\mathrm{out}}} \in \mathbf{U}, \ \|[h(\hat{\mathbf{u}}^{k_{\mathrm{out}}})]_+\| \leq \frac{6}{R_{\mathrm{d}}}\epsilon_{\mathrm{out}} \ ,$$
$$-6\epsilon_{\mathrm{out}} \leq F(\hat{\mathbf{u}}^{k_{\mathrm{out}}}) - F^* \leq 2\epsilon_{\mathrm{out}} \ \text{ and } \ F^* - d(\hat{\lambda}^{k_{\mathrm{out}}}) \leq 3\epsilon_{\mathrm{out}}.$$

Note that for these choices of k_{out} and ϵ_{in} the inner problems (2.3) have to be solved with an accuracy of order $\mathcal{O}\left(\epsilon_{\mathrm{out}}\sqrt{\epsilon_{\mathrm{out}}}\right)$, i.e. $\epsilon_{\mathrm{in}} = \epsilon_{\mathrm{out}}\sqrt{\epsilon_{\mathrm{out}}}/(2R_{\mathrm{d}}\sqrt{L_{\mathrm{d}}})$ in (2.5). We can conclude that the **(IDFG)** method is more sensitive than the **(IDG)** method due to the error accumulation.

Remark 2.11 (*i*) Since in practice we usually cannot compute exactly the value $R_{\mathrm{d}} = \|\lambda^*\|$, we can use instead the following upper bound [19, Lemma 1]:

$$R_{\mathrm{d}} \leq \mathcal{R}_{\mathrm{d}} = \frac{F(\tilde{\mathbf{u}}) - d(\tilde{\lambda})}{\min_{1\leq j\leq p}\{-h^j(\tilde{\mathbf{u}})\}}, \tag{2.22}$$

where $\tilde{\mathbf{u}}$ denotes a Slater vector for problem (1.1) (see Assumption 1.2) and $\tilde{\lambda} \in \mathbb{R}^p_+$. The effects of this choice on the overall performance of the new algorithms are discussed in Section 5.1.

(*ii*) The results presented in Sections 2.2 and 2.3 also hold in the case when we solve the inner problems exactly, i.e. $\epsilon_{\mathrm{in}} = 0$ in (2.5), or when $\mathbf{U} = \mathbb{R}^n$, i.e. the inner problems are unconstrained.

(*iii*) Note that if $\lambda^0 = 0$ and we solve the inner problems exactly, i.e. $\epsilon_{\mathrm{in}} = 0$, then we have $F(\hat{\mathbf{u}}^{k_{\mathrm{out}}}) \leq F^*$, i.e. we are always below the optimal value in algorithms **(IDG)** and **(IDFG)**.

3. Solving the inner problem using a parallel coordinate descent method.

In this section we propose a block-coordinate descent based algorithm which permits to solve in parallel, for a fixed λ^k, the inner optimization problem (2.3):

$$\mathbf{u}^k = \arg\min_{\mathbf{u}\in\mathbf{U}} \mathcal{L}(\mathbf{u}, \lambda^k), \tag{3.1}$$

We consider for the variable $\mathbf{u}$ the partition $\mathbf{u} = \left[\mathbf{u}_1^T \dots \mathbf{u}_M^T\right]^T$ and the constraints set $\mathbf{U}$ can be represented in the form of a Cartesian product $\mathbf{U} = \mathbf{U}_1 \times \dots \times \mathbf{U}_M$, with $\mathbf{u}_i \in \mathbf{U}_i \subseteq \mathbb{R}^{n_i}$ being simple sets, i.e. the projection on these sets can be computed very efficiently. We also define the following partition of the identity matrix: $I = [E_1 \dots E_M] \in \mathbb{R}^{n \times n}$, where $E_i \in \mathbb{R}^{n \times n_i}$ for all $i = 1, \dots, M$, $n = \sum_{i=1}^M n_i$. Thus, $\mathbf{u}$ can be represented as: $\mathbf{u} = \sum_{i=1}^M E_i \mathbf{u}_i$.

Since for each outer iteration the dual variable λ^k is fixed, for the simplicity of the exposition we will drop the second argument of $\mathcal{L}$, i.e. we will use the notation $\mathcal{L}_k(\mathbf{u}) = \mathcal{L}(\mathbf{u}, \lambda^k)$. We will also denote by $\mathcal{L}_k^* = \mathcal{L}_k(\mathbf{u}^k)$ the optimal value of (3.1). We define the partial gradient of $\mathcal{L}_k$ at $\mathbf{u}$, denoted $\nabla_i \mathcal{L}_k(\mathbf{u}) \in \mathbb{R}^{n_i}$, as $\nabla_i \mathcal{L}_k(\mathbf{u}) = E_i^T \nabla \mathcal{L}_k(\mathbf{u})$ for all $i = 1, \dots, M$.

We consider the following assumption on the gradient of $\mathcal{L}_k$:

Assumption 3.1. *The gradient of $\mathcal{L}_k$ is coordinatewise Lipschitz continuous with constants $L_i > 0$, i.e. for all $i = 1, \dots, M$:*

$$\|\nabla_i \mathcal{L}_k(\mathbf{u} + E_i d_i) - \nabla_i \mathcal{L}_k(\mathbf{u})\| \leq L_i \|d_i\| \;\; \forall \mathbf{u} \in \mathbb{R}^n, d_i \in \mathbb{R}^{n_i}.$$

We recall that $\mathcal{L}_k(\mathbf{u}) = F(\mathbf{u}) + \langle \lambda_k, h(\mathbf{u}) \rangle$. Assumption 3.1 is valid for example if F has coordinatewise Lipschitz continuous gradient and the components of h are linear or convex quadratic functions. Note also that coordinatewise Lipschitz continuity also implies global Lipschitz continuity on extended space $\mathbb{R}^n$, with Lipschitz constant $\sum_{i=1}^{M} L_i$. Further, based on Assumption 1.1, since F is σ_{F}-strongly convex we have that $\mathcal{L}_k$ is also strongly convex (with a parameter $\sigma_{\mathcal{L}}$) w.r.t. the Euclidean norm. We also assume that $\mathbf{U}_i \subseteq \mathbb{R}^{n_i}$ are simple, compact, convex sets (e.g. hyperbox, Euclidean ball, entire space $\mathbb{R}^{n_i}$, etc). There exist many parallel algorithms in the literature for solving the optimization problem (3.1): e.g. Jacobi algorithms [2, 8], coordinate descent methods [28], etc. However, the rate of convergence for these algorithms is guaranteed under more conservative assumptions than the ones required for the parallel coordinate descent method proposed in this section.

Due to Assumption 3.1 we have [20, Section 2]:

$$\mathcal{L}_k(\mathbf{u} + E_i d_i) \leq \mathcal{L}_k(\mathbf{u}) + \langle \nabla_i \mathcal{L}_k(\mathbf{u}), d_i \rangle + \frac{L_i}{2} \|d_i\|^2 \;\; \forall \mathbf{u} \in \mathbb{R}^n, \; d_i \in \mathbb{R}^{n_i}, \; i = 1, \dots, M. \tag{3.2}$$

We introduce the following norm for the extended space $\mathbb{R}^n$:

$$\|\mathbf{u}\|_1^2 = \sum_{i=1}^{M} L_i \|\mathbf{u}_i\|^2, \tag{3.3}$$

which will prove useful for estimating the rate of convergence for our algorithm. Since $\mathcal{L}_k$ is $\sigma_{\mathcal{L}}$-strongly convex w.r.t. the Euclidean norm, it is also strongly convex w.r.t. $\|\cdot\|_1$ with parameter $\sigma_1 \leq \frac{\sigma_{\mathcal{L}}}{L_{\max}}$, where $L_{\max} = \max_{i=1,\dots,M} L_i$. Then, the following inequality holds [21]:

$$\mathcal{L}_k(\mathbf{w}) \geq \mathcal{L}_k(\mathbf{u}) + \langle \nabla \mathcal{L}_k(\mathbf{u}), \mathbf{w} - \mathbf{u} \rangle + \frac{\sigma_1}{2} \|\mathbf{w} - \mathbf{u}\|_1^2 \;\; \forall \mathbf{w}, \mathbf{u} \in \mathbb{R}^n \tag{3.4}$$

and combining it with (3.2) we can deduce that $\sigma_1 \leq 1$.

For solving the inner problem (3.1) we propose the following *parallel coordinate descent method*, which is similar to the algorithm from [28], but has much simpler iteration:

Algorithm **(PCD)**$(\mathbf{u}^{k,0})$
Given $\mathbf{u}^{k,0}$, for $l \geq 0$:
For $i = 1, \ldots, M$ compute in parallel
1. $\mathbf{v}_i^{k,l} = \left[\mathbf{u}_i^{k,l} - \frac{1}{L_i}\nabla_i \mathcal{L}_k(\mathbf{u}^{k,l})\right]_{\mathbf{U}_i}$
2. $\mathbf{u}_i^{k,l+1} = \frac{1}{M}\mathbf{v}_i^{k,l} + \frac{M-1}{M}\mathbf{u}_i^{k,l}$.

From the optimality conditions for $\mathbf{v}_i^{k,l}$ we get:

$$\left\langle \nabla_i \mathcal{L}_k(\mathbf{u}^{k,l}) + L_i(\mathbf{v}_i^{k,l} - \mathbf{u}_i^{k,l}), \mathbf{v}_i - \mathbf{v}_i^{k,l} \right\rangle \geq 0 \quad \forall \mathbf{v}_i \in \mathbf{U}_i. \tag{3.5}$$

Taking $\mathbf{v}_i = \mathbf{u}_i^{k,l}$ in (3.5) and combining with (3.2) and convexity of $\mathcal{L}_k$ we can conclude that algorithm **(PCD)** decreases the objective function at each inner iteration l:

$$\mathcal{L}_k(\mathbf{u}^{k,l+1}) \leq \mathcal{L}_k(\mathbf{u}^{k,l}) \quad \forall l \geq 0.$$

Remark 3.2 Note that if the sets $\mathbf{U}_i$ are simple and $\mathcal{L}_k$ has cheap coordinate derivatives, then computing $\mathbf{v}_i^{k,l}$ can be done numerically very efficient. For example, in case of hyperbox sets, the projection on $\mathbf{U}_i$ can be done in $\mathcal{O}(n_i)$ operations and if we also consider $\mathcal{L}_k$ to be quadratic, then the cost of computing $\nabla_i \mathcal{L}_k(\mathbf{u})$ is $\mathcal{O}(n \cdot n_i)$. Moreover, if its Hessian is sparse, then the cost of computing $\nabla_i \mathcal{L}_k(\mathbf{u})$ is usually much cheaper. Thus, for quadratic problems the worst case complexity per iteration of our method is $\mathcal{O}(n^2)$. Note that the complexity per iteration of the Jacobi type methods from [2, 8, 28] is at least $\mathcal{O}(n^2 + \sum_{i=1}^M n_i^3)$ provided that the local quadratic subproblems are solved with an interior point solver.

The following theorem provides the convergence rate of algorithm **(PCD)** and employs standard techniques for proving convergence of the projected gradient method [20, 21].

THEOREM 3.3. *Let Assumption 3.1 hold and $\mathcal{L}_k$ be σ_1-strongly convex w.r.t. $\|\cdot\|_1$. Then, the following linear rate of convergence is achieved for algorithm **(PCD)**:*

$$\mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* \leq \left(1 - \frac{2\sigma_1}{M(1+\sigma_1)}\right)^l \left(\frac{1}{2} r_u^0 + \mathcal{L}_k(\mathbf{u}^{k,0}) - \mathcal{L}_k^*\right),$$

where $r_u^0 = \|\mathbf{u}^{k,0} - \mathbf{u}^k\|_1^2$.

Proof. We introduce the following term: $r_u^l = \left\|\mathbf{u}^{k,l} - \mathbf{u}^k\right\|_1^2 = \sum_{i=1}^M L_i \left\langle \mathbf{u}_i^{k,l} - \mathbf{u}_i^k, \mathbf{u}_i^{k,l} - \mathbf{u}_i^k \right\rangle$, where we recall that $\mathbf{u}^k$ is the optimal solution of (3.1) and $\mathbf{u}_i^k = E_i^T \mathbf{u}^k$. Further, using (3.5) and similar derivations as in [20] we can write:

$$\begin{aligned} r_u^{l+1} &= \sum_{i=1}^M L_i \left\| \frac{1}{M}\mathbf{v}_i^{k,l} + (1 - \frac{1}{M})\mathbf{u}_i^{k,l} - \mathbf{u}_i^k \right\|^2 \\ &\leq r_u^l - \frac{2}{M}\sum_{i=1}^M \left(\frac{L_i}{2}\left\|\mathbf{v}_i^{k,l} - \mathbf{u}_i^{k,l}\right\|^2 + \left\langle \nabla_i \mathcal{L}_k(\mathbf{u}^{k,l}), \mathbf{v}_i^{k,l} - \mathbf{u}_i^{k,l} \right\rangle + \left\langle \nabla_i \mathcal{L}_k(\mathbf{u}^{k,l}), \mathbf{u}_i^k - \mathbf{u}_i^{k,l} \right\rangle \right). \end{aligned}$$

By convexity of $\mathcal{L}_k$ and (3.2) we obtain:

$$r_u^{l+1} \leq r_u^l - 2(\mathcal{L}_k(\mathbf{u}^{k,l+1}) - \mathcal{L}_k(\mathbf{u}^{k,l})) + \frac{2}{M}\left\langle \nabla \mathcal{L}_k(\mathbf{u}^{k,l}), \mathbf{u}^k - \mathbf{u}^{k,l} \right\rangle.$$

If we now take $\mathbf{w} = \mathbf{u}^k$ and $\mathbf{u} = \mathbf{u}^{k,l}$ in (3.4) and use the previous inequality we get:

$$\frac{1}{2}r_u^{l+1} + \mathcal{L}_k(\mathbf{u}^{k,l+1}) - \mathcal{L}_k^* \leq \frac{1}{2}r_u^l + \mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* - \frac{1}{M}(\mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* + \frac{\sigma_1}{2}r_u^l). \tag{3.6}$$

From the strong convexity of $\mathcal{L}_k$ in (3.4) we also get: $\mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* + \frac{\sigma_1}{2}r_u^l \geq \sigma_1 r_u^l$. We now define $\gamma = \frac{2\sigma_1}{1+\sigma_1} \in [0,1]$ and using the previous inequality we obtain:

$$\mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* + \frac{\sigma_1}{2}r_u^l \leq \gamma \left(\mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* + \frac{\sigma_1}{2}r_u^l\right) + (1-\gamma)\sigma_1 r_u^l.$$

Using this inequality in (3.6) we get:

$$\frac{1}{2}r_{l+1}^2 + \mathcal{L}_k(\mathbf{u}^{k,l+1}) - \mathcal{L}_k^* \leq \left(1 - \frac{\gamma}{M}\right)\left(\frac{1}{2}r_l^2 + \mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^*\right).$$

Applying this inequality iteratively, we obtain for $l \geq 0$:

$$\frac{1}{2}r_l^2 + \mathcal{L}_k(\mathbf{u}^{k,l}) - \mathcal{L}_k^* \leq \left(1 - \frac{\gamma}{M}\right)^l \left(\frac{1}{2}r_0^2 + \mathcal{L}_k(\mathbf{u}^{k,0}) - \mathcal{L}_k^*\right),$$

and by replacing $\gamma = \frac{2\sigma_1}{1+\sigma_1}$ we obtain the result. □

We can conclude from Theorem 3.3 that the number of inner iterations l_{in} which has to be performed such that stopping criterion (2.5) holds for an inner accuracy ϵ_{in} is given by [21]:

$$l_{\text{in}} = \left\lfloor \frac{M L_{\max}}{\sigma_{\mathcal{L}}} \ln \frac{3 L_{\max} D_{\mathbf{U}}^2}{\epsilon_{\text{in}}} \right\rfloor. \tag{3.7}$$

The output of algorithm **(PCD)** is $\bar{\mathbf{u}}^k = \mathbf{u}^{k,l_{\text{in}}}$. To conclude, we present now the following algorithmic framework for solving the original problem (1.1):

Algorithm (*Inexact dual (fast) gradient method*).
Initialization: Choose an outer accuracy ϵ_{out}.
Compute ϵ_{in} and k_{out} as in Sections 2.2 or 2.3.
Choose an initial point $\lambda^0 \in \mathbb{R}_+^p$.
Outer loop: For $k = 0, 1, \dots, k_{\text{out}}$, perform:
Step 1. **Inner loop**: For given λ^k, choose $\mathbf{u}^{k,0} \in \mathbf{U}$.
Compute l_{in} as in eq. (3.7).
For $l = 0, 1, \dots, l_{\text{in}}$ apply algorithm **(PCD)** to obtain $\bar{\mathbf{u}}^k = \mathbf{u}^{k,l_{\text{in}}}$.
Step 2. Compute the approximate gradient $\nabla \bar{d}(\lambda_k) = h(\bar{\mathbf{u}}^k)$.
Step 3. Update λ^{k+1} as in Alg. **(IDG)** or $(\lambda^{k+1}, \hat{\lambda}^k)$ as in Alg. **(IDFG)**.
Step 4. Update average sequences $(\hat{\mathbf{u}}^k, \hat{\lambda}^k)$.
Output: generated approximate primal-dual solutions $(\hat{\mathbf{u}}^k, \hat{\lambda}^k)$.

4. Distributed MPC problems for constrained network systems. In this section we apply the algorithms **(IDG)**, **(IDFG)** and **(PCD)** for solving in a distributed fashion MPC problems arising in network systems.

4.1. MPC formulation for network systems. We consider discrete-time network systems, which are usually modelled by a graph whose nodes represents subsystems and whose arcs indicate dynamic couplings between these subsystems, defined by the following linear state equations:

$$x_i(t+1) = \sum_{j \in \mathcal{N}^i} A_{ij} x_j(t) + B_{ij} u_j(t) \qquad \forall i = 1, \dots, M, \tag{4.1}$$

where M denotes the number of interconnected subsystems, $x_i(t) \in \mathbb{R}^{n_{x_i}}$ and $u_i(t) \in \mathbb{R}^{n_{u_i}}$ represent the state and the input of ith subsystem at time t, $A_{ij} \in \mathbb{R}^{n_{x_i} \times n_{x_j}}$ and $B_{ij} \in \mathbb{R}^{n_{x_i} \times n_{u_j}}$ and $\mathcal{N}^i$ denotes the neighbors of the ith subsystem including i. In a particular case frequently found in literature [15, 18, 28] the influence between neighboring subsystems is given only in terms of inputs:

$$x_i(t+1) = A_{ii} x_i(t) + \sum_{j \in \mathcal{N}^i} B_{ij} u_j(t). \tag{4.2}$$

We also impose local state and input constraints:

$$x_i(t) \in X_i, \quad u_i(t) \in U_i \qquad \forall i = 1, \dots, M, \;\; t \geq 0,$$

where $X_i \subseteq \mathbb{R}^{n_{x_i}}$ and $U_i \subseteq \mathbb{R}^{n_{u_i}}$ are simple convex sets. For a prediction horizon of length N, we consider quadratic stage and final costs for each subsystem i:

$$\sum_{t=0}^{N-1} \|x_i(t)\|^2_{Q_i} + \|u_i(t)\|^2_{R_i} + \|x_i(N)\|^2_{P_i},$$

where matrices Q_i, P_i and R_i are positive definite and $\|x\|^2_P = x^T P x$.

We now formulate the centralized MPC problem for (4.1), for a given initial state x:

$$\begin{aligned} F^*(x) = & \min_{x_i(t), u_i(t)} \sum_{i=1}^{M} \sum_{t=0}^{N-1} \|x_i(t)\|^2_{Q_i} + \|u_i(t)\|^2_{R_i} + \|x_i(N)\|^2_{P_i} \\ & \text{s.t.: } x_i(t+1) = \sum_{j \in \mathcal{N}^i} A_{ij} x_j(t) + B_{ij} u_j(t), \; x_i(0) = x_i, \\ & \qquad x_i(t) \in X_i, \; u_i(t) \in U_i, \; x_i(N) \in X_i^{\mathrm{f}} \quad \forall i = 1, \dots, M, \; t = 0, \dots N-1, \end{aligned} \tag{4.3}$$

where X_i^{f} are terminal sets chosen under some appropriate conditions to ensure stability of the MPC scheme (see e.g. [26]). For the input trajectory of subsystem i and the overall input trajectory we use the notations:

$$\mathbf{u}_i = \left[u_i(0)^T \dots u_i(N-1)^T\right]^T \in \mathbb{R}^{n_i}, \;\; \mathbf{u} = \left[\mathbf{u}_1^T \dots \mathbf{u}_M^T\right]^T \in \mathbb{R}^n.$$

We assume in addition that the local constraints sets U_i, X_i and the terminal sets X_i^{f} are polyhedral for all subsystems. An extension to general convex sets is straightforward and we omit it here due to space limitations. By eliminating the states from the dynamics (4.1), problem (4.3) can be expressed as a large-scale quadratic convex optimization problem of the form:

$$\begin{aligned} F^*(x) = & \min_{\mathbf{u}_1 \in \mathbf{U}_1, \dots, \mathbf{u}_M \in \mathbf{U}_M} \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + (\mathbf{W} x + \mathbf{w})^T \mathbf{u} \\ & \text{s.t.: } \;\; \mathbf{G}\mathbf{u} + \mathbf{E} x + \mathbf{g} \leq 0, \end{aligned} \tag{4.4}$$

where $\mathbf{H} \in \mathbb{R}^{n\times n}$ is positive definite due to the assumption that all R_i are positive definite and the inequalities $\mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \leq 0$, with $\mathbf{G} \in \mathbb{R}^{p\times n}$, are obtained by eliminating the states from the constraints $x_i(t) \in X_i$ and $x_i(N) \in X_i^{\mathrm{f}}$ for all i and t. If the projection on the input constraints set U_i is difficult, we can also move the input constraints in the complicating constraints $\mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \leq 0$. In this case $\mathbf{U}_i = \mathbb{R}^{n_i}$. Otherwise, i.e. the set U_i is simple (e.g. hyperbox), the convex set $\mathbf{U}_i = \prod_{t=1}^{N} U_i$. In MPC, at each time instant, given the initial state $x \in X_N$, where $X_N \subseteq \prod_{i=1}^{M} X_i$ is a region of attraction [26], we need to solve the optimization problem (4.3) or equivalently (4.4). We assume for (4.4) that for any $x \in X_N$ there exists a "strict Slater" vector $\tilde{\mathbf{u}}$, i.e. $\tilde{\mathbf{u}} \in \mathbf{U}$ and $\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g} < 0$.

In the following sections we discuss how we can solve the MPC problem (4.4) by combining the algorithms **(IDG)**, **(IDFG)** and **(PCD)** with tightening constraints techniques. We will derive estimates for the number of iterations required for finding a suboptimal feasible solution.

4.2. Tightening the coupling constraints. In many applications, like e.g. the MPC problem discussed above, the constraints may represent different requirements on physical limitation of actuators, safety limits and operating conditions of the controlled plant. Thus, ensuring the feasibility of the primal variables, i.e. $\mathbf{u} \in \mathbf{U}$ and $\mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \leq 0$, becomes a prerequisite. However, as we have seen in Sections 2.2 and 2.3, dual methods can ensure these requirements only at optimality, which is usually impossible to attain in practice. Therefore, in our approach, instead of solving the original problem (4.4), we consider a tightened problem (see also [8] for a similar approach where the tightened dual problem is solved using a subgradient algorithm with very slow convergence rate of order $\mathcal{O}\left(1/\sqrt{k}\right)$ and approximate solutions for the inner problems are computed using the Jacobi algorithm [2]).

We introduce the following tightened problem associated with the original problem (4.4):

$$F^*_{\epsilon_{\mathrm{c}}}(x) = \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) \quad \left(= \frac{1}{2}\, \mathbf{u}^T\mathbf{H}\mathbf{u} + (\mathbf{W}x + \mathbf{w})^T\mathbf{u}\right) \qquad \text{s.t.:} \quad \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} + \epsilon_{\mathrm{c}}\mathbf{e} \leq 0, \tag{4.5}$$

where $\mathbf{e} \in \mathbb{R}^p$ denotes the vector with all entries 1 and

$$0 < \epsilon_{\mathrm{c}} \leq \frac{1}{2} \min_{j=1,\dots,p} \{-\left(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g}\right)_j\}, \tag{4.6}$$

with $\tilde{\mathbf{u}}$ being a strict Slater vector for (4.4). Note that for this choice of ϵ_{c}, we have that $\tilde{\mathbf{u}}$ is also a strict Slater vector for the tightened problem (4.5). Similar to Section 2, for problem (4.5) we also denote by $\mathcal{L}_{\epsilon_{\mathrm{c}}}$ the partial Lagrangian w.r.t. the complicating constraints $\mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} + \epsilon_{\mathrm{c}}\mathbf{e} \leq 0$ and by $d_{\epsilon_{\mathrm{c}}}$ the corresponding dual function.

In the following sections we will see how we can ensure the feasibility, suboptimality and stability of the MPC scheme given in (4.3) based on the suboptimal input $\hat{\mathbf{u}}^{k_{\mathrm{out}}}$ obtained by solving the tightened problem (4.5) with the newly developed algorithms **(IDG)**/**(IDFG)** and **(PCD)**.

4.3. Feasibility and suboptimality of the MPC scheme. At each time instant of the MPC scheme, given the initial state x in the region of attraction X_N, instead of solving the optimization problem (4.4) we solve the tightened problem (4.5) using the algorithms **(IDG)** or **(IDFG)** for the outer problem and algorithm **(PCD)** for the inner problem. At each step we obtain a suboptimal input $\hat{\mathbf{u}}^{k_{\text{out}}}$ and according to the receding horizon strategy we apply to the system only the first input $\hat{\mathbf{u}}^{k_{\text{out}}}(0)$. However, we want that the generated control sequence $\hat{\mathbf{u}}^{k_{\text{out}}}$ to be suboptimal and feasible for the original MPC problem (4.4). Thus, we first need to find a relation between $F^*_{\epsilon_{\text{c}}}(x)$ and $F^*(x)$. Let us denote by $\lambda^*_{\epsilon_{\text{c}}}$ an optimal Lagrange multiplier for the inequality constraints in (4.5). The following upper bound can be established for any strict Slater vector $\tilde{\mathbf{u}}$ and dual multiplier $\tilde{\lambda} \in \mathbb{R}^p_+$:

$$\begin{aligned}
\|\lambda^*_{\epsilon_{\text{c}}}\| &\overset{(2.22)}{\leq} \frac{F(x,\tilde{\mathbf{u}}) - \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) + \left\langle \tilde{\lambda}, \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} + \epsilon_{\text{c}}\mathbf{e} \right\rangle}{\min\limits_{j=1,\dots,p} \{-\left(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g} + \epsilon_{\text{c}}\right)_j\}} \\
&= \frac{\left[F(x,\tilde{\mathbf{u}}) - \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) + \left\langle \tilde{\lambda}, \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \right\rangle\right] - \left\langle \tilde{\lambda}, \epsilon_{\text{c}}\mathbf{e} \right\rangle}{\min\limits_{j=1,\dots,p} \{-\left(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g}\right)_j\} - \epsilon_{\text{c}}} \\
&\leq 2\mathcal{R}_{\text{d}} \qquad \forall x \in X_N, \qquad (4.7)
\end{aligned}$$

where in the last inequality we used (4.6) and the fact that both $\tilde{\lambda}$ and ϵ_{c} are nonnegative. Taking into account that $\{\mathbf{u} : \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} + \epsilon_{\text{c}}\mathbf{e} \leq 0\} \subseteq \{\mathbf{u} : \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \leq 0\}$ we have:

$$F^*_{\epsilon_{\text{c}}}(x) \geq F^*(x) \qquad \forall x \in X_N. \qquad (4.8)$$

On the other hand, from the dual formulation of the tightened problem (4.5) we have:

$$\begin{aligned}
F^*_{\epsilon_{\text{c}}}(x) &= \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) + \left\langle \lambda^*_{\epsilon_{\text{c}}}, \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} + \epsilon_{\text{c}}\mathbf{e} \right\rangle \\
&= \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) + \left\langle \lambda^*_{\epsilon_{\text{c}}}, \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \right\rangle + \left\langle \lambda^*_{\epsilon_{\text{c}}}, \epsilon_{\text{c}}\mathbf{e} \right\rangle \qquad (4.9) \\
&\leq \max_{\lambda\geq 0} \min_{\mathbf{u}\in\mathbf{U}} F(x,\mathbf{u}) + \langle \lambda, \mathbf{G}\mathbf{u} + \mathbf{E}x + \mathbf{g} \rangle + \sqrt{p}\epsilon_{\text{c}}\|\lambda^*_{\epsilon_{\text{c}}}\| \leq F^*(x) + 2\sqrt{p}\mathcal{R}_{\text{d}}\epsilon_{\text{c}}.
\end{aligned}$$

We will further see how we can use relations (4.8) and (4.9) to recover the primal suboptimality for the original problem (4.4) from the suboptimality of the tightened problem (4.5), based on the results from Section 2. We now discuss the suboptimality and the feasibility of the MPC scheme based on the algorithms **(IDG)** and **(IDFG)**.

For the algorithm **(IDG)** we assume that the outer accuracy ϵ_{out} is chosen such that:

$$\epsilon_{\text{out}} \leq \left(\sqrt{p} + 0.05\right) \mathcal{R}_{\text{d}} \min_{j=1,\dots,p} \{-\left(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g}\right)_j\}.$$

Based on the results stated in Section 2.2 and relations (4.8) and (4.9) we can choose, for example, the following values for the number of outer iterations k_{out}, the inner accuracy ϵ_{in} and also for the tightening parameter ϵ_{c}:

$$\begin{aligned}
k_{\text{out}} &= \left\lfloor \frac{10\left(2\sqrt{p} + 0.1\right) L_{\text{d}}\mathcal{R}^2_{\text{d}}}{\epsilon_{\text{out}}} \right\rfloor \qquad (4.10) \\
\epsilon_{\text{in}} &= \frac{\epsilon_{\text{out}}}{20\left(2\sqrt{p} + 0.1\right)}, \quad \epsilon_{\text{c}} = \frac{\epsilon_{\text{out}}}{\left(2\sqrt{p} + 0.1\right)\mathcal{R}_{\text{d}}}.
\end{aligned}$$

Using the previous choices for $k_{\rm out}$, $\epsilon_{\rm in}$ and $\epsilon_{\rm c}$ in Theorem 2.5 we have:

$$\left\| \left[\mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} + \epsilon_{\rm c}\mathbf{e}\right]_+ \right\| \leq \frac{8L_{\rm d}\mathcal{R}_{\rm d}}{k_{\rm out}+1} + 2\sqrt{\frac{L_{\rm d}}{k_{\rm out}+1}\epsilon_{\rm in}} < \epsilon_{\rm c},$$

which implies that for all $j = 1, \dots, p$, we can write:

$$\left[(\mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} + \epsilon_{\rm c})_j\right]_+ < \epsilon_{\rm c}.$$

Since $(\mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} + \epsilon_{\rm c})_j \leq \left[(\mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} + \epsilon_{\rm c})_j\right]_+$ we have that $\hat{\mathbf{u}}^{k_{\rm out}} \in \mathbf{U}$ and $\mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} < 0$ and thus algorithm **(IDG)** guarantees feasibility of the primal variable $\hat{\mathbf{u}}^{k_{\rm out}}$. Further, using now Theorem 2.6 together with (4.8) and (4.9) we have that $-\frac{\epsilon_{\rm out}}{\sqrt{p}} \leq F(x, \hat{\mathbf{u}}^{k_{\rm out}}) - F^*(x) \leq \epsilon_{\rm out}$ and since $\hat{\mathbf{u}}^{k_{\rm out}}$ is feasible, we get:

$$0 \leq F(x, \hat{\mathbf{u}}^{k_{\rm out}}) - F^*(x) \leq \epsilon_{\rm out}$$

and thus the MPC scheme based on algorithm **(IDG)** is also $\epsilon_{\rm out}$-suboptimal.

In order to prove the suboptimality and feasibility of the MPC scheme based on algorithm **(IDFG)** we proceed in a similar way as for algorithm **(IDG)**. We assume that the outer accuracy $\epsilon_{\rm out}$ is chosen such that:

$$\epsilon_{\rm out} \leq (\sqrt{p} + 0.5)\,\mathcal{R}_{\rm d} \min_{j=1,\dots,p} \{-(\mathbf{G}\tilde{\mathbf{u}} + \mathbf{E}x + \mathbf{g})_j\}.$$

Based on the convergence properties of algorithm **(IDFG)** presented in Section 2.3 and relations (4.8) and (4.9) we can choose:

$$k_{\rm out} = \left\lfloor 8\sqrt{\frac{\left(2\sqrt{p}+1\right) L_{\rm d}\mathcal{R}_{\rm d}^2}{\epsilon_{\rm out}}} \right\rfloor \tag{4.11}$$

$$\epsilon_{\rm in} = \frac{\epsilon_{\rm out}\sqrt{\epsilon_{\rm out}}}{8\sqrt{2}\sqrt{L_{\rm d}}\mathcal{R}_{\rm d}\left(2\sqrt{p}+1\right)^{\frac{3}{2}}}, \quad \epsilon_{\rm c} = \frac{\epsilon_{\rm out}}{\left(2\sqrt{p}+1\right)\mathcal{R}_{\rm d}}.$$

The $\epsilon_{\rm out}$-suboptimality and feasibility of the MPC scheme based on algorithm **(IDFG)** can be proved now in a similar way as the one for algorithm **(IDG)** using Theorems 2.9 and 2.10, i.e.:

$$0 \leq F(x, \hat{\mathbf{u}}^{k_{\rm out}}) - F^*(x) \leq \epsilon_{\rm out} \quad \text{and}$$
$$\hat{\mathbf{u}}^{k_{\rm out}} \in \mathbf{U}, \quad \mathbf{G}\hat{\mathbf{u}}^{k_{\rm out}} + \mathbf{E}x + \mathbf{g} < 0.$$

In conclusion, in our MPC scheme from our suboptimal and feasible control sequence $\hat{\mathbf{u}}^{k_{\rm out}}$ only the first input $\hat{\mathbf{u}}^{k_{\rm out}}(0)$ is applied to the system according to the receding horizon strategy.

4.4. Stability of the MPC scheme. For stability analysis, we express for the entire network system the dynamics, the matrices corresponding to the total stage and final costs, and the total terminal set as: $x(t+1) = Ax(t) + Bu(t), Q, R, P$ and $X^{\rm f}$, respectively. Further, the next state in our MPC scheme is denoted $x^+ = Ax + B\hat{\mathbf{u}}^{k_{\rm out}}(0)$ and a new sequence of feasible inputs for the MPC problem at the next state x^+ is denoted with $\tilde{\mathbf{u}}^+ = \left[\left(\hat{\mathbf{u}}^{k_{\rm out}}(1)\right)^T \dots \left(\hat{\mathbf{u}}^{k_{\rm out}}(N-1)\right)^T (Kx(N))^T\right]^T$, where $u = Kx$ is a linear feedback controller. In this section we will make use of the following assumptions:

Assumption 4.1. *(i) The terminal constraint set X^{f} is positively invariant for the closed-loop system $x(t+1) = (A+BK)x(t)$, i.e. for all $x \in \mathrm{int}(X^{\mathrm{f}})$ we have that $(A+BK)x \in \mathrm{int}(X^{\mathrm{f}})$.*

(ii) The following relation holds:

$$F(x^+, \tilde{\mathbf{u}}^+) \leq F(x, \hat{\mathbf{u}}^{k_{\mathrm{out}}}) - \|x\|_Q^2 \quad \forall x \in X_N. \tag{4.12}$$

Assumption 4.1 is standard in the the MPC framework (see also [8, 26]). Moreover, distributed synthesis procedures for finding the matrices K an P for the terminal controller and terminal cost such that Assumption 4.1 holds can be found e.g. in [16].

Based on Assumption 4.1 (i) and the fact that $\mathbf{G}\hat{\mathbf{u}}^{k_{\mathrm{out}}} + \mathbf{E}x + \mathbf{g} < 0$ we can immediately see that $\tilde{\mathbf{u}}^+$ is a strict Slater vector of the MPC problem (4.4) with initial state x^+. Therefore, in the MPC problem for the next state x^+ we update the strict Slater vector as explained above, i.e.:

$$\tilde{\mathbf{u}}^+ = \left[\left(\hat{\mathbf{u}}^{k_{\mathrm{out}}}(1)\right)^T \dots \left(\hat{\mathbf{u}}^{k_{\mathrm{out}}}(N-1)\right)^T (Kx(N))^T\right]^T$$

and thus $\tilde{\mathbf{u}}^+$ is also feasible for tightened problem (4.5).

In order to prove asymptotic stability of the MPC scheme for all $x \in X_N$ we use similar arguments as in [8, 26] by showing that $F(x, \hat{\mathbf{u}}^{k_{\mathrm{out}}})$ is a Lyapunov function:

$$\begin{aligned} F(x^+, (\hat{\mathbf{u}}^{k_{\mathrm{out}}})^+) &\leq F^*(x^+) + \epsilon_{\mathrm{out}}^+ \leq F^*_{\epsilon_{\mathrm{c}}}(x^+) + \epsilon_{\mathrm{out}}^+ \leq F(x^+, \tilde{\mathbf{u}}^+) + \epsilon_{\mathrm{out}}^+ \\ &\overset{(4.12)}{\leq} F(x, \hat{\mathbf{u}}^{k_{\mathrm{out}}}) - \|x\|_Q^2 + \epsilon_{\mathrm{out}}^+, \end{aligned}$$

where $\epsilon_{\mathrm{out}}^+$ denotes the outer accuracy for solving MPC problem (4.5) at initial state x^+. From the previous discussion we have that choosing e.g.

$$\epsilon_{\mathrm{out}}^+ \leq \min\left\{\frac{1}{2}\|x\|_Q^2,\ c(p) \min_{j=1,\dots,p}\{-\left(\mathbf{G}\tilde{\mathbf{u}}^+ + \mathbf{E}x^+ + \mathbf{g}\right)_j\}\right\}, \tag{4.13}$$

we get asymptotic stability of the closed-loop system. Here, $c(p) = \sqrt{p} + 0.05$ for algorithm **(IDG)** and $c(p) = \sqrt{p} + 0.5$ for **(IDFG)**.

4.5. Distributed implementation. In this section we discuss some technical aspects for the distributed implementation of our inexact dual decomposition methods in the case of MPC problem (4.3) and its equivalent form (4.4).

Usually, for the dynamics (4.1) the corresponding matrices $\mathbf{H}$ and $\mathbf{G}$ obtained after eliminating the states are dense and despite the fact that algorithms **(IDG)**, **(IDFG)** and **(PCD)** can perform parallel computations (i.e. each subsystem needs to solve small local problems) we need communication between N steps neighborhood subsystems [4, 8]. However, for the dynamics (4.2) the corresponding matrices $\mathbf{H}$ and $\mathbf{G}$ are sparse and in this case in our algorithms **(IDG)**, **(IDFG)** and **(PCD)** we can perform distributed computations (i.e. the subsystems solve small local problems in parallel and they need to communicate only with one neighborhood subsystems as detailed below). Indeed, if the dynamics of the subsystems are given by (4.2), then $x_i(t) = A_{ii}^t x_i(0) + \sum_{l=1}^t \sum_{j\in\mathcal{N}^i} A_{ii}^{l-1} B_{ij} u_j(t-l)$ and thus the matrices $\mathbf{H}$ and $\mathbf{G}$ have a sparse structure (see e.g. [4, 28]). In particular, the complicating constraints have the following structure: for matrix $\mathbf{G}$ the (i,j) block matrices of $\mathbf{G}$, denoted $\mathbf{G}_{ij}$, are zero for all $j \notin \mathcal{N}^i$ for a given subsystem i, while the matrix $\mathbf{E}$ is block diagonal. Further, if we define the neighborhood subsystems of a certain subsystem

i as $\hat{\mathcal{N}}^i = \mathcal{N}^i \cup \{l : \; l \in \mathcal{N}^j, j \in \bar{\mathcal{N}}^i\}$, where $\bar{\mathcal{N}}^i = \{j : \; i \in \mathcal{N}^j\}$, then the matrix $\mathbf{H}$ has all the block matrices $\mathbf{H}_{ij} = 0$ for all $j \notin \hat{\mathcal{N}}^i$ and the matrix $\mathbf{W}$ has all the block matrices $\mathbf{W}_{ij} = 0$ for all $j \notin \bar{\mathcal{N}}^i$, for any given subsystem i. Thus, the ith block components of both $\bar{\nabla} d_{\epsilon_c}$ and $\nabla L_{\epsilon_c}(\mathbf{u}, \lambda)$ can be computed using only local information, i.e. each subsystem $i = 1, \dots, M$ does the following synchronous computations:

$$\bar{\nabla}_i d_{\epsilon_c}(\lambda) = \sum_{j \in \mathcal{N}^i} \mathbf{G}_{ij}\mathbf{u}_j + \mathbf{E}_{ii} x_i + \mathbf{g}_i + \epsilon_c \mathbf{e} \tag{4.14}$$

$$\nabla_i L_{\epsilon_c}(\mathbf{u}, \lambda) = \sum_{j \in \hat{\mathcal{N}}^i} \mathbf{H}_{ij}\mathbf{u}_j + \sum_{j \in \bar{\mathcal{N}}^i} \left(\mathbf{W}_{ij} x_j + \mathbf{G}_{ji}^T \lambda_j\right) + \mathbf{w}_i. \tag{4.15}$$

Note that in the algorithm **(PCD)** the only parameters that we need to compute are the Lipschitz constants L_i. However, in the MPC problems, L_i does not depend on the initial state x and can be computed once, offline, locally by each subsystem i as: $L_i = \lambda_{\max}(\mathbf{H}_{ii})$. From the previous discussion it follows immediately that each subsystem i performs the inner iterations of algorithm **(PCD)** in parallel using distributed computations (see (4.15)) for all $x \in X_N$.

Since the algorithms **(IDG)** and **(IDFG)** use only first order information, we can observe that once $\bar{\nabla}_i d_{\epsilon_c}(\lambda)$ has been computed distributively, as proved in (4.14), all the computations for updating the block component corresponding to subsystem i in λ^k or $\hat{\lambda}^k$ can be done in parallel due to the fact that we have to do only vector operations. However, in these schemes all subsystems need to know the global Lipschitz constant $L_{\mathrm{d}} = \frac{\|\mathbf{G}\|^2}{\lambda_{\min}(\mathbf{H})}$ that usually is difficult to be computed distributively. In practice, a good upper bound on L_{d} is sufficient, e.g. $L_{\mathrm{d}} \leq \frac{\|\mathbf{G}\|_F^2}{\min_i \lambda_{\min}(R_i)}$, where recall that $\| \cdot \|_F$ denotes the Frobenius norm. Note that L_{d} does not depend on x and can be computed offline, before starting the MPC scheme.

In both algorithms **(IDG)** and **(IDFG)**, another global constant that has to be updated is the upper bound on the norm of the optimal multiplier, $\mathcal{R}_{\mathrm{d}}$. Based on the theory developed in the previous sections, after some long but straightforward computations an easily computed upper bound for the next $\mathcal{R}_{\mathrm{d}}^+$ corresponding to the MPC problem with initial state x^+ is given by:

$$\mathcal{R}_{\mathrm{d}}^+ \leq \frac{\epsilon_c \langle \hat{\lambda}^{k_{\mathrm{out}}}, e\rangle + 4\epsilon_{\mathrm{out}} - \|x\|_Q^2}{\min\limits_{j=1,\dots,M} \{-(\mathbf{G}\tilde{\mathbf{u}}^+ + \mathbf{E}x^+ + g)_j\}}. \tag{4.16}$$

Note that these upper bounds on L_{d} and $\mathcal{R}_{\mathrm{d}}^+$ can be computed distributively in an efficient way.

From the previous discussion we can conclude that the sequences λ_k, $\hat{\lambda}_k$ and $\bar{\mathbf{u}}_k$, generated by the algorithms **(IDG)**/**(IDFG)** and **(PCD)** can be computed in parallel and distributively provided that good estimates for L_{d} and $\mathcal{R}_{\mathrm{d}}$ are known by each subsystem. The effects of the upper bound for $\mathcal{R}_{\mathrm{d}}$ on the overall performance of the MPC scheme are discussed in Sections 5.2.

5. Numerical tests. In order to certify the efficiency of the proposed algorithms, we consider different numerical scenarios. We first analyze the behavior of algorithms **(IDG)**, **(IDFG)** and **(PCD)** on randomly generated QP problems and then we compare our algorithms with other QP solvers used in the context of distributed MPC. The algorithms were implemented on a PC, with 2 Intel Xeon E5310 CPUs at 1.60 GHz and 4Gb of RAM.

5.1. Practical behavior of newly developed algorithms (IDG), (IDFG) and (PCD) . We consider random QP problems of the form:

$$F^* = \min_{\mathrm{lb} \le \mathbf{u} \le \mathrm{ub}, G\mathbf{u}+g \le 0} F(\mathbf{u}) \quad (= 0.5\mathbf{u}^T H \mathbf{u} + w^T \mathbf{u}), \tag{5.1}$$

where matrices $H \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{2n\times n}$ are taken from a normal distribution with zero mean and unit variance. Matrix H is then made positive definite by the transformation $H \leftarrow H^T H + I_n$. Further, $\mathrm{ub} = -\mathrm{lb} = 1$ and w, g are taken from a uniform distribution. For different QP dimensions ranging from $n = 100$ to $n = 1000$, we first analyze the behavior of algorithms **(IDG)** and **(IDFG)** in terms of the parameters choice.

For each n, we consider two different estimates for the number of outer iterations depending on the way we compute $R_\mathrm{d} = \|\lambda^*\|$, where λ^* is an optimal Lagrange multiplier. For algorithm **(IDG)**, k^G_{out} is the average number of iterations obtained using the bound $\mathcal{R}_\mathrm{d}$ given in (2.22) - Section 2.2, while $k^G_{\mathrm{out,samp}}$ is the average number of iterations obtained with $R_\mathrm{d} = \|\lambda^*\|$, where λ^* is computed exactly using Matlab's `Quadprog`, iterations which correspond to 10 random QP problems. We also compute the average number of outer iterations $k^G_{\mathrm{out,real}}$ observed in practice, obtained by imposing the stopping criteria $|F(\hat{\mathbf{u}}^{k^G_{\mathrm{out,real}}}) - F^*|$ and $\| \left[G\hat{\mathbf{u}}^{k^G_{\mathrm{out,real}}} + g\right]_+ \|$ to be less than the estimates established in Section 2.2 for an outer accuracy $\epsilon_{\mathrm{out}} = 10^{-3}$. Using the results from Section 2.3 we compute in a similar way k^{FG}_{out}, $k^{FG}_{\mathrm{out,samp}}$ and $k^{FG}_{\mathrm{out,real}}$ for algorithm **(IDFG)**. The results for both algorithms are presented in Figure 5.1. We

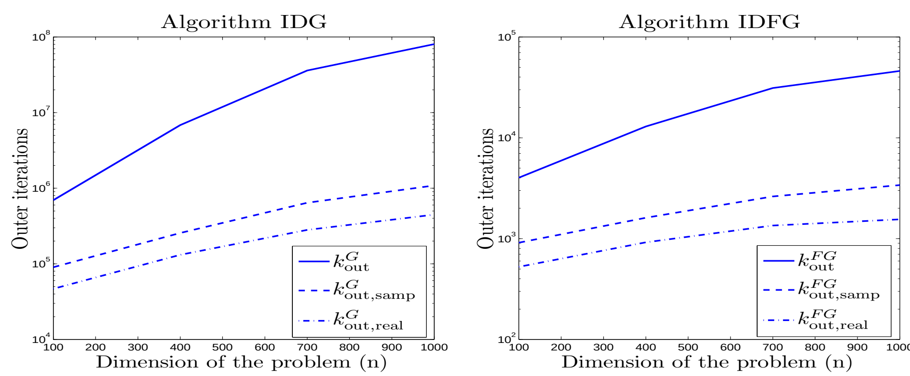

FIG. 5.1. *Values of* k^s_{out}, $k^s_{\mathrm{out,samp}}$ *and* $k^s_{\mathrm{out,real}}$ $(s = \{G; FG\})$ *for algorithms **(IDG)** (left) and **(IDFG)** (right),* $\epsilon_{out} = 10^{-3}$.

can observe that in practice algorithm **(IDFG)** performs much better than algorithm **(IDG)**. Note that the expected number of outer iterations $k^G_{\mathrm{out,samp}}$ and $k^{FG}_{\mathrm{out,samp}}$ obtained from our derived bounds in Sections 2.2 and 2.3 offer a good approximation for the real number of iterations of the two algorithms. Thus, these simulations show that our derived bounds are tight. But, when in our derived estimates we use $\mathcal{R}_\mathrm{d}$, then k^{FG}_{out} is about one order of magnitude, while k^G_{out} is about two orders of magnitude greater than the real number of iterations.

Since the estimates for suboptimality and feasibility violation are also dependent on the way the inner accuracy ϵ_{in} is chosen, we are also interested in the behavior of the two algorithms w.r.t. ϵ_{in}. For this purpose, we apply algorithms **(IDG)** and **(IDFG)**

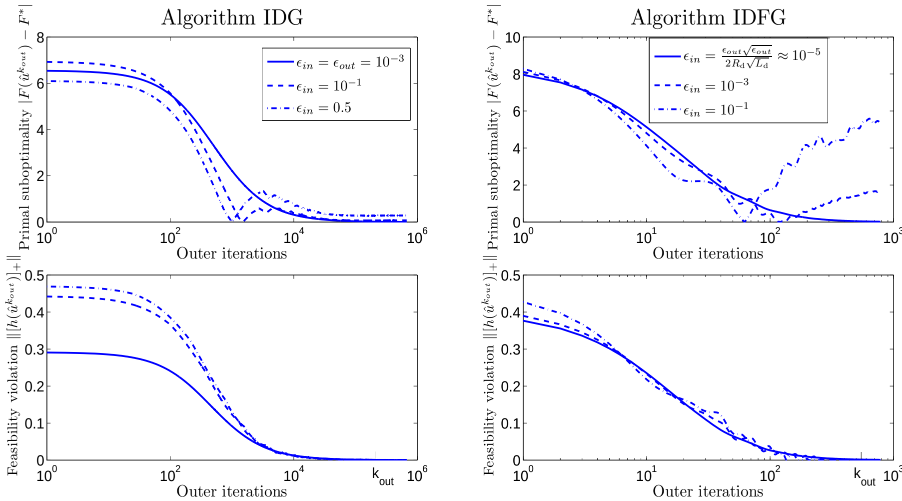

FIG. 5.2. *Suboptimality and feasibility violation for algorithms **(IDG)** (left) and **(IDFG)** (right) for $\epsilon_{out} = 10^{-3}$ and different ϵ_{in}.*

for solving a random QP problem of dimension $n = 300$, with a fixed outer accuracy $\epsilon_{\rm out} = 10^{-3}$ and different values of $\epsilon_{\rm in}$. In Figure 5.2 we plot the primal suboptimality and the feasibility violation by letting the two algorithms perform the number of outer iterations computed in Sections 2.2 and 2.3. We can observe from Figure 5.2 that if the inner accuracy $\epsilon_{\rm in}$ is chosen too large, the desired level of suboptimality cannot be attained. We can also see that algorithm **(IDG)** is less sensitive to the choice of the inner accuracy $\epsilon_{\rm in}$ than algorithm **(IDFG)** due to the fact that algorithm **(IDFG)** accumulates errors (see Theorems 2.6 and 2.10).

In conclusion, we notice from the results of Sections 2.2 and 2.3 and simulations that there is a tradeoff between the speed of convergence and robustness: e.g. algorithm **(IDFG)** is faster than algorithm **(IDG)**, but the second one is more robust since it does not accumulate the errors. Thus, depending on the application, one can choose between the two algorithms.

		(PCD)		Jacobi [28]		**(PCD)** centralized	Quadprog
M	n	CPU (sec)	Iter	CPU (sec)	Iter	CPU (sec)	CPU (sec)
10	100	0.09	262	0.38	82	0.16	0.08
	500	0.61	1244	2.12	715	0.75	1.27
	800	2.11	2702	19.3	1274	9.3	3.8
	1000	2.69	2851	23.05	1375	10.1	4.1

TABLE 5.1
*CPU time (seconds) and number of inner iterations for algorithms **(PCD)** and Jacobi [28].*

We also implemented for comparison, algorithms **(PCD)** and Jacobi from [28]. Both algorithms were implemented in C code, with parallelization ensured via MPI. Table 5.1 presents the average CPU time in seconds and number of iterations for each algorithm for 10 random QP problems (5.1) with only box constraints. Since the convergence rate for algorithm in [28] is not known, the stopping criterion for each algorithm is $F(\mathbf{u}^k) - F^* \leq 10^{-3}$, with F^* being precomputed using `Quadprog`. As we can see algorithm **(PCD)** is about 10 times faster than the algorithm in [28].

5.2. MPC for traffic networks. In this section we analyze the behavior of algorithms **(IDG)** and **(IDFG)** on MPC problems for traffic network systems. In [5] the authors show that traffic network systems can be modeled in the form (4.2). We generated ring traffic networks with M even junctions (subsystems) and having $M/2$ input links and $M/2$ output links distributed randomly. In order to work with small costs, we normalized the state of the system as: $x \leftarrow x/10^3$. For the parameters of the system and of the MPC problem see [5] and the references therein. Note that the number of states or inputs in this traffic network is $3M/2$.

Avg. no. of iter.	$M = 6$	$M = 12$	$M = 18$
k_{out}^{FG}	194	327	443
k_{out}^{G}	9452	26734	49113
k_{out}^{SG}	$4.9 \cdot 10^5$	$9.9 \cdot 10^5$	$1.5 \cdot 10^6$
$k_{\text{out,real}}^{FG}/F(x, \hat{\mathbf{u}}^{k_{\text{out,real}}^{FG}}) - F^*(x)$	$57/1.3 \cdot 10^{-4}$	$71/2.7 \cdot 10^{-4}$	$89/3.8 \cdot 10^{-4}$
$k_{\text{out,real}}^{G}/F(x, \hat{\mathbf{u}}^{k_{\text{out,real}}^{G}}) - F^*(x)$	$726/1.8 \cdot 10^{-4}$	$1289/2.5 \cdot 10^{-4}$	$1836/3.4 \cdot 10^{-4}$
$k_{\text{out,real}}^{SG}$	$2 \cdot 10^4$	$2 \cdot 10^4$	$2 \cdot 10^4$
$\lvert F(x, \hat{\mathbf{u}}^{k_{\text{out,real}}^{SG}}) - F^*(x)\rvert$	$9.8 \cdot 10^{-3}$	$5.8 \cdot 10^{-2}$	$8.2 \cdot 10^{-2}$
$\max_j \{(\mathbf{G}\hat{\mathbf{u}}^{k_{\text{out,real}}^{SG}} + \mathbf{E}x + \mathbf{g})_j\}$	$1.2 \cdot 10^{-4}$	$5.6 \cdot 10^{-4}$	$2.4 \cdot 10^{-3}$

Table 5.2
Averaged number of iterations and cost decrease for different number of junctions (subsystems).

The distributed MPC approach with a prediction horizon of $N = 10$ steps was applied for solving a single time step of the traffic network with $M \in \{6, 12, 18\}$ number of junctions using the newly developed algorithms **(IDG)** and **(IDFG)** and the dual subgradient algorithm in [8]. For each M the results are shown for a set of 10 initial states obtained at random. Additionally to the input constraints considered in [5] we also assume box constraints on the states. We solve the tightened problem (4.5), obtained from the MPC problem of form (4.3) or equivalently (4.4), with an outer accuracy $\epsilon_{\text{out}} = 10^{-2}$. In Table 5.2 we report the average number of outer iterations k_{out}^{FG}, k_{out}^{G} and k_{out}^{SG} performed by the algorithms **(IDFG)**, **(IDG)** and the algorithm in [8], respectively. We also count the average real number of iterations performed by algorithms **(IDFG)** and **(IDG)** by imposing the stopping criterion $F(x, \hat{\mathbf{u}}^{k_{\text{out}}}) - F^* \leq \epsilon_{\text{out}}$ and $\mathbf{G}\hat{\mathbf{u}}^{k_{\text{out}}} + \mathbf{E}x + \mathbf{g} \leq 0$. For the dual subgradient algorithm in [8] the stopping criterion was chosen as follows: $F(x, \hat{\mathbf{u}}^{k_{\text{out}}}) - F^* \leq \epsilon_{\text{out}}^{SG}$ and $\mathbf{G}\hat{\mathbf{u}}^{k_{\text{out}}} + \mathbf{E}x + \mathbf{g} \leq 0$, where $\epsilon_{\text{out}}^{SG}$ and the rest of the parameters for this algorithm are computed as in [8, Section III.C]. In all three algorithms the inner problems were solved with algorithm **(PCD)**. From Table 5.2 we observe that algorithm **(IDFG)** has the best behavior compared to **(IDG)** and the dual subgradient algorithm in [8]. Thus, algorithm **(IDFG)** is superior in terms of both, predicted (theoretical) and real number of iterations (e.g. from 10 to 100 times faster than **(IDG)**). Further, algorithm **(IDG)** is able to produce a feasible and suboptimal solution in a reasonable number of outer iterations, while the dual subgradient algorithm in [8] failed to generate a feasible solution within $2 \cdot 10^4$ outer iterations. We observed that this behavior is due mainly to the fact that the step size in **(IDG)** is larger than that in [8].

6. Conclusions. Motivated by MPC problems for complex interconnected systems, we have proposed two dual based methods for solving large-scale smooth convex optimization problems with coupling constraints. We moved the coupling constraints into the cost using duality theory. We solved the inner subproblems only up to a certain accuracy by means of a parallel coordinate descent method for which we have proved linear convergence. For solving the outer problems, we developed inexact dual gradient and fast gradient schemes for which we provide a full convergence analysis, deriving upper bounds on dual and primal suboptimality and primal feasibility violation. We also discussed some implementation issues of the new algorithms for distributed MPC problems and tested them on several practical applications.

Acknowledgment. The research leading to these results has received funding from: the European Union (FP7/2007–2013) under grant agreement no 248940; CNCS (project TE-231, 19/11.08.2010); ANCS (project PN II, 80EU/2010); POSDRU/89/1.5/S/62557 and POSDRU/107/1.5/S/76909.
The authors thank Y. Nesterov, D. Doan and T. Keviczky for interesting discussions.

Appendix. *Proof of Lemma 2.1.*
Case 1 - We first consider the unconstrained case, i.e. $\mathbf{U} = \mathbb{R}^n$. Since F is strongly convex, it follows that $\mathbf{u}(\lambda)$ is unique and thus d is a differentiable function having the gradient:

$$\begin{aligned}\nabla d(\lambda) &= \nabla \mathbf{u}(\lambda)^T \nabla F(\mathbf{u}(\lambda)) + h(\mathbf{u}(\lambda)) + \nabla \mathbf{u}(\lambda)^T \nabla h(\mathbf{u}(\lambda))^T \lambda \\ &= \nabla \mathbf{u}(\lambda)^T \left[\nabla F(\mathbf{u}(\lambda)) + \nabla h(\mathbf{u}(\lambda))^T \lambda\right] + h(\mathbf{u}(\lambda)) = h(\mathbf{u}(\lambda)),\end{aligned}$$

where the last equality is obtained using the optimality conditions for $\mathbf{u}(\lambda)$, i.e.:

$$\nabla F(\mathbf{u}(\lambda)) + \nabla h(\mathbf{u}(\lambda))^T \lambda = 0 \quad \forall \lambda \geq 0. \tag{6.1}$$

Taking now into account that the components of h are twice differentiable we have:

$$\nabla^2 d(\lambda) = \nabla h(\mathbf{u}(\lambda)) \nabla \mathbf{u}(\lambda). \tag{6.2}$$

Differentiating now the optimality conditions (6.1) w.r.t. to λ we can write:

$$\nabla \mathbf{u}(\lambda)^T \nabla^2 F(\mathbf{u}(\lambda)) + \nabla h(\mathbf{u}(\lambda)) + \nabla \mathbf{u}(\lambda)^T \sum_{i=1}^{p} \lambda_i \nabla^2 h_i(\mathbf{u}(\lambda)) = 0,$$

from which we obtain:

$$\nabla \mathbf{u}(\lambda)^T = -\nabla h(\mathbf{u}(\lambda)) \left[\nabla^2 F(\mathbf{u}(\lambda)) + \sum_{i=1}^{p} \lambda_i \nabla^2 h_i(\mathbf{u}(\lambda))\right]^{-1}.$$

Introducing this relation into (6.2) and taking into account that $\sum_{i=1}^{p} \lambda_i \nabla^2 h_i(\mathbf{u}(\lambda)) \succeq 0$ we have:

$$\begin{aligned}-\nabla^2 d(\lambda) &= \nabla h(\mathbf{u}(\lambda)) \left[\nabla^2 F(\mathbf{u}(\lambda)) + \sum_{i=1}^{p} \lambda_i \nabla^2 h_i(\mathbf{u}(\lambda))\right]^{-1} \nabla h(\mathbf{u}(\lambda))^T \\ &\preceq \nabla h(\mathbf{u}(\lambda)) \left[\nabla^2 F(\mathbf{u}(\lambda))\right]^{-1} \nabla h(\mathbf{u}(\lambda))^T.\end{aligned}$$

Since F is σ_{F}-strongly convex and thus $\nabla^2 F(\mathbf{u}(\lambda)) \succeq \sigma_{\mathrm{F}} I_n$ and the Jacobian of h is bounded (see Assumption (1.1)), we can write further:

$$\|\nabla^2 d(\lambda)\| \leq \| \left[\nabla^2 F(\mathbf{u}(\lambda))\right]^{-1} \| \cdot \|\nabla h(\mathbf{u}(\lambda))\|^2 \leq \| \left[\nabla^2 F(\mathbf{u}(\lambda))\right]^{-1} \| \cdot \|\nabla h(\mathbf{u}(\lambda))\|_F^2 \leq \frac{c_{\mathrm{h}}^2}{\sigma_{\mathrm{F}}}.$$

Thus, we can conclude using Lemma 1.2.2 from [21] that $L_{\mathrm{d}} = \frac{c_{\mathrm{h}}^2}{\sigma_{\mathrm{F}}}$.

Case 2 - We assume now that $\mathbf{U}$ is a compact convex set. Since F is strongly convex, the dual function d is still differentiable and given by $\nabla d(\lambda) = h(\mathbf{u}(\lambda))$. In order to show Lipschitz continuity of the gradient, we consider the following family of dual functions $(d_\tau)_{\tau>0}$:

$$d_\tau(\lambda) = \min_{\mathbf{u}\in\mathbb{R}^n} F(\mathbf{u}) + \langle \lambda, h(\mathbf{u})\rangle + \tau b_{\mathbf{U}}(\mathbf{u}), \tag{6.3}$$

where $b_{\mathbf{U}}$ is a self-concordant barrier function for the set $\mathbf{U}$. Let $u(\lambda,\tau)$ be the optimal solution of (6.3). Using the same reasoning as in the unconstrained case and taking into account that $\nabla^2 b_{\mathbf{U}}(\mathbf{u}) \succeq 0$ (see [21, Section 4.2.2]), we have that for any given $\tau > 0$ the gradient $\nabla d_\tau(\lambda) = h(\mathbf{u}(\lambda,\tau))$ is Lipschitz continuous with constant $L_{\mathrm{d}_\tau} = \frac{c_{\mathrm{h}}^2}{\sigma_{\mathrm{F}}}$, i.e. $\|h(u(\lambda,\tau)) - h(u(\nu,\tau))\| \leq \frac{c_{\mathrm{h}}^2}{\sigma_{\mathrm{F}}}\|\lambda-\nu\|$ for all $\lambda,\nu \geq 0$. Since for all $\lambda \geq 0$ we have $d_\tau(\lambda) \to d(\lambda)$, $u(\lambda,\tau) \to u(\lambda)$ as $\tau \to +0$ and h is a continuous function we can conclude that the gradient of the dual function d is also Lipschitz continuous with constant $L_{\mathrm{d}} = \frac{c_{\mathrm{h}}^2}{\sigma_{\mathrm{F}}}$. ◻

REFERENCES

[1] I. Alvarado, D. Limon, D. Munoz de la Pena, J.M. Maestre, M.A. Ridao, H. Scheu, W. Marquardt, R.R. Negenborn, B. De Schutter, F. Valencia, and J. Espinosa, "A comparative analysis of distributed MPC techniques applied to the HD-MPC four-tank benchmark", *Journal of Process Control*, vol. 21, no. 5, pp. 800 - 815, 2011.

[2] D.P. Bertsekas and J.N. Tsitsiklis, *Paralel and distributed computation: Numerical Methods*, Prentice Hall, 1989.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers", *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1 - 124, 2011.

[4] E. Camponogara and H.F. Scherer, "Distributed Optimization for Model Predictive Control of Linear Dynamic Networks With Control-Input and Output Constraints", *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 233 - 242, 2011.

[5] E. Camponogara and L.B. de Oliveira, "Distributed Optimization for Model Predictive Control of Linear-Dynamic Networks", *IEEE Transactions on Systems, Man and Cybernetics: Part A*, vol. 39, no. 6, pp. 1331 - 1338, 2009.

[6] P.D. Christofides, R. Scattolini, D. Munoz de la Pena and J. Liu, "Distributed model predictive control: A tutorial review and future research directions", *Computers and Chemical Engineering*, pp. 1 - 21, 2012.

[7] W.B. Dunbar and R. Murray, "Distributed Receding Horizon Control with Application to Multi-Vehicle Formation Stabilization", *Automatica*, vol. 42, pp. 549 - 558, 2006.

[8] M.D. Doan, T. Keviczky and B. De Schutter, "A distributed optimization-based approach for hierarchical model predictive control of large-scale systems with coupled dynamics and constraints", in *Proc. 50th IEEE CDC*, pp. 5236 - 5241, 2011.

[9] O. Devolder, F. Glineur and Y. Nesterov, "First order methods of smooth convex optimization with inexact oracle", *CORE Discussion Paper 2011/02*, 2011, `http://www.optimization-online.org/DB_FILE/2010/12/2865.pdf`.

[10] M. Farina and R. Scattolini, "Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems", *Automatica*, vol. 48, no. 6, pp. 1088 - 1096, 2012.

[11] K.C. Kiwiel, T. Larsson and P.O. Lindberg, "Lagrangian relaxation via ballstep subgradient methods", *Mathematics of Operations Research*, vol. 32, no. 3, pp. 669 - 686, 2007.

[12] Y. Kuwata, A. Richards, T. Schouwenaars and J.P. How, "Distribted robust receding horizon control for multivariable guidance", *IEEE Transactions on Control Systems Technology*, vol. 15, no. 4, pp. 627 - 641, 2007.

[13] T. Larsson, M. Patriksson and A. Stromberg, "Ergodic convergence in subgradient optimization", *Optimization Methods and Software*, vol. 9, no. 1-3, pp. 93 - 120, 1998.

[14] J. Liu, X. Chen, D. Munoz de la Pena and P.D. Christofides, "Iterative distributed model

predictive control of nonlinear systems: Handling asynchronous, delayed measurements", *IEEE Transactions on Automatic Control*, vol. 57, pp. 528 534, 2012.

[15] J.M. Maestre, D. Munoz de la Pena, E.F. Camacho, T. Alamo, "Distributed model predictive control based on agent negotiation", *Journal of Process Control*, vol. 21, no. 12, pp. 685 - 697, 2011.

[16] I. Necoara and D. Clipici, " Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC", *Journal of Process Control*, vol. 23, no. 3, pp. 243 - 253, 2013.

[17] I. Necoara, V. Nedelcu and I. Dumitrache, "Parallel and distributed optimization methods for estimation and control in networks", *Journal of Process Control*, vol. 21, no. 5, pp. 756 - 766, 2011.

[18] I. Necoara and J. Suykens, "Application of a Smoothing Technique to Decomposition in Convex Optimization", *IEEE Transactions on Automatic Control*, vol. 53, no. 11, pp. 2674 - 2679, 2008.

[19] A. Nedic and A. Ozdaglar, "Approximate primal solutions and rate analysis for dual subgradient methods", *SIAM Journal on Optimization*, vol. 19, no. 4, pp. 1757 - 1780, 2009.

[20] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems", *SIAM Journal on Optimization*, vol. 22, no. 2, pp. 341 - 362, 2012.

[21] Y. Nesterov, *Introductory lectures on convex optimization*, Springer, 2004.

[22] Y. Nesterov, "Smooth minimization of non-smooth functions", *Mathematical Programming*, vol. 103, pp. 127 - 152, 2004.

[23] P. Patrinos and A. Bemporad, "An accelerated dual gradient-projection algorithm for embedded linear model predictive control", submitted, 2012.

[24] A. Richards and J.P. How, "Robust distributed model predictive control", *International Journal of Control*, vol. 80, no. 9, pp. 1517 - 1531, 2007.

[25] R. Scattolini, "Architectures for distributed and hierarchical model predictive control - A review". Journal of Process Control, vol. 19, no. 5, pp. 723 731, 2009.

[26] P.O.M. Scokaert, D.Q. Mayne and J.B. Rawlings, "Suboptimal model predictive control (feasibility implies stability)", *IEEE Transactions on Automatic Control*, vol. 44, no. 3, pp. 648 - 654, 1999.

[27] S. Sen and H.D. Sherali, "A class of convergent primal-dual subgradient algorithms for decomposable convex programs", *Mathematical Programming*, vol. 35, no. 3, pp. 279 - 297, 1986.

[28] B.T. Stewart, A.N. Venkat, J.B. Rawlings, S.J. Wright and G. Pannocchia, "Cooperative distributed model predictive control", *Systems & Control Letters*, vol. 59, no. 8, pp. 460 - 469, 2010.