A proximal center-based decomposition method for multi-agent convex optimization

Ion Necoara and Johan A.K. Suykens

***Abstract*— In this paper we develop a new dual decomposition method for optimizing a sum of convex objective functions corresponding to multiple agents but with coupled constraints. In our method we define a smooth Lagrangian, by using a smoothing technique developed by Nesterov , which preserves separability of the problem. With this approach we propose a new decomposition method (the proximal center method) for which efficiency estimates are derived and which improves the bounds on the number of iterations of the classical dual gradient scheme by an order of magnitude. The method involves every agent optimizing an objective function that is the sum of his own objective function and a smoothing term while the coordination between agents is performed via the Lagrange multipliers corresponding to the coupled constraints. Applications of the new method for solving distributed model predictive control or network optimization problems are also illustrated.**

I. INTRODUCTION

Recent advances in computer power have led to the derivation of many parallel and distributed computation methods for solving large-scale optimization problems (e.g. [1]). For separable convex problems, i.e. separable objective function but with coupled constraints (they arise in many fields of engineering: e.g. network optimization [9], [14], distributed model predictive control (MPC) [13]), many researchers have proposed dual decomposition algorithms such as the dual subgradient method [1], the alternating direction method [1], [4], [5], [12], proximal method of multipliers [2], partial inverse method [11], etc. In general, these methods are based on alternating minimization in a Gauss-Seidel fashion of an (Augmented) Lagrangian followed by a steepest ascent update for the multipliers. However, the stepsize parameter which has a very strong influence on the convergence rate of these methods is very difficult to tune and the minimization in a Gauss-Seidel fashion slows down the convergence towards an optimum. Moreover, they do not provide any complexity estimates for the general case (linear convergence is obtained e.g. for strongly convex functions).

The purpose of this paper is to propose a new decomposition method for separable convex optimization problems that overcomes the disadvantages mentioned above. Based on a smoothing technique recently developed by Nesterov in [8], we obtain a smooth Lagrangian that preserves separability of the problem. Using this smooth Lagrangian, we derive a new dual decomposition method in which the corresponding parameters are selected optimally and thus straightforward to tune. In contrast to the dual gradient update for the multipliers used by most of the decomposition methods from the literature, our method uses an optimal first-order oracle scheme (see e.g. [7], [8]) for updating the multipliers. Therefore, we derive for the new method an efficiency estimate for the general case which improves with one order of magnitude the complexity of the classical dual gradient method (i.e. the steepest ascent update). Up to our knowledge these are the first efficiency estimate results of a dual decomposition method for separable non-strongly convex programs. The new algorithm is suitable for solving distributed model predictive control problems or network optimization problems since it is highly parallelizable and thus it can be effectively implemented on parallel processors.

The reminder of this paper is organized as follows. Section II contains the problem formulation, followed by a brief description of some of the dual decomposition methods derived in the literature. The main results of the paper are presented in Section III, where we describe our new decomposition method and its main properties. In Section IV we show that network optimization or distributed model predictive control problems corresponding to linear systems with interacting subsystem dynamics and decoupled costs can be recast in the framework of separable convex problems for which our algorithm can be applied. Finally, numerical simulations on some test problems are included.

II. PRELIMINARIES

A. Splitting methods for separable convex programs

An important application of convex duality theory is in decomposition algorithms for solving large-scale problems but with special structure. This type of problems arises e.g. in the context of large-scale networks which consists of multiple agents with different objectives or in the context of distributive MPC for linear systems with coupled subsystem dynamics. The following separable convex program will be considered in this paper:

$$f^* = \min_{x \in X, z \in Z} \big\{ \phi_1(x) + \phi_2(z) : \; Ax + Bz = b \big\}, \qquad (1)$$

where $\phi_1 : \mathbb{R}^m \to \mathbb{R}$ and $\phi_2 : \mathbb{R}^p \to \mathbb{R}$ are continuous convex functions on X and Z, respectively, A is an $n \times m$ matrix, B an $n \times p$ matrix, and b is in $\mathbb{R}^n$. In this paper we do *not* assume ϕ_1 and/or ϕ_2 to be strongly convex or even smooth. Moreover, we assume that $X \subseteq \mathbb{R}^m$ and $Z \subseteq \mathbb{R}^p$ are compact convex sets. We use different norms on $\mathbb{R}^n, \mathbb{R}^m$ and $\mathbb{R}^p$, not necessarily the Euclidean norms. However, for simplicity in notation we do not use indices to specify the norms on $\mathbb{R}^n, \mathbb{R}^m$ and $\mathbb{R}^p$, since from the context it is clear in which Euclidean space we are and which norm we use.

I. Necoara and J.A.K. Suykens are with the Katholieke Universiteit Leuven, Department of Electrical Engineering, ESAT–SCD, Kasteelpark Arenberg 10, B–3001 Leuven (Heverlee), Belgium. {ion.necoara,johan.suykens}@esat.kuleuven.be

Remark 2.1 Note that the method developed in this paper can treat both coupled inequalities $Ax + Bz \leq b$ and/or equalities $Ax + Bz = b$ (see [6]). Moreover, we can consider any finite sum of ϕ_i's. However, for simplicity of the exposition we restrict ourselves to problems of the form (1).

Let $\langle \cdot, \cdot \rangle$ denote a scalar product on the Euclidean space $\mathbb{R}^n$. By forming the Lagrangian corresponding to the linear constraints (with the Lagrange multipliers $\lambda \in \mathbb{R}^n$), i.e.

$$\mathcal{L}_0(x, z, \lambda) = \phi_1(x) + \phi_2(z) + \langle \lambda, Ax + Bz - b \rangle,$$

and using the dual decomposition method, one arrives at the following decomposition algorithm:

Algorithm 2.2: ([1]) **for** $k \geq 0$ **do**

1. given λ^k, minimize the Lagrangian $(x^{k+1}, z^{k+1}) = \arg\min_{x \in X, z \in Z} \mathcal{L}_0(x, z, \lambda^k)$, or equivalently minimize over x and z independently

$$x^{k+1} = \arg\min_{x \in X} [\phi_1(x) + \langle \lambda^k, Ax \rangle]$$
$$z^{k+1} = \arg\min_{z \in Z} [\phi_2(z) + \langle \lambda^k, Bz \rangle]$$

2. update the multipliers by the iteration

$$\lambda^{k+1} = \lambda^k + c_k(Ax^{k+1} + Bz^{k+1} - b),$$

where c_k is a positive step-size.

The following assumption holds throughout the paper:

Assumption 2.3: Constraint qualification holds for (1).

It can be shown that Algorithm 2.2 is convergent under Assumption 2.3 and the assumption that both ϕ_1 and ϕ_2 are strongly convex functions (the later guarantees that the minimizer (x^{k+1}, z^{k+1}) is unique). In fact, under the assumption of strong convexity it follows that the dual function

$$f_0(\lambda) = \min_{x \in X, z \in Z} \phi_1(x) + \phi_2(z) + \langle \lambda, Ax + Bz - b \rangle$$

is differentiable [10], and thus Algorithm 2.2 is the gradient method with stepsize c_k for maximizing the dual function.

However, for many interesting problems, especially arising from transformations that leads to decomposition, the functions ϕ_1 and ϕ_2 are not strongly convex. There are some methods (e.g. alternating direction method [1], [5], [12], proximal point method [2], partial inverse method [11]) that overcome this difficulty based on alternating minimization in a Gauss-Seidel fashion of the *Augmented Lagrangian*, followed by a steepest ascent update of the multipliers. A computational drawback of these schemes is that the *prox-term* $\frac{c}{2}\|Ax+Bz-b\|^2$, using the Euclidean norm framework, present in the Augmented Lagrangian is not separable in x and z. Another disadvantage is that they cannot deal with coupled *inequalities* in general. Moreover, these schemes were shown to be very sensitive to the value of the parameter c, with difficulties in practice to obtain the best convergence rate. Some heuristics for choosing c can be found in the literature [2], [5], [12]. Note that alternating direction method variants which allow for inexact minimization was proposed in e.g. [2]. A closely related method is the partial inverse of a monotone operator developed in [11].

B. Accelerated scheme for smooth maximization

In this section we briefly describe an accelerated scheme for smooth convex functions developed by Nesterov in [7], [8]. Let f be a given concave and differentiable function on a closed convex set $\hat{Q} \subseteq \mathbb{R}^n$ that has a Lipschitz continuous gradient with Lipschitz constant L.

Definition 2.4: [8] We define a *prox-function* d of the set $\hat{Q}$ as a function with the following properties: d is continuous, strongly convex on $\hat{Q}$ with convexity parameter σ, and u^0 is the center of the set $\hat{Q}$, i.e. $u^0 = \arg\min_{x \in \hat{Q}} d(x)$ such that $d(u^0) = 0$.

The goal is to find an approximate solution of the convex optimization problem $x^* = \arg\max_{x \in \hat{Q}} f(x)$. In Nesterov's scheme three sequences of points from $\hat{Q}$ are updated recursively: $\{u^k\}_{k \geq 0}, \{x^k\}_{k \geq 0}$, and $\{v^k\}_{k \geq 0}$. The algorithm can be described as follows:

Algorithm 2.5: ([8]) **for** $k \geq 0$ **do**

1. compute $\nabla f(u^k)$

2. find $\bar{x}^k = \arg\max_{x \in \hat{Q}} \langle \nabla f(u^k), x - u^k \rangle - \frac{L}{2}\|x - u^k\|^2$ and define $x^k = \arg\max_{x \in \{\bar{x}^k, x^{k-1}, u^k\}} f(x)$

3. find $v^k = \arg\max_{x \in \hat{Q}} -\frac{L}{\sigma} d(x) + \sum_{l=0}^{k} \frac{l+1}{2} \langle \nabla f(u^l), x - u^l \rangle$

4. set $u^{k+1} = \frac{k+1}{k+3} x^k + \frac{2}{k+3} v^k$.

The main property of the Algorithm 2.5 is the following relation [8]:

$$\frac{(k+1)(k+2)}{4} f(x^k) \geq \max_{x \in \hat{Q}} -\frac{L}{\sigma} d(x) + \sum_{l=0}^{k} \frac{l+1}{2} [f(u^l) + \langle \nabla f(u^l), x - u^l \rangle]. \quad (2)$$

Using this property Nesterov proves in [8] that the efficiency estimates of Algorithm 2.5 is of the order $\mathcal{O}(\sqrt{\frac{L}{\epsilon}})$, higher than the corresponding pure gradient method for the same smooth problem by an order of magnitude.

III. A NEW DECOMPOSITION METHOD BASED ON SMOOTHING THE LAGRANGIAN

In this section we propose a new method to smoothen the Lagrangian of (1), inspired from [8]. This smoothing technique preserves the separability of the problem and moreover the corresponding parameters are easy to tune. Since separability is preserved under this smoothing technique, we derive a new dual decomposition method in which the multipliers are updated according to Algorithm 2.5. Moreover, we obtain an efficiency estimate of the new method for the general case. Note that with our method we can treat both coupled inequalities ($Ax + Bz \leq b$) and/or equalities ($Ax + Bz = b$).

A. Smoothing the Lagrangian

Let d_X and d_Z be two prox-functions for the compact convex sets X and Z, with convexity parameter σ_X and σ_Z, respectively. Denote $x^0 = \arg\min_{x \in X} d_X(x)$, $z^0 = \arg\min_{z \in Z} d_Z(z)$. Since X and Z are compact and d_X and d_Z are continuous, we can choose finite and positive constants $D_X \geq \max_{x \in X} d_X(x)$ and $D_Z \geq$

$\max_{z\in Z} d_Z(z)$. We also introduce the following notation $\|A\| = \max_{\|\lambda\|=1,\|x\|=1}\langle\lambda, Ax\rangle$. Note that

$$\|A\| = \max_{\|x\|=1} \|Ax\|_* \text{ and } \|Ax\|_* \le \|A\|\|x\| \ \forall x,$$

where $\|s\|_* = \max_{\|x\|=1}\langle s, x\rangle$ is the corresponding dual norm of the norm used on $\mathbb{R}^n$ [10]. Similarly for B.

Let us introduce the following function:

$$f_c(\lambda) = \min_{x\in X, z\in Z} \phi_1(x) + \phi_2(z) + \langle\lambda, Ax + Bz - b\rangle + c\big(d_X(x) + d_Z(z)\big), \quad (3)$$

where c is a positive *smoothness parameter* that will be defined in the sequel. Note that the objective function in (3) is separable in x and z, i.e.

$$f_c(\lambda) = -\langle\lambda, b\rangle + \min_{x\in X}[\phi_1(x) + \langle\lambda, Ax\rangle + c\ d_X(x)] + \min_{z\in Z}[\phi_2(z) + \langle\lambda, Bz\rangle + c\ d_Z(z)]. \quad (4)$$

Denote by $x(\lambda)$ and $z(\lambda)$ the optimal solution of the minimization problem in x and z, respectively. Function f_c has the following smoothness properties:

Theorem 3.1: [6] The function f_c is concave and continuously differentiable at any $\lambda \in \mathbb{R}^n$. Moreover, its gradient $\nabla f_c(\lambda) = Ax(\lambda) + Bz(\lambda) - b$ is Lipschitz continuous with Lipschitz constant $L_c = \frac{\|A\|^2}{c\sigma_X} + \frac{\|B\|^2}{c\sigma_Z}$. The following inequalities also hold:

$$f_c(\lambda) \ge f_0(\lambda) \ge f_c(\lambda) - c(D_X + D_Z) \ \ \forall\lambda \in \mathbb{R}^n. \quad (5)$$

Proof: The proof follows from standard convex argument and can be found in [6]. ■

B. A proximal center–based decomposition method

In this section we derive a new decomposition method based on the smoothing technique described in Section III-A. The new algorithm, called here the *proximal center algorithm*, has the nice feature that the *coordination* between agents involves the maximization of a smooth convex objective function (i.e. with Lipschitz continuous gradient). Moreover, the *resource allocation* stage consists in solving in *parallel* by the each agent of a minimization problem with strongly convex objective. The new method belongs to the class of two-level algorithms [3] and is particularly suitable for separable convex problems where the minimizations over x and z in (4) are easily carried out.

Let us apply Nesterov's accelerated method described in Algorithm 2.5 to the concave function f_c that has a Lipschitz continuous gradient:

$$\max_{\lambda\in Q} f_c(\lambda), \quad (6)$$

where Q is a given closed convex set in $\mathbb{R}^n$ that contains at least one optimal multiplier $\lambda^* \in \Lambda^*$ (here Λ^* denotes the set of optimal multipliers). Notice that $Q \subseteq \mathbb{R}^n$ in the presence of linear equalities (i.e. $Ax+Bz-b = 0$), $Q \subseteq \mathbb{R}^n_+$, where $\mathbb{R}_+$ denotes the set of nonnegative real numbers, in the presence of linear inequalities (i.e. $Ax + Bz - b \le 0$), or $Q \subseteq (\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}_+)$, where $n_1 + n_2 = n$, when both, equalities and inequalities are present. Note that according to Algorithm 2.5 we also need to choose a prox-function d_Q for the set Q with the convexity parameter σ_Q and center u^0. The proximal center algorithm can be described as follows:

Algorithm 3.2: **for** $k \ge 0$ **do**

1. given u^k compute in *parallel*

$$x^{k+1} = \arg\min_{x\in X}[\phi_1(x) + \langle u^k, Ax\rangle + c\ d_X(x)],$$
$$z^{k+1} = \arg\min_{z\in Z}[\phi_2(z) + \langle u^k, Bz\rangle + c\ d_Z(z)].$$

2. compute $f_c(u^k), \nabla f_c(u^k) = Ax^{k+1} + Bz^{k+1} - b$

3. find $\bar\lambda^k = \arg\max_{\lambda\in Q}\langle\nabla f_c(u^k), \lambda - u^k\rangle - \frac{L_c}{2}\|\lambda - u^k\|^2$ and define $\lambda^k = \arg\max_{\lambda\in\{\bar\lambda^k, \lambda^{k-1}, u^k\}} f_c(\lambda)$

4. find $v^k = \arg\max_{\lambda\in Q} -\frac{L_c}{\sigma_Q} d_Q(\lambda) + \sum_{l=0}^k \frac{l+1}{2}\langle\nabla f_c(u^l), \lambda - u^l\rangle$

5. set $u^{k+1} = \frac{k+1}{k+3}\lambda^k + \frac{2}{k+3}v^k$.

The proximal center algorithm is suitable for decomposition since it is highly parallelizable: the agents can solve their corresponding minimization problems in parallel. This is an advantage of our method compared to most of the alternating direction methods based on Gauss-Seidel iterations that do not have this feature. We now derive a lower bound for the value of the objective function which will be used frequently in the sequel:

Lemma 3.3: [6] For any $\lambda^* \in \Lambda^*$ and $\hat x \in X, \hat z \in Z$, the following lower bound on primal gap holds: $[\phi_1(\hat x)+\phi_2(\hat z)] - f^* \ge \langle A\hat x + B\hat z - b, \lambda^*\rangle \ge -\|\lambda^*\|_*\|A\hat x + B\hat z - b\|_*$.

The previous lemma shows that if $\|A\hat x + B\hat z - b\|_* \le \epsilon_c$, then the primal gap is bounded: for all $\hat\lambda \in Q$

$$-\epsilon_c\|\lambda^*\|_* \le \phi_1(\hat x) + \phi_2(\hat z)] - f^* \le \phi_1(\hat x) + \phi_2(\hat z) - f_0(\hat\lambda). \quad (7)$$

Therefore, if we are able to derive an upper bound ϵ for the duality gap and ϵ_c for the coupling constraints for some given $\hat\lambda$ and $\hat x \in X, \hat z \in Z$, then we conclude that $(\hat x, \hat z)$ is an (ϵ, ϵ_c)-solution for problem (1) (since in this case $-\epsilon_c\|\lambda^*\|_* \le \phi_1(\hat x) + \phi_2(\hat z) - f^* \le \epsilon$ for all $\lambda^* \in \Lambda^*$). The next theorem derives an upper bound on the duality gap for our method.

Theorem 3.4: Assume that there exists a closed convex set Q that contains a $\lambda^* \in \Lambda^*$. Then, after k iterations we obtain an approximate solution to the problem (1) $(\hat x, \hat z) = \sum_{l=0}^k \frac{2(l+1)}{(k+1)(k+2)}(x^{l+1}, z^{l+1})$ and $\hat\lambda = \lambda^k$ which satisfy the following duality gap:

$$0 \le [\phi_1(\hat x) + \phi_2(\hat z)] - f_0(\hat\lambda) \le c(D_X + D_Z) - \max_{\lambda\in Q}\big[-\frac{4L_c}{\sigma_Q(k+1)^2} d_Q(\lambda) + \langle A\hat x + B\hat z - b, \lambda\rangle\big]. \quad (8)$$

Proof: For an arbitrary c, we have from the inequality (2) that after k iterations the Lagrange multiplier $\hat\lambda$ satisfies the following relation:

$$\frac{(k+1)(k+2)}{4} f_c(\hat\lambda) \ge \max_{\lambda\in Q} -\frac{L_c}{\sigma_Q} d_Q(\lambda) + \sum_{l=0}^k \frac{l+1}{2}[f_c(u^l) + \langle\nabla f_c(u^l), \lambda - u^l\rangle].$$

In view of the previous inequality we have:

$$f_c(\hat{\lambda}) \geq \max_{\lambda \in Q} -\frac{4L_c}{\sigma_Q(k+1)^2} d_Q(\lambda) + \sum_{l=0}^{k} \frac{2(l+1)}{(k+1)(k+2)} [f_c(u^l) + \langle \nabla f_c(u^l), \lambda - u^l \rangle].$$

Now, we replace $f_c(u^l)$ and $\nabla f_c(u^l)$ with the expressions given in step 2 of Algorithm 3.2:

$$\sum_{l=0}^{k} \frac{2(l+1)}{(k+1)(k+2)} [f_c(u^l) + \langle \nabla f_c(u^l), \lambda - u^l \rangle] =$$
$$\sum_{l=0}^{k} \frac{2(l+1)}{(k+1)(k+2)} [\phi_1(x^{l+1}) + \phi_2(z^{l+1}) +$$
$$c\big(d_X(x^{l+1}) + d_Z(z^{l+1})\big) + \langle Ax^{l+1} + Bz^{l+1} - b, \lambda \rangle] \geq$$

$$\sum_{l=0}^{k} \frac{2(l+1)}{(k+1)(k+2)} [\phi_1(x^{l+1}) + \phi_2(z^{l+1}) + \langle Ax^{l+1} + Bz^{l+1} - b, \lambda \rangle] \geq$$
$$\phi_1(\hat{x}) + \phi_2(\hat{z}) + \langle A\hat{x} + B\hat{z} - b, \lambda \rangle, \quad \forall \lambda \in Q.$$

The first inequality follows from the definition of a prox-function and the second inequality follows from the fact that ϕ_1 and ϕ_2 are convex functions. Using the last relation and (5) we derive the bound (8) on the duality gap. ■

We now show how to construct the set Q and how to choose optimally the smoothness parameter c. In the next two sections we discuss two cases depending on the choices for Q and d_Q.

C. Efficiency estimates for compact Q

Let D_Q be a positive constant satisfying

$$\max_{\lambda \in Q} d_Q(\lambda) \leq D_Q. \tag{9}$$

Let us note that we can choose D_Q finite whenever Q is compact. In this section we specialize the result of Theorem 3.4 for the case when Q has the following form: $Q = \{\lambda \in \mathbb{R}^n : \|\lambda\| \leq R\}$.

Theorem 3.5: [6] Assume that Λ^* is nonempty and bounded. Then, the sequence $\{\lambda^k\}_{k \geq 0}$ generated by Algorithm 3.2 is bounded.

Since Assumption 2.3 holds, then Λ^* is nonempty. Conditions under which Λ^* is bounded can be found in [10] (page 301). Under the assumptions of Theorem 3.5, it follows that there exists $R > 0$ sufficiently large such that the set $Q = \{\lambda \in \mathbb{R}^n : \|\lambda\| \leq R\}$ contains Λ^*, and thus we can assume D_Q to be finite. Notice that similar arguments were used in order to prove convergence of two-level algorithms for convex optimization problems (see e.g. [3]).

Denote $\alpha = 4\sqrt{D_Q(D_X + D_Z)\big(\frac{\|A\|^2}{\sigma_Q \sigma_X} + \frac{\|B\|^2}{\sigma_Q \sigma_Z}\big)}$. The next theorem shows how to choose optimally the smoothness parameter c and provides the complexity estimates of our method when Q is a ball. Our proof follows a similar reasoning as in [8], but requires more complex arguments.

Theorem 3.6: Assume that there exists R such that the set $Q = \{\lambda \in \mathbb{R}^n : \|\lambda\| \leq R\}$ contains a $\lambda^* \in \Lambda^*$. Taking $c = \frac{2}{k+1}\sqrt{\frac{D_Q}{D_X + D_Z}\big(\frac{\|A\|^2}{\sigma_Q \sigma_X} + \frac{\|B\|^2}{\sigma_Q \sigma_Z}\big)}$, then after k iterations we obtain an approximate solution to the problem (1) $(\hat{x}, \hat{z}) = \sum_{l=0}^{k} \frac{2(l+1)}{(k+1)(k+2)}(x^{l+1}, z^{l+1})$ and $\hat{\lambda} = \lambda^k$ which satisfy the following bounds on the duality gap and constraints:

$$0 \leq [\phi_1(\hat{x}) + \phi_2(\hat{z})] - f_0(\hat{\lambda}) \leq \frac{\alpha}{k+1}$$
$$\|A\hat{x} + B\hat{z} - b\|_* \leq \frac{\alpha}{(R - \|\lambda^*\|_*)(k+1)}.$$

Proof: Using (9) and the form of Q we obtain that

$$\max_{\lambda \in Q} -\frac{4L_c}{\sigma_Q(k+1)^2} d_Q(\lambda) + \langle A\hat{x} + B\hat{z} - b, \lambda \rangle \geq$$
$$-\frac{4L_c D_Q}{\sigma_Q(k+1)^2} + \max_{\|\lambda\| \leq R} \langle A\hat{x} + B\hat{z} - b, \lambda \rangle =$$
$$-\frac{4L_c D_Q}{\sigma_Q(k+1)^2} + R\|A\hat{x} + B\hat{z} - b\|_*.$$

In view of the previous relation and Theorem 3.4 we obtain the following duality gap:

$$[\phi_1(\hat{x}) + \phi_2(\hat{z})] - f_0(\hat{\lambda}) \leq c(D_X + D_Z) + \frac{4L_c D_Q}{\sigma_Q(k+1)^2} -$$
$$R\|A\hat{x} + B\hat{z} - b\|_* \leq c(D_X + D_Z) + \frac{4L_c D_Q}{\sigma_Q(k+1)^2}$$

Minimizing the right-hand side of this inequality over c we get the above expressions for c and for the upper bound on the duality gap from the theorem. Moreover, for the constraints using Lemma 3.3 and inequality (7) we have that

$$(R - \|\lambda^*\|_*)\|A\hat{x} + B\hat{z} - b\|_* \leq c(D_X + D_Z) + \frac{4L_c D_Q}{\sigma_Q(k+1)^2}$$

and replacing c derived above we also get the bound on the constraints violation. ■

From Theorem 3.6 we obtain that the complexity for finding an (ϵ, ϵ_c)-approximation of the optimum f^*, when the set Q is a ball, is $k + 1 = 4\sqrt{D_Q(D_X + D_Z)\big(\frac{\|A\|^2}{\sigma_Q \sigma_X} + \frac{\|B\|^2}{\sigma_Q \sigma_Z}\big)}\,\frac{1}{\epsilon}$, i.e. the efficiency estimates of our scheme is of the order $\mathcal{O}(\frac{1}{\epsilon})$, better than most non-smooth optimization schemes such as the subgradient method that have an efficiency estimate of the order $\mathcal{O}(\frac{1}{\epsilon^2})$ (see e.g. [7]). The dependence of the parameters c and L_c on ϵ is as follows: $c = \frac{\epsilon}{2(D_X + D_Z)}$ and $L_c = \big(\frac{\|A\|^2}{\sigma_X} + \frac{\|B\|^2}{\sigma_Z}\big)\frac{D_X + D_Z}{2\epsilon}$. The main advantage of our scheme is that it is fully automatic, the parameter c is chosen unambiguously, which is crucial for justifying the convergence properties of Algorithm 3.2. Another advantage of the proximal center method is that we are free in the choice of the norms in the spaces $\mathbb{R}^n, \mathbb{R}^m$ and $\mathbb{R}^p$, while most of the decomposition schemes are based on the Euclidean norm. Thus, we can choose the norms which make the ratio $\frac{\|A\|^2}{\sigma_Q \sigma_X}$ as small as possible.

D. Efficiency estimates for the Euclidean norm

In this section we assume the Euclidean norm on $\mathbb{R}^n$.

Theorem 3.7: Assume that $Q=\mathbb{R}^n$ and $d_Q(\lambda)=\frac{1}{2}\|\lambda\|^2$, with the Euclidean norm on $\mathbb{R}^n$. Taking $c=\frac{\epsilon}{D_X+D_Z}$ and $k+1=2\sqrt{\big(\frac{\|A\|^2}{\sigma_X}+\frac{\|B\|^2}{\sigma_Z}\big)(D_X+D_Z)}\ \frac{1}{\epsilon}$, then after k iterations the duality gap is less than ϵ and the constraints satisfy $\|A\hat{x}+B\hat{z}-b\|\le\epsilon\big(\|\lambda^*\|+\sqrt{\|\lambda^*\|^2+2}\big)$.

Proof: Let us note that for these choices for Q and d_Q we have $\sigma_Q=1$ and thus

$$\max_{\lambda\in\mathbb{R}^n}-\frac{4L_c}{\sigma_Q(k+1)^2}d_Q(\lambda)+\langle A\hat{x}+B\hat{z}-b,\lambda\rangle=\frac{(k+1)^2}{8L_c}\|A\hat{x}+B\hat{z}-b\|^2.$$

We obtain the following bound on the duality gap (Th. 3.4):

$$[\phi_1(\hat{x})+\phi_2(\hat{z})]-f_0(\hat{\lambda})\le c(D_X+D_Z)-\frac{(k+1)^2}{8L_c}\|A\hat{x}+B\hat{z}-b\|^2\le c(D_X+D_Z).$$

It follows that taking $c=\frac{\epsilon}{D_X+D_Z}$, the duality gap is less than ϵ. For the constraints using Lemma 3.3 and inequality (7) we get that $\|A\hat{x}+B\hat{z}-b\|$ satisfies the second order inequality in y: $\frac{(k+1)^2}{8L_c}y^2-\|\lambda^*\|y-\epsilon\le 0$. Therefore, $\|A\hat{x}+B\hat{z}-b\|$ must be less than the largest root of the corresponding second-order equation, i.e.

$$\|A\hat{x}+B\hat{z}-b\|\le\big(\|\lambda^*\|+\sqrt{\|\lambda^*\|^2+\frac{\epsilon(k+1)^2}{2L_c}}\big)\frac{4L_c}{(k+1)^2}.$$

After some long but straightforward computations we get that after k iterations, where k defined as in the theorem, we also get $\|A\hat{x}+B\hat{z}-b\|\le\epsilon\big(\|\lambda^*\|+\sqrt{\|\lambda^*\|^2+2}\big)$. ■

Remark 3.8 When coupling inequalities $Ax+Bz-b\le 0$ are present, then we choose $Q=\mathbb{R}^n_+$. Using the same reasoning as before we get that $\max_{\lambda\ge0}-\frac{4L_c}{\sigma_Q(k+1)^2}d_Q(\lambda)+\langle A\hat{x}+B\hat{z}-b,\lambda\rangle=\frac{(k+1)^2}{8L_c}\|[A\hat{x}+B\hat{z}-b]^+\|^2$, where $[y]^+$ denotes the projection of $y\in\mathbb{R}^n$ onto $\mathbb{R}^n_+$. Taking for c and k the same values as in Theorem 3.7, we can conclude that after k iterations the duality gap is less than ϵ and using a modified version of Lemma 3.3 (i.e. a generalized version of Cauchy-Schwarz inequality: $-\langle y,\lambda\rangle\ge-\|[y]^+\|\ \|\lambda\|$ for any $y\in\mathbb{R}^n,\lambda\ge0$) the constraints violation satisfy $\|[A\hat{x}+B\hat{z}-b]^+\|\le\epsilon\big(\|\lambda^*\|+\sqrt{\|\lambda^*\|^2+2}\big)$.

IV. Applications

A. Applications with separable structure

In this section we briefly discuss some of the applications to which our method can be applied.

First application that we will discuss here is the control of large scale systems with interacting subsystem dynamics. A distributed MPC framework is appealing in this context since this framework allows us to design local subsystem-based controllers that take care of the interactions between different subsystems and physical constraints. We assume that the overall system model can be decomposed into M appropriate subsystem models:

$$x^i(k+1)=\sum_{j\in\mathcal{N}(i)}A_{ij}x^j(k)+B_{ij}u^j(k)\ \ \forall i=1\cdots M,$$

where $\mathcal{N}(i)$ denotes the set of subsystems that interact with the ith subsystem, including itself. The control and state sequence must satisfy local constraints

$$x^i(k)\in\Omega_i,\ u^i(k)\in U_i\ \ \forall i=1\cdots M \text{ and } \forall k\ge0,$$

where the sets Ω_i and U_i are usually convex compact sets with the origin in their interior. In general the control objective is to steer the state of the system to origin or any other set point in a "best" way. Performance is expressed via a stage cost, which in this paper we assume to have the following form (see also [13]): $\sum_{i=1}^M\ell_i(x^i,u^i)$, where usually ℓ_i is a convex quadratic function, but not necessarily strictly convex. Let N denote the prediction horizon. In MPC we must solve at each step k, given $x^i(k)=x^i$, a finite-horizon optimal control problem of the following form:

$$\begin{aligned}&\min_{x_l^i,u_l^i}\sum_{l=0}^{N-1}\sum_{i=1}^{M}\ell_i(x_l^i,u_l^i)\\ &\text{s.t.: } x_0^i=x^i,\ x_{l+1}^i=\sum_{j\in\mathcal{N}(i)}A_{ij}x_l^j+B_{ij}u_l^j\\ &x_N^i\in\Omega_i,\ x_l^i\in\Omega_i,\ u_l^i\in U_i\ \forall l=0\cdots N-1,\ \forall i=1\cdots M\end{aligned}\tag{10}$$

Note that a similar formulation of distributed MPC for coupled linear subsystems with decoupled costs was given in [13], but without state constraints (i.e. without imposing $x^i(k)\in\Omega_i$). Let us introduce $\mathbf{x}_i=(x_0^i\cdots x_N^i\ u_0^i\cdots u_{N-1}^i), X_i=\Omega_i^{N+1}\times U_i^N$ and $\psi_i(\mathbf{x}^i)=\sum_{l=0}^{N-1}\ell_i(x_l^i,u_l^i)$. Then, the control problem (10) can be recast as a separable convex program:

$$\min_{\mathbf{x}_i\in X_i}\big\{\sum_{i=1}^M\psi_i(\mathbf{x}_i):\sum_{i=1}^M C_i\mathbf{x}_i-\gamma=0\big\},\tag{11}$$

where the matrices C_i's are defined appropriately.

Network optimization furnishes another areas in which our Algorithm 3.2 leads to a new method of solution. In this application the convex optimization problem has the following form [9], [14]:

$$\min_{x_i\in X_i}\big\{\sum_{i=1}^M\psi_i(x_i):\sum_{i=1}^M C_ix_i-\gamma=0,\ \sum_{i=1}^M D_ix_i-\beta\le0\big\},\tag{12}$$

where X_i are compact convex sets (in general balls) in $\mathbb{R}^m$, ψ_i's are non-strictly convex functions and M denotes the number of agents in the network. Note that (11) is a particular case of the separable convex problem (12).

In [13] the optimization problem (10) (or equivalently (11)) was solved in a decentralized fashion, iterating the Jacobi algorithm $p_{\max}$ times [1]. But, there is no theoretical guarantee of the Jacobi algorithm about how good is the approximation of the optimum after $p_{\max}$ iterations and moreover we need strictly convex functions ψ_i to prove asymptotic convergence to the optimum. However, if we solve (10) using our Algorithm 3.2, we have a guaranteed upper bound on the approximation after $p_{\max}$ iterations (see Theorem 3.4). In [9], [14] the optimization problem (12) is

solved using the dual subgradient method described in the Algorithm 2.2. From the simulation tests of Section IV-B we see that our Algorithm 3.2 is superior to Algorithm 2.2.

Let us describe briefly the main ingredients of Algorithm 3.2 for problem (12). Let d_{X_i} be properly chosen prox-functions, according to the structure of the sets X_i's and the norm[1] used on $\mathbb{R}^m$. In this case the smooth dual function f_c has the form:

$$f_c(\lambda) = \min_{x_i \in X_i} \sum_{i=1}^{M} \psi_i(x_i) + \langle \lambda_1, \sum_{i=1}^{M} C_i x_i - \alpha \rangle + \langle \lambda_2, \sum_{i=1}^{M} D_i x_i - \beta \rangle + c \sum_{i=1}^{M} d_{X_i}(x_i).$$

Moreover, $Q \subseteq (\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}_+)$, where n_1 and n_2 denote the number of equality and inequality constraints. It is worth noting that in this case all the minimization problems involved in Algorithm 3.2 are decomposable in x_i and thus the agents can solve the optimization problem (12) in a decentralized fashion.

B. Computational results

We conclude this paper with the results of computational experiments on a random set of problems of the form (12), where ψ_i's are convex quadratic functions (but not strictly convex) and X_i's are balls in $\mathbb{R}^m$ defined with the Euclidean norm. Similarly, $\mathbb{R}^n$ (corresponding to the Lagrange multipliers) is endowed with the Euclidean norm.

m	nr iter Alg 3.2	nr iter Alg 2.2
50	1384(0.01)	5000(0.31)
100	3289(0.01)	5000(0.49)
1000	7312(0.01)	10000(0.57)
50	5000(0.0011)	5000(0.31)
100	5000(0.004)	5000(0.49)
1000	10000(0.007)	10000(0.57)

In the table we display the number of iterations of the Algorithm 2.2 and 3.2 for different values of $m = 50, 100$ and 1000, and for $M = 10$ agents. For $m = 50$ and $m = 100$ the maximum number of iterations that we allow is 5000. For $m = 1000$ we iterate at most 10000 times. We also display between brackets the corresponding accuracy. We see that the accuracy of the approximation of the optimum is much better with our Algorithm 3.2 than with Algorithm 2.2.

V. CONCLUSIONS

A new decomposition method in convex programming is developed in this paper using the framework of dual decomposition. Our method combines the computationally non-expensive cost of the gradient method with the efficiency of structural optimization for solving non-smooth separable convex programs. Although our decomposition method resembles proximal based methods, it differs both in the computational steps and in the choice of the parameters of the method. Contrary to most proximal based methods that enforce the next iterates to be close to the previous ones, our method uses fixed centers in the prox-terms which leads to more freedom in the next iterates. Another advantage of our proximal center method is that it is fully automatic, i.e. the parameters of the scheme are chosen optimally, which are crucial for justifying the convergence properties of the new scheme. We presented efficiency estimate results of the new decomposition method for general separable convex problems. We also show that our algorithm can be used to solve network optimization or distributed model predictive control problems. The computations on some test problems confirm that the proposed method works well in practice.

[1]For example if $X_i = \{x \in \mathbb{R}^m : \|x - x_0\|_2 \leq R\}$, where $\|\cdot\|$ denotes here the Euclidean norm, then it is natural to take $d_{X_i}(x) = \frac{\|x-x_0\|_2^2}{2}$. If $X_i = \{x \in \mathbb{R}^m : x \geq 0, \ \sum_{i=1}^m x_{(i)} = 1\}$, then the norm $\|x\| = \sum_{i=1}^m |x_{(i)}|$ and $d_{X_i}(x) = \ln m + \sum_{i=1}^m x_{(i)} \ln x_{(i)}$ is more suitable (see [8] for more details).

Acknowledgment. We acknowledge the financial support by Research Council K.U. Leuven: GOA AMBioRICS, CoE EF/05/006, OT/03/12, PhD/postdoc & fellow grants; Flemish Government: FWO PhD/postdoc grants, FWO projects G.0499.04, G.0211.05, G.0226.06, G.0302.07; Research communities (ICCoS, ANMMM, MLDM); AWI: BIL/05/43, IWT: PhD Grants; Belgian Federal Science Policy Office: IUAP DYSCO.

REFERENCES

[1] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical Methods*. Prentice-Hall, Englewood Cliffs, NJ, 1989.

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

[3] G. Cohen. Optimization by decomposition and coordination: A unified approach. *IEEE Transactions on Automatic Control*, AC–23(2):222–232, 1978.

[4] J. Eckstein and D. Bertsekas. On the Douglas–Rachford splitting method and the proximal point method for maximal monotone operators. *Mathematical Programming (A)*, 55(3):293–318, 1992.

[5] S. Kontogiorgis, R. De Leone, and R. Meyer. Alternating direction splitings for block angular parallel optimization. *Journal of Optimization Theory and Applications*, 90(1):1–29, 1996.

[6] I. Necoara and J.A.K. Suykens. Application of a smoothing technique to decomposition in convex programming. Technical Report 08–07, ESAT-SISTA, K.U. Leuven (Leuven, Belgium), January 2008.

[7] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

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

[9] D. Palomar and M. Chiang. Alternative distributed algorithms for network utility maximization: Framework and applications. *IEEE Transactions on Automatic Control*, 52(12):2254–2268, 2007.

[10] R. T. Rockafellar. *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, 1970.

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

[12] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal of Control and Optimization*, 29(1):119–138, 1991.

[13] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright. Distributed MPC strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, to appear, 2007.

[14] L. Xiao, M. Johansson, and S. Boyd. Simultaneous routing and resource allocation via dual decomposition. *IEEE Transactions on Communications*, 52(7):1136–1144, 2004.