52nd IEEE Conference on Decision and Control
December 10-13, 2013. Florence, Italy

A Dual Decomposition Algorithm for Separable Nonconvex Optimization Using the Penalty Function Framework

Quoc Tran Dinh[†,*], Ion Necoara[‡,§] and Moritz Diehl[†]

***Abstract*— We propose a dual decomposition method for solving separable *nonconvex* optimization problems that arise e.g. in distributed model predictive control over networks. We first derive a new sequential convex programming (SCP) scheme based on penalty function approach to handle nonconvexity. Then, we combine this SCP scheme with a dual decomposition algorithm to obtain a two-level decomposition algorithm. The global convergence of this algorithm is analyzed under standard assumptions. Some preliminary numerical results are also given to illustrate the theoretical results.**

I. Introduction

For the synthesis of large-scale network systems, centralized control is often impractical and computationally too demanding for online implementation. Consider for example an electric distribution network or a transportation network. Designing a centralized controller for these systems is not practically viable. Gathering to one single place all the data about the current state of such systems and calculating all the optimal control inputs from a single problem is clearly a very difficult task. Therefore, in the last decades, distributed 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 and its capability of computing the optimal control input distributively. When we deal with nonlinear network systems, one can usually formulate the control problem as a separable nonconvex optimization problem.

Decomposition methods represent a powerful tool for solving large-scale optimization problems arising in distributed control, estimation and other engineering disciplines. The basic idea is to decompose the given problem into smaller subproblems, which then can be coordinated by a master problem. In large-scale separable convex optimization, Lagrangian dual decomposition has gained popularity. Thanks to strong duality, this approach produces primal solutions in the limit or approximate primal solutions with tight estimates on the convergence rate, see e.g. [1], [7], [8], [16], [17].

Unlike convex optimization, strong duality does no longer hold in the nonconvex case in general. Decomposition approach for nonconvex problems encounters many difficulties [2], [4], [5], [9], [18]. The first difficulty is the lack of zero duality gap in primal-dual formulations. The second relates to numerical solution methods, where globalization strategies such as line-search or trust-region are hard to implement in a distributed manner. As opposed to those, penalty or augmented Lagrangian methods are more appropriate for our problems. However, the penalty methods usually lead to a nonsmoothness of the resulting problem, while the augmented Lagrangian methods possess a crossproduct term destroying the separable structure. To overcome such difficulties, Bertsekas in [2] proposed a convexification procedure using proximal point techniques. The authors in [15] approximated the crossproduct term by linearization but led to a complex and poorly efficient method. Tanikawa et al. in [18] combined the well-known Fletcher augmented Lagrangian with a decoupling technique. Tatjewski further exploited this technique to obtain a two-level decomposition method which has local convergence [19]. Hamdi [4] combined the augmented Lagrangian and proximal point techniques to design a decomposition algorithm for separable nonconvex optimization. Necoara in [9] proposed a decomposition scheme based on sequential convex programming and smoothing techniques. While several works mentioned here deal with augmented Lagrangian functions, we consider in this paper a penalty function approach as an alternative.

Contribution: We propose a sequential convex programming (SCP) scheme relying on a penalty function approach for separable nonconvex optimization. Since the subproblem rendered at each iteration is separable and strongly convex, our recent work in dual decomposition [7], [8], [16], [17] can be applied to solve such a problem and leads to a two-level decomposition algorithm suitable for parallel and distributed implementation. The convergence of the SCP scheme as well as the two-level algorithm is analyzed under standard assumptions in nonconvex optimization. The main contribution is the SCP scheme and its global convergence guarantee, which is relatively new.

Paper outline: Motivated by distributed control for nonlinear network systems from Section II, in Section III we provide a sequential convex programming scheme to tackle nonconvex problems and prove its global convergence. In Section IV, we present a two-level decomposition algorithm and analyze its properties. Section V deals with preliminary numerical tests to verify the proposed algorithm.

II. Motivation: distributed control of nonlinear network systems

We consider discrete-time network systems, possibly nonlinear, which are usually modeled by a graph whose nodes

[†]Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center (OPTEC), KU Leuven, Belgium. Email: {quoc.trandinh, moritz.diehl}@esat.kuleuven.be

[‡]Automatic Control and Systems Engineering Department, University Politehnica Bucharest, Romania. Email: ion.necoara@acse.pub.ro

[*]Laboratory for Information and Inference Systems (LIONS), EPFL, Switzerland. Email: quoc.trandinh@epfl.ch

[§] Corresponding author: Ion Necoara.

978-1-4673-5716-6/13/$31.00 ©2013 IEEE

represent subsystems and whose arcs indicate dynamic couplings, defined by the following difference equations:

$$x_i^{t+1} = \Phi_i\Big(x_i^t, u_i^t, \sum_{j\in\mathcal{N}^i} A_{ij}x_j^t + B_{ij}u_j^t\Big),\ \forall i = 1,\dots,M. \quad (1)$$

The vectors $x_i^t \in \mathbb{R}^{n_{x_i}}$, $u_i^t \in \mathbb{R}^{n_{u_i}}$ represent the state and the input of the subsystem i at time t. The index set $\mathcal{N}^i$ contains all the indices of the subsystems which interact with the subsystem i. We further assume that the inputs and state vectors must satisfy local mixed constraints:

$$(x_i^t, u_i^t) \in \mathbb{X}_i,\quad \forall t \geq 0,\quad x_i^N \in \mathbb{X}_i^N,\quad \forall i = 1,\dots,M, \quad (2)$$

where $\mathbb{X}_i \subseteq \mathbb{R}^{n_{x_i}} \times \mathbb{R}^{n_{u_i}}$ are convex compact sets. The system performance is expressed via a stage and a final cost $\ell_i(x_i, u_i)$ and $\ell_i^f(x_i)$, which are convex for each subsystem i. The centralized optimal control problem for a prediction horizon of length N and initial state $\zeta = [\zeta_1^T \dots \zeta_M^T]^T$ reads:

$$\min_{x_i^t, u_i^t} \sum_{i=1}^{M}\sum_{t=0}^{N-1} \ell_i(x_i^t, u_i^t) + \sum_{i=1}^{M} \ell_i^f(x_i^N), \quad (3)$$

$$\text{s.t. } x_i^{t+1} = \Phi_i\Big(x_i^t, u_i^t, \sum_{j\in\mathcal{N}^i} A_{ij}x_j^t + B_{ij}u_j^t\Big),\ x_i^0 = \zeta_i \quad (3.1)$$

$$(x_i^t, u_i^t) \in \mathbb{X}_i, \forall t \geq 0,\ x_i^N \in \mathbb{X}_i^N,\ \forall i = 1,\dots,M. \quad (3.2)$$

A similar formulation of distributed control but for coupled linear subsystems with decoupled costs was given in [7], [20] in the context of MPC. See also [3], [6], [13] for recent surveys of distributed and hierarchical MPC methods. Problem (3) becomes interesting if the computations can be distributed among the subsystems (agents) and the amount of information that the agents must exchange is limited.

Now, we show that problem (3) can be recast as a separable optimization problem but with a particular structure. For each i, we introduce the following auxiliary variables:

$$\tilde{x}_i^t := \sum_{j\in\mathcal{N}^i} A_{ij}x_j^t + B_{ij}u_j^t \in \mathbb{R}^{n_{y_i}}. \quad (4)$$

Denoting by

$$x_i = [(u_i^0)^T\ (x_i^0)^T\ (u_i^1)^T\ (x_i^1)^T\ (\tilde{x}_i^1)^T \cdots (u_i^{N-1})^T\ (x_i^{N-1})^T\ (\tilde{x}_i^{N-1})^T\ (x_N^i)^T]^T,$$

where $x_i \in \mathbb{R}^{n_i}$ and $x = \left[{x_1}^T \cdots {x_M}^T\right]^T \in \mathbb{R}^n$, the convex set $X = X_1 \times \cdots \times X_M \subseteq \mathbb{R}^n$, where each set

$$X_i = \mathbb{X}_i \times \underbrace{(\mathbb{X}_i \times \mathbb{R}^{n_{y_i}}) \times \cdots \times (\mathbb{X}_i \times \mathbb{R}^{n_{y_i}})}_{N-1 \text{ times}} \times \mathbb{X}_i^N \subseteq \mathbb{R}^{n_i}$$

is also convex and the convex functions

$$g_i(x_i) = \sum_{t=0}^{N-1} \ell_i(x_i^t, u_i^t) + \ell_i^f(x_i^N),\quad g(x) = \sum_{i=1}^{M} g_i(x_i).$$

With this notation, problem (3) now reads

$$\min_{x_1\in X_1,\cdots,x_M\in X_M} \sum_{i=1}^{M} g_i(x_i) \quad (5)$$

$$\text{s.t. } \sum_{i=1}^{M}(A_ix_i - b_i) = 0,\ F_i(x_i) = 0,\ \forall i = 1,\dots,M,$$

where the functions $F_i(x_i)$ are obtained by stacking the constraints (3.1) for a given i and $\sum_{i=1}^M (A_ix_i - b_i) = 0$ is deduced from (4). In a more compact form we can write:

$$\begin{aligned} \min_{x\in X}\quad & g(x) \\ \text{s.t.}\quad & Ax - b = 0,\ F(x) = 0, \end{aligned} \quad (6)$$

where $F : \mathbb{R}^n \to \mathbb{R}^m$ is obtained by stacking all the functions F_i and $Ax - b = \sum_{i=1}^M (A_ix_i - b_i)$. Notice that the nonconvexity in problems (5) and (6) is concentrated in the local equality constraints $F_i(x_i) = 0$ for all $i = 1,\dots,M$, while the objective function g is convex.

III. Sequential convex programming approach for separable nonconvex optimization

Let $\eta_i : \mathbb{R}^{m_i} \to \mathbb{R}\cup\{+\infty\}$ be nonnegative, strictly convex functions such that $\eta_i(0) = 0$ and $\lim_{\|v_i\|\to+\infty} \eta_i(v_i) = +\infty$ for $i = 1,\dots,M$. We consider the following penalized optimization problem of (6):

$$\phi^* := \min_{x\in\Omega} \{\phi(x) := g(x) + \rho\eta(F(x))\}, \quad (7)$$

where $\eta(v) = \sum_{i=1}^M \eta_i(v_i)$, $\rho > 0$ is a penalty parameter, and Ω is defined as:

$$\Omega = \{x \in X \mid Ax = b\}. \quad (8)$$

As an example, we can choose η as an ℓ_1-penalty function, i.e. $\eta(v) = \sum_{i=1}^M \|v_i\|_1 = \|v\|_1$. We assume that for $\rho > 0$, ϕ^* is finite. From the theory of exact penalty functions, see, e.g., [14], it is well-known that if x^* is a local minimizer of (6) and the second-order sufficient conditions are satisfied, then there exists a constant $\bar{\rho} > 0$ such that for all $\rho > \bar{\rho}$ the point x^* is a strictly local minimizer of the penalized problem (7). Conversely, if x^* is a local minimizer of (7) such that $F(x^*) = 0$, then it is also a local minimizer of (6). Practical methods for updating ρ can be found, e.g., in [10].

In fact, problem (7) can be written in a separable form as

$$\left\{\begin{aligned} \min_{x_i\in X_i}\quad & \sum_{i=1}^{M} g_i(x_i) + h_i(F_i(x_i)), \\ \text{s.t.}\quad & \sum_{i=1}^{M}(A_ix_i - b_i) = 0, \end{aligned}\right. \quad \text{(SepNCOP)}$$

where $h_i(\cdot) = \rho\eta_i(\cdot)$ and $h = \sum_{i=1}^M h_i$. The functions g_i and h_i are assumed to be proper, lower semicontinuous and convex. In addition, g_i is possibly smooth, while h_i is not necessarily smooth. The function $F_i : \mathbb{R}^{n_i} \to \mathbb{R}^{m_i}$ is continuously differentiable on its domain. As we will see later a good estimation for the penalty parameter ρ leads to a smaller Lipschitz constant of h_i in (SepNCOP).

A. Optimality condition

The optimality condition for (SepNCOP) becomes

$$0 \in \partial g(x^*) + F'(x^*)^T \partial h(F(x^*)) + \mathcal{N}_\Omega(x^*), \quad (9)$$

where $\mathcal{N}_\Omega(x)$ is the normal cone of the convex set Ω at x, $\partial g(\cdot)$ and $\partial h(\cdot)$ are the subdifferential of g and h, respectively, and F' is the Jacobian mapping of F. The condition (9) can be written equivalently as

$$\partial h(F(x^*)) \cap \left\{v \mid -F'(x^*)^T v \in \partial g(x^*) + \mathcal{N}_\Omega(x^*)\right\} \neq \emptyset.$$

Any point x^* satisfying (9) is called a stationary point. We denote by Ω^* the set of stationary points of (SepNCOP). Instead of (9), we consider an approximate optimality condition for (SepNCOP) as follows

$$[\xi_g + F'(\tilde{x}^*)^T\xi_h]^T(u - \tilde{x}^*) \geq -\varepsilon \quad \forall u \in \Omega, \tag{10}$$

where $\varepsilon \geq 0$ is a given tolerance, $\xi_g \in \partial g(\tilde{x}^*)$ and $\xi_h \in \partial h(F(\tilde{x}^*))$ are subgradients of g at $\tilde{x}^*$ and of $h(F(\cdot)$ at $F(\tilde{x}^*)$, respectively. In this case $\tilde{x}^*$ is called an ε-*approximate stationary point* of (SepNCOP).

B. Sequential convex programming scheme

Let $\mathcal{D}$ be a closed convex set in $\mathbb{R}^n$ with nonempty interior such that $\Omega \subseteq \mathcal{D}$. We make the following assumptions:

Assumption 1: The function g is convex in $\mathcal{D}$. The function h is convex and L_h-Lipschitz continuous in $\mathbb{R}^m$, i.e., there exists $L_h > 0$ such that

$$|h(u) - h(v)| \leq L_h\|u - v\|, \quad \forall u, v \in \mathbb{R}^m.$$

The function F is differentiable in $\mathcal{D}$ and its Jacobian mapping is $L_{F'}$-Lipschitz continuous in $\mathcal{D}$, i.e., there exists $L_{F'}$ such that

$$\|F'(x) - F'(\hat{x})\| \leq L_{F'}\|x - \hat{x}\|, \quad \forall x, \hat{x} \in \mathcal{D}.$$

As a simple example, the function $h(u) = \rho\|u\|$ is convex and Lipschitz continuous with a Lipschitz constant $L_h = \rho$ on $\mathbb{R}^m$ for any $\rho > 0$.

For a given $x^k \in \mathcal{D}$, let us define

$$\psi(x; x^k) = g(x) + h\left(F(x^k) + F'(x^k)(x - x^k)\right). \tag{11}$$

Since g and h are convex, $\psi(\cdot; x^k)$ is also convex. If, in addition, g is differentiable and its gradient is $L_{g'}$-Lipschitz continuous in $\mathcal{D}$, then instead of ψ, we can consider

$$\begin{aligned}\psi_L(x; x^k) = g(x^k) &+ \nabla g(x^k)^T(x - x^k)\\ &+ h\left(F(x^k) + F'(x^k)(x - x^k)\right).\end{aligned} \tag{12}$$

We have the following estimates.

Lemma 1: Under Assumption 1, the function $\psi(\cdot; x^k)$ defined by (11) satisfies

$$\left|\phi(x) - \psi(x; x^k)\right| \leq \frac{M_\psi}{2}\|x - x^k\|^2, \ \forall x \in \mathcal{D}, \tag{13}$$

where $M_\psi = L_h L_{F'} > 0$.

If, in addition, g is differentiable and its gradient is $L_{g'}$-Lipschitz continuous in $\mathcal{D}$ then the following estimate holds

$$\left|\phi(x) - \psi_L(x; x^k)\right| \leq \frac{M_{\psi_L}}{2}\|x - x^k\|^2, \ \forall x \in \mathcal{D}, \tag{14}$$

where $M_{\psi_L} := L_{g'} + L_h L_{F'} > 0$.

Proof: Since F' is $L_{F'}$-Lipschitz continuous, for any $x,\ x^k \in \mathcal{D}$, we have $\|F(x) - F(x^k) - F'(x^k)(x - x^k)\| \leq \frac{1}{2}L_{F'}\|x - x^k\|^2$. By using this estimate and the Lipschitz continuity of h, we have

$$\begin{aligned}\left|\phi(x) - \psi(x; x^k)\right| &= \left|h\left(F(x)\right) - h\left(F(x^k) + F'(x^k)(x - x^k)\right)\right|\\ &\leq L_h\|F(x) - F(x^k) - F'(x^k)(x - x^k)\|\\ &\leq \frac{1}{2}L_h L_{F'}\|x - x^k\|^2,\end{aligned}$$

which is indeed (13). Similarly, by the Lipschitz continuity of ∇g we have

$$\begin{aligned}&\left|\phi(x) - \psi_L(x; x^k)\right| \leq \left|g(x) - g(x^k) - \nabla g(x^k)^T(x - x^k)\right|\\ &+ \left|h(F(x)) - h\left(F(x^k) + F'(x^k)(x - x^k)\right)\right|\\ &\leq \frac{L_{g'}}{2}\|x - x^k\|^2 + L_h\|F(x) - F(x^k) - F'(x^k)(x - x^k)\|\\ &\leq \frac{L_{g'} + L_h L_{F'}}{2}\|x - x^k\|^2,\end{aligned}$$

which proves (14). ∎

For simplicity of presentation, we will use ψ defined by (11) in the following SCP scheme. However, the obtained results remain true if we replace ψ by ψ_L defined by (12) with simple modifications.

For $x^k \in \Omega$, we also define

$$\begin{aligned}q(x; x^k, \beta) &= \psi(x; x^k) + \frac{\beta}{2}\|x - x^k\|^2\\ v_0(x^k; \beta) &= \operatorname*{argmin}_{x \in \Omega} q(x; x^k, \beta).\end{aligned} \tag{15}$$

Since $q(\cdot; x^k, \beta)$ is strongly convex with a parameter $\beta > 0$, v_0 is well-defined and single-valued. The optimality condition for problem (15) becomes

$$\xi_q^T(u - v_0(x^k; \beta)) \geq 0, \quad \forall u \in \Omega,$$

for some $\xi_q \in \partial q(v_0(x^k; \beta); x^k, \beta)$, which is the necessary and sufficient optimality condition of (15). Now, we define the *proximal-gradient mapping* of function $q(\cdot; x^k, \beta)$ as

$$G_0(x^k; \beta) = \beta(x^k - v_0(x^k; \beta)), \tag{16}$$

We also define the error norm and the optimal value of (15), respectively, as

$$\begin{aligned}&e_0(x^k; \beta) = \|x^k - v_0(x^k; \beta)\|\\ &\varphi_0(x^k; \beta) = \psi(v_0(x^k; \beta); x^k) + \frac{\beta}{2}\|v_0(x^k; \beta) - x^k\|^2.\end{aligned}$$

In practice, we can not solve problem (15) *exactly*. We can only solve this problem up to a given accuracy $\varepsilon > 0$ to get

$$v_\varepsilon(x^k; \beta) \approx \operatorname*{argmin}_{x \in \Omega} q(x; x^k, \beta), \tag{17}$$

as in the sense of the following definition.

Definition 3.1: For a given tolerance $\varepsilon \geq 0$, a point v_ε is said to be an ε-solution to (15) if

$$\xi_{q\varepsilon}^T(u - v_\varepsilon) \geq -\varepsilon, \quad \forall u \in \Omega, \tag{18}$$

for some $\xi_{q\varepsilon} \in \partial q(v_\varepsilon; x^k, \beta)$, where $\partial q(v_\varepsilon; x^k, \beta)$ is the subdifferential of $q(\cdot; x^k, \beta)$ at v_ε, which can be computed as $\partial q(v_\varepsilon; x^k, \beta) = \partial g(v_\varepsilon) + F'(x^k)^T\partial h\left(F(x^k) + F'(x^k)(v_\varepsilon - x^k)\right) + \beta(v_\varepsilon - x^k)$.

Alternatively to G_0 and e_0, we define the *approximate proximal-gradient mapping*, the approximate error norm and the approximate optimal value of (15), respectively by

$$
\begin{aligned}
&G_\varepsilon(x^k;\beta) = \beta(x^k - v_\varepsilon(x^k;\beta))\\
&e_\varepsilon(x^k;\beta) = \|x^k - v_\varepsilon(x^k;\beta)\| \qquad (19)\\
&\varphi_\varepsilon(x^k;\beta) = \psi(v_\varepsilon(x^k;\beta);x^k) + \frac{\beta}{2}\|v_\varepsilon(x^k;\beta) - x^k\|^2.
\end{aligned}
$$

Remark 1: The condition (18) does not include the inexact feasibility. However, this condition can be modified to handle the inexact feasibility of the constraint $x \in \Omega$, where the subdifferential of the indicator function δ_Ω for Ω is expressed as $\partial\delta_\Omega(x) = \mathcal{N}_\Omega(x)$. By adopting the ε-subdifferential definition, we will be able to handle the inexact feasibility of $x \in \Omega$.

The following statement is obvious and can be obtained directly from (18) and (10).

Lemma 2: If x^k is a fixed point of the mapping $v_\varepsilon(\cdot;\beta)$, i.e. $x^k = v_\varepsilon(x^k;\beta)$, then it is an ε- stationary point of (SepNCOP).

Now, we establish the main estimate, which will be used in the sequel.

Lemma 3: Suppose that Assumption 1 is satisfied. Let $x^k \in \Omega$ be given, and $\beta > 0$. Then, the point $v_\varepsilon(x^k;\beta)$ defined by (15) satisfies the following estimate

$$
\begin{aligned}
\phi(x^k) - \phi(v_\varepsilon(x^k;\beta)) &\geq \frac{2\beta - M_\psi}{2} e_\varepsilon(x^k;\beta)^2 - \varepsilon\\
&= \frac{2\beta - M_\psi}{2\beta^2}\|G_\varepsilon(x^k;\beta)\|_*^2 - \varepsilon, \quad (20)
\end{aligned}
$$

where $M_\psi := L_h L_{F'}$.

Proof: Let $v_\varepsilon := v_\varepsilon(x^k;\beta)$. Using (13), we have

$$
\phi(v_\varepsilon) \leq g(v_\varepsilon) + h\big(F(x^k) + F'(x^k)(v_\varepsilon - x^k)\big) + \frac{M_\psi}{2}\|v_\varepsilon - x^k\|^2.
$$

Now, since g and h are convex, for any v we have

$$
\begin{aligned}
&g(x^k) - g(v) \geq \xi_g(v)^T(x^k - v)\\
&h(F(x^k)) - h(F(x^k) + F'(x^k)(v - x^k)) \qquad (21)\\
&\geq (\xi_h(F))^T F'(x^k)(x^k - v),
\end{aligned}
$$

where $\xi_h(F) \in \partial h\big(F(x^k) + F'(x^k)(v - x)\big)$. Next, by letting $u = x^k$ into (18), then combining the result and (21), we obtain

$$
\begin{aligned}
&g(x^k) + h(F(x^k)) \geq g(v_\varepsilon) + h(F(x^k) + F'(x^k)(v_\varepsilon - x^k))\\
&+ \big[\xi_g(v_\varepsilon) + F'(x^k)^T\xi_h(F)\big]^T(x^k - v_\varepsilon) \qquad (22)\\
&\overset{(18)}{\geq} g(v_\varepsilon) + h(F(x^k) + F'(x^k)(v_\varepsilon - x^k)) + \beta\|v_\varepsilon - x^k\|^2 - \varepsilon.
\end{aligned}
$$

Now, by using the Lipschitz continuity of h and F' in Assumption 1, for any v, we have

$$
\begin{aligned}
&h(F(v)) - h(F(x^k) + F'(x^k)(v - x^k))\\
&\leq L_h\|F(v) - F(x^k) - F'(x^k)(v - x^k)\| \leq 0.5M_\psi\|v - x^k\|^2.
\end{aligned}
$$

Substituting this inequality with $v = v_\varepsilon$ into (22) we obtain

$$
g(x^k) + h(F(x^k)) - g(v_\varepsilon) - h(F(v_\varepsilon)) \geq \frac{2\beta - M_\psi}{2}\|v_\varepsilon - x^k\|^2 - \varepsilon.
$$

Finally, by using the definitions of $e_\varepsilon(x^k;\beta)$, $G_\varepsilon(x^k;\beta)$ and ϕ, it follows from the last inequality that (20) holds. ■

Next, we define a sublevel set of $\phi(\cdot)$ restricted to Ω as

$$
\mathcal{L}_\phi(\alpha) = \{x \in \Omega \mid \phi(x) \leq \alpha\}. \qquad (23)
$$

The following lemma can be proved similarly as in [11, Lemma 2.5].

Lemma 4: Suppose that $\mathcal{L}_\phi(\phi(x^k)) + \varepsilon\mathcal{B}(0,1) \subseteq \mathrm{int}(\mathcal{D})$ for some $\varepsilon \geq 0$. If $\beta \geq M_\psi$, then $v_\varepsilon(x^k;\beta) \in \mathcal{L}_\phi(\phi(x^k))$.

Now, we can describe our *SCP algorithm* as follows.

ALGORITHM 1: (SCP scheme)

Initialization: Fix $\mu \in (0,1)$. Choose $x^0 \in \Omega$, $\varepsilon_{\text{outer}} > 0$ and $\varepsilon_0 > 0$ sufficiently small.

Iteration: For $k = 0, 1, \ldots$, perform the following steps:

Step 1. Find β_k such that $M_\psi \leq \beta_k \leq 2M_\psi$.

Step 2. Compute $x^{k+1} = v_{\varepsilon_k}(x^k;\beta_k)$ from (17).

Step 3. If $\|x^{k+1} - x^k\| \leq \varepsilon_{\text{outer}}$, then terminate.

Step 4. Update $\varepsilon_{k+1} = \min\left\{\varepsilon_k, \frac{\mu M_\psi}{2}\|x^{k+1} - x^k\|^2\right\}$.

End.

The main step of Algorithm 1 is Step 2, where we need to solve a *separable convex optimization* subproblem. The stopping criterion at Step 3 is based on Lemma 2. The following theorem shows the convergence of Algorithm 1.

Theorem 1: Let $\{x^k\}_{k\geq 0}$ be a sequence generated by Algorithm 1. Then,

$$
\phi(x^k) - \phi^* \geq \frac{1}{2}(1 - \mu)M_\psi\sum_{j=k}^{\infty} e_{\varepsilon_j}(x^j;\beta_j)^2 - \varepsilon_k, \quad (24)
$$

where $\varepsilon_k \geq 0$ and ϕ^* is the optimal value of (SepNCOP) in $\mathcal{L}_\phi(\phi(x^0))$. Consequently, one has $\lim_{k\to\infty}\|x^{k+1} - x^k\| = 0$ and the set of limit points Ω^* of $\{x^k\}$ is either *empty* or *nonempty and connected*. Suppose further that $\mathcal{L}_\phi(\phi(x^0))$ is bounded from below. Then, every limit point of $\{x^k\}$ is a stationary point of (SepNCOP). Moreover, if Ω^* is finite, then $\{x^k\}_{k\geq 0}$ converges to a point $x^* \in \Omega^*$.

Proof: Since $\beta_k \geq M_\psi$, it follows from Lemma 3 that

$$
\phi(x^k) - \phi(x^{k+1}) \geq 0.5M_\psi e_{\varepsilon_k}(x^k;\beta_k)^2 - \varepsilon_k, \ \forall k \geq 0.
$$

Summing up this inequality from $j = k$ to $j = N$ and noting that $\varepsilon_k = \min\left\{\frac{\mu M_\psi}{2}\|x^k - x^{k-1}\|^2, \varepsilon_{k-1}\right\} \leq \frac{\mu M_\psi}{2}\|x^k - x^{k-1}\|^2$, we get

$$
\begin{aligned}
\phi(x^k) - \ \phi(x^{N+1}) &\geq \sum_{j=k}^{N-1} 0.5(1 - \mu)M_\psi e_{\varepsilon_j}(x^j;\beta_j)^2\\
&+ 0.5M_\psi e_{\varepsilon_N}(x^N;\beta_N)^2 + 0.5M_\psi e_{\varepsilon_k}(x^k;\beta_k)^2 - \varepsilon_k\\
&\geq 0.5(1 - \mu)M_\psi\sum_{j=k}^{N} e_{\varepsilon_j}(x^j;\beta_j)^2 - \varepsilon_k + \varepsilon_{N+1}, \qquad (25)
\end{aligned}
$$

which can be rewritten as

$$
\big(\phi(x^k) + \varepsilon_k\big) - \big(\phi(x^{N+1}) + \varepsilon_{N+1}\big) \geq \frac{(1 - \mu)M_\psi}{2}\sum_{i=k}^{N} e_{\varepsilon_j}(x^j;\beta_j)^2.
$$

Since the sequence $\{\phi(x^k) + \varepsilon_k\}$ is bounded from below and the sequence $\{\varepsilon_k\}$ does not increase, by passing to the limit as $N \to \infty$ in (25), we obtain (24). Next, we set $k = 0$ in (25) and passing to the limit as $N \to \infty$, we have $\sum_{j=0}^{\infty} e_{\varepsilon_j}(x^j; \beta_j)^2 < +\infty$. Therefore, $\lim_{k\to\infty} \|x^k - x^{k+1}\| = 0$, which shows that the set of limit points of $\{x^k\}_{k\geq 0}$ is either empty or nonempty and connected.

Since $\mathcal{L}_\phi(\phi(x^0))$ is bounded, by Lemma 4, we conclude that $\{x^k\}_{k\geq 0}$ is bounded. Thus Ω^* is nonempty. Next, by passing to the limit into the subsequence and then combining the result and Lemma 2, we can easily prove that every limit point is a stationary point of (SepNCOP). If Ω^* is finite, by applying the result in [12, Chapt. 28], we obtain the proof of the remaining statement in Theorem 1. ■

IV. Two-level decomposition algorithm

Since the primal subproblem (17) in Algorithm 1 is strongly convex and separable, one can apply several dual decomposition algorithms developed recently in [7], [16] to solve this problem.

Problem (17) can be written explicitly as follows:

$$\min_x \sum_{i=1}^{M} g_i(x_i) + h_i(F_i(x_i^k) + F_i'(x_i^k)(x_i - x_i^k)) + \frac{\beta}{2}\|x_i - x_i^k\|^2,$$

$$\text{s.t.} \sum_{i=1}^{M}(A_i x_i - b_i) = 0,\ x_i \in X_i,\ \forall i = 1,\dots,M. \quad (26)$$

In this section we combine Algorithm 1 and a dual decomposition algorithm from [7, Algorithm 1] or [16, Algorithm 3] to obtain a two-level algorithm for solving (SepNCOP). For simplicity of exposition, we also assume that the Lipschitz constants L_h and $L_{F'}$ are known *a priori*. The two-level algorithm is described in detail as follows.

Algorithm 2: *(Two-level SCP decomposition)*

Initialization: Perform the following steps:

1. Given a tolerance $\varepsilon_{\text{outer}} > 0$ for the outer loop.
2. Choose $2M_\psi > \underline{\beta} \geq M_\psi = L_h L_{F'}$ and $\mu \in (0, 1)$.
3. For each component i, choose an initial point $x_i^0 \in X_i$.
4. Select an accuracy level $0 < \varepsilon_0 < \varepsilon_{\text{outer}}$.

Outer iteration: For $k = 0, 1, \dots,$ perform the 4 steps:

Step 1: Select an appropriate value $\beta_k \in [\underline{\beta},\ 2M_\psi]$.

*Step 2: (**inner iteration**).* For a given $x^k = [(x_1^k)^T, \dots, (x_M^k)^T]^T$, apply [7, Alg. 1] or [16, Alg. 3] to solve (26) up to the accuracy ε_k to obtain the solution x^{k+1}.

Step 3: If $\|x_i^{k+1} - x_i^k\| \leq \varepsilon_{\text{outer}}$, $i = 1,\dots,M$, then terminate the outer loop k.

Step 4: For $i = 1,\dots,M$, evaluate F_i and its Jacobian F_i' at x_i^{k+1} *in parallel*. Update $\varepsilon_{k+1} = \min\left\{\frac{\mu M_\psi}{2}\|x^{k+1} - x^k\|^2, \varepsilon_k\right\}$.

End.

Since each objective component of (26) is *strongly convex*, the **inner-loop** (Step 2) carried out by [7, Algorithm 1] or [16, Algorithm 3] converges sublinearly at the rate $\mathcal{O}(1/j^2)$, where j is the inner-iteration counter. Therefore, we can terminate the inner-loop after a certain number of iterations which can be defined *a priori*. In [7], [16] the authors proved the convergence rate for the dual decomposition algorithm for solving (26).

The following theorem shows the convergence of Algorithm 2 as a simple consequence of Theorem 1. We omit the proof details here.

Theorem 2: Under the assumptions of Theorem 1, the sequence $\{x^k\}_{k\geq 0}$ generated by Algorithm 2 still satisfies the conclusions of Theorem 1.

If g is differentiable and its gradient is $L_{g'}$-Lipschitz continuous in $\mathcal{D}$, then we can modify Algorithm 2 to process this case. The computation of the accuracy ε_k in Algorithm 2 still requires global information, the norm of the difference of the vectors x^k and x^{k+1}. In a distributed implementation, we can estimate an upper bound of this quantity and distribute it into M subsystems.

V. Numerical tests

We consider a simple *distributed model predictive control* problem associated with a *bilinear* dynamic system and quadratic costs. Under a change of variables and given assumptions, we can reformulate this problem as a nonconvex optimization problem of the form:

$$\begin{cases} \min_{x\in\mathbb{R}^n} & \sum_{i=1}^{n} \frac{1}{2}p_i x_i^2 + q_i x_i \\ \text{s.t.} & \frac{1}{2}c_i x_i^2 + d_i x_i + e_i = 0,\ 0 \leq x_i \leq 1,\ i = 1,\dots,n, \\ & \sum_{i=1}^{n}(A_i x_i - b_i) = 0. \end{cases} \quad (27)$$

Here, p_i, q_i, c_i, d_i, e_i $(i = 1,\dots,n)$, are the components of vectors p, q, c, d, e in $\mathbb{R}^n$, respectively. Further, $A = [A_1, \dots, A_n] \in \mathbb{R}^{p\times n}$ and $b_i \in \mathbb{R}^p$ for $i = 1,\dots,n$. Let $h(u) = \rho\|u\|_1$, which is convex and ρ-Lipschitz continuous. The Jacobian $\nabla F(x) = \text{diag}(c_1x_1 + d_1, \cdots, c_nx_n + d_n)$ is also $L_{F'}$-Lipschitz continuous, where $L_{F'} = \max_{1\leq i\leq n}|c_i|$. Hence, problem (27) is indeed of the form (SepNCOP). We implemented Algorithm 2 in C++ running on a 16 cores Intel ®Xeon 2.7GHz workstation with 12 GB of RAM. The **inner-loop** at Step 2 of Algorithm 2 was parallelized using `OpenMP`. We terminated the **outer-loop** of the algorithm if the relative feasibility gap $\texttt{rfgap} := \|Ax^k - b\|/\max\{\|Ax^0 - b\|, 1.0\} \leq 10^{-3}$ and either $\texttt{error} := \|x^{k+1} - x^k\|/\max\{\|x^k\|, 1.0\} \leq 10^{-3}$ or the quantity $\text{rfval}_{kj} := \left|\phi(x^k) - \phi(x^{k-j})\right|/\max\{\left|\phi(x^k)\right|, 1.0\}$ does not change significantly after five successive iterations. The initial tolerance ε_0 for the **inner-loop** was set to $\varepsilon_0 = 0.5 \times 10^{-3}$, and then was updated by $\varepsilon_{k+1} = \left\{\frac{0.5M_\psi}{2}\|x^{k+1} - x^k\|^2, \varepsilon_k\right\}$. The maximum number of iterations in the **inner-loop** was set to $j_{\max} = 10,000$. The parameter β_k was fixed at $\beta_k = 1.005M_\psi$. We notice that the primal subproblems formed from each component of (26) in this example can be solved in a *closed form* due to its univariate form.

The data of the problem instances was generated as follows. Vectors d, e, q and matrix A were generated randomly in $[-1,\ 1]$. Vectors c and p were also generated randomly in

$[-0.5,\ 0.5]$ and $[0,\ 1]$, respectively. Vector b was computed by $b = Ax_t$ for a given test point $x_t \in [0,\ 1]^n$. The penalty parameter ρ was fixed at $\rho = 10$, which is appropriate in this example.

We have tested Algorithm 2 for 25 problem instances. The performance information and results reported by this algorithm are shown in Table I. Here, the first column is the

TABLE I

PERFORMANCE INFORMATION AND RESULTS OF ALGORITHM 2

P_{n^o}	Size		Performance			Results		
	m	n	`oit`	`iit`	`time`	`error`	`rfgap`	`objval`
#1	50	1000	11	1868	5.31	9.012×10^{-3}	0.475×10^{-3}	3539.450
#2	100	1000	11	2149	9.15	6.775×10^{-3}	0.586×10^{-3}	3469.270
#3	150	1000	11	2749	16.91	4.566×10^{-3}	0.558×10^{-3}	3685.160
#4	200	1000	11	3143	23.82	7.307×10^{-3}	0.606×10^{-3}	3929.740
#5	250	1000	11	3464	30.48	8.298×10^{-3}	0.619×10^{-3}	3940.790
#6	300	1000	11	3852	40.08	5.861×10^{-3}	0.587×10^{-3}	3973.160
#7	350	1000	11	4004	47.76	5.299×10^{-3}	0.723×10^{-3}	4259.940
#8	400	1000	11	4114	54.33	5.174×10^{-3}	0.731×10^{-3}	4486.220
#9	450	1000	11	4711	67.86	5.831×10^{-3}	0.740×10^{-3}	4730.130
#10	500	1000	11	4885	78.33	8.147×10^{-3}	0.691×10^{-3}	4848.320
#11	550	2000	11	4804	226.33	8.743×10^{-3}	0.672×10^{-3}	8324.100
#12	600	2000	11	5159	316.39	5.292×10^{-3}	0.623×10^{-3}	8087.260
#13	650	2000	11	5255	366.47	7.940×10^{-3}	0.667×10^{-3}	8373.030
#14	700	2000	11	5645	402.23	7.805×10^{-3}	0.707×10^{-3}	8976.730
#15	750	2000	11	5898	493.21	5.860×10^{-3}	0.668×10^{-3}	8530.690
#16	800	3000	11	6184	974.17	8.942×10^{-3}	0.648×10^{-3}	12270.300
#17	850	3000	11	6103	788.78	6.926×10^{-3}	0.674×10^{-3}	12361.500
#18	900	3000	11	6214	900.54	6.939×10^{-3}	0.661×10^{-3}	12305.300
#19	950	3000	11	6444	1029.44	7.514×10^{-3}	0.662×10^{-3}	12469.400
#20	1000	3000	11	6666	1076.14	7.629×10^{-3}	0.714×10^{-3}	13007.200
#21	1000	3500	11	6404	1364.64	6.842×10^{-3}	0.703×10^{-3}	14458.000
#22	1000	4000	11	6721	1482.57	10.381×10^{-3}	0.648×10^{-3}	16178.600
#23	1000	4250	11	6847	1317.29	7.160×10^{-3}	0.615×10^{-3}	16349.500
#24	1000	4500	11	6668	1552.98	7.750×10^{-3}	0.635×10^{-3}	17722.600
#25	1000	5000	11	6777	1973.95	7.895×10^{-3}	0.611×10^{-3}	19873.700

problem dimension; `oit` is the number of outer iterations; `iit` is the number of average inner iterations in Algorithm 2; `time` is the total of computational time in seconds; two quantities `rfgap` and `error` are defined as above; and `objval` is the objective values.

As we can observe from Table I the number of outer iterations in Algorithm 2 is rather small and does not change when the size of the problem increases. However, since the algorithm of the **inner-loop** is just a first-order method, it requires many iterations and the number of iterations increases significantly when the problem size increases. Note that we can reduce the number of inner iterations by reducing the accuracy ε_0. In contrast, the number of outer iterations increases accordingly.

VI. CONCLUSION

In this paper, we have presented a two-level decomposition algorithm for solving a class of separable *nonconvex* optimization problems. This algorithm can be considered as a combination of an *inexact SCP scheme* and a decomposition algorithm for separable and strongly convex optimization problems. We have proved the global convergence of the SCP outer loop and, consequently, we have obtained the convergence of the whole algorithm. This algorithm has been tested via some preliminary numerical examples.

ACKNOWLEDGMENTS

The research leading to these results has received funding from: Research Council KUL: PFV/10/002 Optimization in Engineering Center OPTEC; the European Union Seventh Framework Programme (FP7-EMBOCON/2007–2013) under grant agreement no 248940; CNCS-UEFISCDI (project TE-231, no. 19/11.08.2010); POSDRU/89/1.5/S/62557.

REFERENCES

[1] D. Bertsekas and J. N. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Prentice Hall, 1989.

[2] D. Bertsekas, "Convexification Procedures and Decomposition Methods for Nonconvex Optimization Problems," *J. Optim. Theory and Appl.*, vol. 29, no. 2, pp. 169–197, 1979.

[3] P. Christofides, R. Scattolini, D. M. de la Pena, and J. Liu, "Distributed model predictive control: A tutorial review and future research directions," *Computers and Chemical Engineering,*, pp. 1–12, 2012.

[4] A. Hamdi, "Two-level primal-dual proximal decomposition technique to solve large-scale optimization problems," *Appl. Math. Comput.*, vol. 160, pp. 921–938, 2005.

[5] S. Lin, "Complete Decomposition Algorithm for Nonconvex Separable Optimization Problems and Applications," *Automatica*, vol. 28, no. 6, pp. 1249–1254, 1992.

[6] 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.

[7] I. Necoara and J. Suykens, "Applications of a smoothing technique to decomposition in convex optimization," *IEEE Trans. Automatic control*, vol. 53, no. 11, pp. 2674–2679, 2008.

[8] I. Necoara, V. Nedelcu, "Rate analysis of inexact dual first order methods with application to dual decomposition", *IEEE Transactions on Automatic Control*, http://arxiv.org/abs/1302.3129, 2013.

[9] I. Necoara, C. Savorgnan, Q. Tran-Dinh, J. Suykens, M. Diehl, "Distributed Nonlinear Optimal Control using Sequential Convex Programming and Smoothing Techniques", Proceedings of Conference on Decision and Control, pp. 543-548, 2009.

[10] J. Nocedal and S. Wright, *Numerical Optimization*, 2nd ed., ser. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[11] Y. Nesterov, "Dual extrapolation and its applications to solving variational inequalities and related problems," *Math. Program.*, vol. 109, no. 2–3, pp. 319–344, 2007.

[12] A. Ostrowski, *Solutions of Equations and Systems of Equations*. New York: Academic Press, 1966.

[13] R. Scattolini, "Architectures fior distributed and hierarchical model predictive control," *Journal of Process Control*, vol. 19, pp. 723–731, 2009.

[14] W. Sun and Y.-X. Yuan, *Optimization Theory and Methods*. Springer, 2010.

[15] G. Stephanopoulos and W. A.W., "The use of Hestenes' method of multipliers to resolve dual gaps in engineering system optimization," *J. Optim. Theory and Appl.*, vol. 15, pp. 285–309, 1975.

[16] Q. Tran-Dinh, C. Savorgnan, and M. Diehl, "Combining lagrangian decomposition and excessive gap smoothing technique for solving large-scale separable convex optimization problems," *Compt. Optim. Appl.*, vol. 55, no. 1, pp. 75–111, 2013.

[17] Q. Tran-Dinh, I. Necoara, C. Savorgnan, and M. Diehl, "An Inexact Perturbed Path-Following Method for Lagrangian Decomposition in Large-Scale Separable Convex Optimization," *SIAM J. Optim.*, vol. 23, no. 1, pp. 95–125, 2013.

[18] A. Tanikawa and H. Mukai, "A new technique for nonconvex primal-dual decomposition of a large-scale separable optimization problem," *IEEE Trans. Automatic Control*, vol. 30, no. 2, pp. 133–143, 1985.

[19] P. Tatjewski and B. Engelmann, "Two-Level Primal-Dual Decomposition Technique for Large-Scale Nonconvex Optimization Problems with Constraints," *J. Optim. Theory and Appl.*, vol. 64, no. 1, pp. 183–205, 1990.

[20] A. Venkat, I. Hiskens, J. Rawlings, and S. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1192–126, 2008.