Mathematical Programming manuscript No.
(will be inserted by the editor)

Alexandre Belloni · Claudia Sagastizábal

Dynamic Bundle Methods

Third Revision:March 15, 2007

Abstract. Lagrangian relaxation is a popular technique to solve difficult optimization problems. However, the applicability of this technique depends on having a relatively low number of hard constraints to dualize. When there are many hard constraints, it may be preferable to relax them dynamically, according to some rule depending on which multipliers are active. From the dual point of view, this approach yields multipliers with varying dimensions and a dual objective function that changes along iterations.

We discuss how to apply a bundle methodology to solve this kind of dual problems. Our framework covers many separation procedures to generate inequalities that can be found in the literature, including (but not limited to) the most violated inequality. We analyze the resulting dynamic bundle method giving a positive answer for its primal-dual convergence properties, and, under suitable conditions, show finite termination for polyhedral problems.

1. Introduction

Consider the following optimization problem:

$$\begin{cases} \max\limits_{p} C(p) \\ p \in \mathcal{Q} \subset \mathbb{R}^{\mathtt{p}} \\ g_j(p) \leq 0\,, j \in L := \{1, \ldots, n\}, \end{cases} \tag{1}$$

where $C : \mathbb{R}^{\mathtt{p}} \to \mathbb{R}$ is a concave function and, for each $j \in L$, $g_j : \mathbb{R}^{\mathtt{p}} \to \mathbb{R}$ denotes an affine constraint, usually *hard* to deal with. Easy constraints are included in the (possibly discrete) set $\mathcal{Q}$, whose convex hull, *conv* $\mathcal{Q}$, is assumed to be a compact set. Our interest lies in applications where n, the number of hard constraints, is an huge number, possibly exponential in the primal dimension $\mathtt{p}$. This is a common setting in Combinatorial Optimization, as shown by many works in the area devoted to the study of different families of inequalities aimed

Alexandre Belloni: IBM T. J. Watson Research Center and MIT, 32-221, 1101 Kitchawan Road, Yorktown Heights, New York 10598, e-mail: `belloni@mit.edu`

Claudia Sagastizábal: CEPEL, Electric Energy Research Center. On leave from INRIA Rocquencourt, France. Postal address: IMPA, Estrada Dona Castorina 110, Jardim Botânico, Rio de Janeiro RJ 22460-320, Brazil. Research supported by CNPq Grant No. 303540-03/6, e-mail: `sagastiz@impa.br`

Correspondence to: Claudia Sagastizábal

at strengthening linear relaxations of several prototypical problems. Similar features are present in Stochastic Programming, specifically when dealing with the solution of multi-stage programs with recourse, [RS03].

Assuming that problem (1) has enough structure, for example when it is convex and a constraint qualification condition holds, solutions to (1) can be found by using Lagrangian Relaxation techniques. In the absence of enough structure, the approach yields solutions to a convexified form of (1), with $\mathcal{Q}$ therein replaced by its convex hull $conv\,\mathcal{Q}$:

$$\begin{cases} \max\limits_{p} C(p) \\ p \in conv\,\mathcal{Q} \\ g_j(p) \leq 0\,, j \in L. \end{cases} \qquad conv\,(1)$$

Letting $x \in \mathbb{R}^n$ denote the nonnegative multipliers associated to the hard constraints, the Lagrangian dual problem of (1) is given by

$$\min_{x \geq 0} f(x)\,, \qquad \text{where} \qquad f(x) := \max_{p \in \mathcal{Q}} \left\{ C(p) - \sum_{j \in L} g_j(p) x_j \right\} \tag{2}$$

is the dual function, possibly nondifferentiable.

For the Lagrangian Relaxation approach to make sense in practice, two points are fundamental. First, the dual function should be much simpler to evaluate (at any given x) than solving the primal problem directly; we assume this is the case for (1). Second, the dual problem should not be too big: in nonsmooth optimization (NSO) this means that the dimension of the dual variable is (roughly) less than one hundred thousand. Therefore, the approach is simply not applicable in (1) when n is too large.

Instead of dualizing all the n hard constraints at once, an alternative approach is to choose at each iteration *subsets* of constraints to be dualized. In this dynamical relaxation, subsets J have cardinality $|J|$ much smaller than n. As a result, the corresponding dual function, defined on $\mathbb{R}^{|J|}$, is manageable from the NSO point of view.

The idea of replacing a problem with difficult feasible set by a sequence of subproblems with simpler constraints can be traced back to the cutting-planes methods [CG59, Kel60]. An important matter for preventing subproblems from becoming too difficult is how and when to drop constraints (i.e., how to choose the subsets J). In Combinatorial Optimization, a method to dynamically insert and remove inequalities was already used for the Traveling Salesman Problem (TSP) in [BC81] and for network design in [Gav85]. Related works are [CB83, Luc92, Bea90, LB96], and more recently, [HFK01, BL02]. Being an special class of cutting-planes method, the approach was named *Relax-and-Cut* algorithm in [EGM94].

All the works mentioned above use a subgradient type method ([Erm66, Sho85, HWC74]) to update the dual multipliers. More recently, more robust NSO methods have been used for the dual step, such as the proximal analytical center cutting planes method in [BdMV05] and bundle methods in [FG99, FGRS06, RS06, FLR06, Hel01]. Although numerical experience shows the efficiency of a dynamic technique, except for the last work no convergence proof for the algorithm is given. The algorithm in [Hel01] is tailored to solve semidefinite (SDP) relaxations of combinatorial problems, with a dual step that uses the so-called spectral bundle method. To choose subsets J, a *maximum violation oracle* seeks for the most violated inequality in the primal problem at each iteration.

In this paper we consider these methods from a broader point of view, focusing on a general dual perspective. Our procedure applies to general problems, not only combinatorial problems. Primal information produced by a *Separation Oracle* (which identifies constraints in (1) that are not satisfied, i.e., "violated inequalities") is used at each iteration to choose the subset J. Dual variables, or multipliers, are updated using a particular form of bundle methods, that we call *Dynamic Bundle Method* and is specially adapted to the setting.

When compared to [Hel01], our method is more general, because our separation procedure depends on the current index set J; see Section 4 below. This is an important feature for applications where the family of inequalities defined by L does not have an efficient maximum violation oracle and only heuristics can be used to search for violated inequalities. Naturally, the quality of the separation procedure does limit the quality of the solution obtained by solving dynamically the dual problem. For this reason, to prove global convergence we force the separation procedure to eventually satisfy a *complementarity rule* (cf. Section 5).

The well-known descent and stability properties of the bundle methodology, together with a sound management of dualized constraints, yield convergence of the dynamic strategy in a rather general setting. More precisely, we show that the method asymptotically solves the dual and the convexified primal problem (respectively (2) and $conv(1)$), with finite termination when $\mathcal{Q}$ in (1) is finite (cf. Theorems 6, 9, and 11 below).

This paper is organized as follows. Sections 2 and 3 discuss the general dynamic formulation and the bundle methodology adapted to the new context. The important problem of how to select inequalities with a Separation Oracle is addressed in section 4. In section 5 we describe the dynamic bundle method. Sections 6 and 7 contain our convergence results. Finally, in Section 8 we give some concluding remarks.

Our notation and terminology is standard in bundle methods; see [HUL93] and [BGLS06]. We use the euclidean inner product in $\mathbb{R}^n$: $\langle x, y\rangle = \sum_{j=1}^n x_j y_j$, with induced norm denoted by $|\cdot|$. For an index set $J \subset L = \{1, \ldots, n\}$, $|J|$ stands for its cardinality. We write $\mathbb{R}^n = \mathbb{R}^{|J|} \times \mathbb{R}^{n-|J|}$ and denote the respective components as follows:

$$\mathbb{R}^n \ni x = \big(x_J, x_{L\setminus J}\big) := \big(x_{j\in J}, x_{j\in L\setminus J}\big).$$

Finally, given $\tilde{x}_J$ nonnegative, we often consider the cone

$$N_J(\tilde{x}) = \{\nu \in \mathbb{R}^n : \langle \nu, x - \tilde{x}\rangle \leq 0 \text{ for all } x_J \geq 0\},$$

which is normal to the set $\{x \geq 0 : x_{L\setminus J} = 0\}$ at the point $\tilde{x} = \big(\tilde{x}_J, \tilde{x}_{L\setminus J}\big)$.

2. Combining primal and dual information

The dual function f from (2) is the maximum of a collection of affine functions of x, so it is convex. Under mild regularity conditions on (1), the dual problem (2) has a solution (cf. (19) below). In addition, the well-known weak duality property

$$f(x) \geq C(p) \quad \text{for all } x \geq 0 \text{ and } p \text{ primal feasible,} \tag{3}$$

implies in this case that f is bounded from below. Moreover, note that the evaluation of the dual function at a given value of x gives straightforwardly a subgradient. More precisely, letting $p_x \in \mathcal{Q}$ be a maximizer in the right hand side expression in (2), i.e., p_x satisfying $f(x) = C(p_x) - \sum_{j\in L} g_j(p_x)x_j$, it holds that

$$-g(p_x) = -\left(g_j(p_x)_{j\in L}\right) = -(g_j(p_x),_{\{j=1,\ldots,n\}}) \in \partial f(x)\,.$$

Based on this somewhat minimal information, *black-box* NSO methods generate a sequence $\{x^k\}$ converging to a solution $\bar{x}$ of the dual problem (2); see [BGLS06, Chapters 9-10]. As mentioned, if there is no duality gap, the corresponding $p_{\bar{x}}$ solves *conv*(1). Otherwise, it is necessary to recover a primal solution with some heuristic technique. Such techniques usually make use of $p_{\bar{x}}$ or even of the primal iterates p_{x^k}; see for example [BA00], [BMS02].

An important consequence of considering a subset J instead of the full L is that complete knowledge of the dual function is no longer available. Namely, for any given x, only

$$\max_{p\in\mathcal{Q}} \left\{ C(p) - \sum_{j\in\mathbf{J}} g_j(p)x_j \right\} \quad \textbf{and not} \quad \max_{p\in\mathcal{Q}} \left\{ C(p) - \sum_{j\in\mathbf{L}} g_j(p)x_j \right\}$$

is known. For this reason, primal and dual information should be combined adequately. In the next section we analyze how the introduction of a dynamic setting enters into a bundle algorithm. The analysis sheds a light on how to select constraints to be dualized at each iteration.

3. Bundle methods

To help getting a better understanding of how the dynamic bundle method works, we emphasize the main modifications that are introduced by the dynamic dualization scheme when compared to a standard bundle algorithm.

Since (2) is a constrained NSO problem, there is an additional complication in the algebra commonly used in unconstrained bundle methods. We assume that the primal problem (1) is such that the dual function f is finite everywhere. In this case, bundle methods are essentially the same than for unconstrained NSO, because the dual problem in (2) is equivalent to the unconstrained minimization of f appended with an indicator function.

3.1. An overview of bundle methods

Let ℓ denote the current iteration of a bundle algorithm. Classical bundle methods keep memory of the past in a *bundle* of information $\mathcal{B}_\ell$ consisting of

$$\Big\{f(x^i)\,,s^i \in \partial f(x^i)\Big\}_{i\in\mathcal{B}_\ell}$$

and containing the current "serious" iterate, $\hat{x}^{k(\ell)}$ (although not always explicitly denoted, the superscript k in $\hat{x}^k$ depends on the iteration ℓ, i.e., $k = k(\ell)$). Serious iterates, also called stability centers, form a subsequence $\{\hat{x}^k\} \subset \{x^i\}$ such that $\{f(\hat{x}^k)\}$ is decreasing.

When not all the hard constraints are dualized, there is no longer a fixed dual function f, but dual objectives depending on some index set J, with J varying along iterations. For example, $\hat{J}_k$ below denotes the index set used when generating the stability center $\hat{x}^k$ and, similarly, J_i corresponds to an iterate x^i. In addition, keeping in mind that f is a Lagrangian function, we let p^i be a maximizer defining $f(x^i)$:

$$p^i \in \operatorname{Arg}\max_{p\in\mathcal{Q}} \left\{ C(p) - \sum_{j\in J_i} g_j(p)x^i_j \right\}, \tag{4}$$

so that $-g(p^i) = -\big(g_j(p^i)_{j\in L}\big)$ is a subgradient of f at x^i. We will see that, by construction, it always holds that $x^i_{L\setminus J_i} = 0$. Since $f(x^i) = C(p^i) - \big\langle g(p^i), x^i\big\rangle$, for convenience, in Algorithm 1 below

$$p^i = DualEval(x^i)$$

denotes these computations. With this notation, the bundle data is

$$\Big\{\Big(C_i := C(p^i)\,,p^i\Big)\Big\}_{i\in\mathcal{B}_\ell}.$$

Although at this stage keeping the values C_i may seem superfluous, we will see in Remark 4 that such information becomes important when there is bundle compression: in this case, C_i no longer corresponds to $C(p^i)$, but rather to a convex combination of functional values.

Bundle information gives at each iteration a *model* of the dual function f, namely the cutting-planes function

$$\check{f}_\ell(x) = \max_{i\in\mathcal{B}_\ell}\big\{C_i - \big\langle g(p^i), x\big\rangle\big\}\ , \tag{5}$$

where the bundle $\mathcal{B}_\ell$ varies along iterations. The model is used to approximate f in the (simple) optimization subproblem defining the iterate x^ℓ; see for example (8) below.

An iterate x^ℓ is considered good enough to become the new stability center when $f(x^\ell)$ provides a significant decrease, measured in terms of the nominal decrease Δ_ℓ, which is defined as

$$\Delta_\ell := f(\hat{x}^k) - \check{f}_\ell(x^\ell), \tag{6}$$

and is a nonnegative quantity by construction. When x^ℓ is declared a serious iterate, it becomes the next stability center, i.e., $\hat{x}^{k+1} := x^\ell$; otherwise, the stability center is not updated, and a null step is declared. In all cases, dual information of the iterate is used to add the pair $(C(p^\ell), p^\ell)$ to the next bundle, i.e., to $\mathcal{B}_{\ell+1}$.

3.2. Defining iterates in a dynamic setting

We now consider in more detail the effect of working with a dynamic dualization scheme.

Let J_ℓ be an index set such that $\hat{J}_k \subseteq J_\ell$ for $k = k(\ell)$. To define a nonnegative iterate satisfying $x^\ell_{L\setminus J_\ell} = 0$, we solve a quadratic programming problem (QP) depending on the varying set J_ℓ and on a positive parameter μ_ℓ. More specifically, x^ℓ solves:

$$\begin{cases} \min\limits_{x\geq 0} \check{f}_\ell(x) + \dfrac{1}{2}\mu_\ell|x - \hat{x}^k|^2 \\ x_{L\setminus J_\ell} = 0\,. \end{cases}$$

Here arises a major difference with a purely static bundle method. Instead of just solving a QP in the whole space $\mathbb{R}^n$:

$$\min_{x\geq 0} \check{f}_\ell(x) + \frac{1}{2}\mu_\ell|x - \hat{x}^k|^2, \tag{7}$$

we introduce an additional constraint zeroing components out of J_ℓ, and solve a QP of reduced dimension, i.e., on $\mathbb{R}^{|J_\ell|}$ (cf. (8) below).

We now derive some technical relations that are useful for the convergence analysis. We use the notation $S_\ell := \left\{z \in \mathbb{R}^{|\mathcal{B}_\ell|} : z_i \geq 0\,, \sum_{i\in\mathcal{B}_\ell} z_i = 1\right\}$ to denote the unit simplex associated to the bundle $\mathcal{B}_\ell$.

Lemma 1. *Let $\hat{x}^k \geq 0$ be such that $\hat{x}^k_{L\setminus\hat{J}_k} = 0$. Having the affine function $g : \mathbb{R}^{\mathsf{p}} \to \mathbb{R}^n$, let $\left\{(C_i, p^i) \in \mathbb{R}\times \in \mathbb{R}^{\mathsf{p}} : i \in \mathcal{B}_\ell\right\}$ define the cutting-planes function in* (5)*. Given an index set J_ℓ such that $\hat{J}_k \subset J_\ell \subset L$ and $\mu_\ell > 0$, consider the QP problem*

$$\min_{x_{J_\ell}\geq 0}\left\{\check{f}_\ell\Big((x_{J_\ell}, 0_{L\setminus J_\ell})\Big) + \frac{1}{2}\mu_\ell|x_{J_\ell} - \hat{x}^k_{J_\ell}|^2\right\}, \tag{8}$$

with unique solution $x^\ell_{J_\ell}$ and let $x^\ell = (x^\ell_{J_\ell}, 0_{L\setminus J_\ell})$. Then

$$\exists\Big(\alpha^\ell, \hat{s}^\ell\Big) \in S_\ell \times \partial\check{f}_\ell(x^\ell) \text{ with } \hat{s}^\ell = -g(\hat{\pi}^\ell) \text{ and } \hat{\pi}^\ell := \sum_{i\in\mathcal{B}_\ell} \alpha_i^\ell p^i \tag{9a}$$

$$\exists\hat{\nu}^\ell \in N_{J_\ell}(x^\ell) \text{ with } N_{J_\ell}(x^\ell) = \left\{\nu \in \mathbb{R}^n : \nu_{J_\ell} \leq 0\,, \left\langle\hat{\nu}^\ell, x^\ell\right\rangle = 0\right\} \tag{9b}$$

such that $x^\ell = \hat{x}^k - \dfrac{1}{\mu_\ell}(\hat{s} + \hat{\nu})$.

Proof. Apply Lemma 3.1.1 in [HUL93, Ch. XV, vol. II], written with $\Big(C, x, y_+, \hat{p}\Big)$ therein replaced by $\Big(\{x \geq 0 : x_{L\setminus J_\ell} = 0\}, \hat{x}^k, x^\ell, \hat{\nu}^\ell\Big)$, respectively. To obtain the expression for $\hat{s}^\ell$, use the fact that the cutting-planes model (5) is a piecewise affine function with subdifferential

$$\partial \check{f}_\ell(x) = \left\{ -\sum_{i \in \mathcal{B}_\ell} \alpha_i g(p^i) \text{ for } \alpha \in S_\ell \text{ and } i \in \mathcal{B}_\ell : \check{f}_\ell(x) = C_i - \left\langle g(p^i), x \right\rangle \right\},$$

and the fact that g is affine. The expression for the normal cone in (9b) can be found in [HUL93, Ex. III.5.2.6(b), vol. I]. □

In the algorithm stated in section 5 below, for the QP information we use the condensed notation

$$\left(x^\ell, \hat{s}^\ell, \hat{\nu}^\ell, \alpha^\ell\right) = QPIt(\hat{x}^k, \check{f}_\ell, J_\ell, \mu_\ell)\,.$$

We finish this section with additional technical relations and an important corollary of Lemma 1.

Lemma 2. *The following relations are a consequence of* (5) *and* (9)*:*

$$\check{f}_\ell(x^\ell) \leq C(\hat{\pi}^\ell) - \left\langle g(\hat{\pi}^\ell), x^\ell \right\rangle, \tag{10a}$$

$$g_j(\hat{\pi}^\ell) \leq -(\hat{s}^\ell_j + \hat{\nu}^\ell_j) \text{ for all } j \in J_\ell, \text{ and} \tag{10b}$$

$$\left\langle g(\hat{\pi}^\ell), x^\ell \right\rangle = -\left\langle \hat{s}^\ell + \hat{\nu}^\ell, x^\ell \right\rangle. \tag{10c}$$

Proof. Relation (10a) follows from (5), the definition of $\hat{\pi}^\ell$, and the concavity of C, recalling that $\check{f}_\ell(x^\ell) = \sum_{i \in \mathcal{B}_\ell} \alpha^\ell_i \Big(C_i - \left\langle g(p^i), x^\ell \right\rangle\Big)$ with $C_i = C(p^i)$. Relations (10b) and (10c) follow, respectively, from the inequality $-\hat{\nu}^\ell_j \geq 0$ for all $j \in J_\ell$, and the expression $\hat{s}^\ell = g(\hat{\pi}^\ell)$ in (9a); and the identity $\left\langle \hat{\nu}^\ell, x^\ell \right\rangle = 0$ from (9b). □

Corollary 3. *Consider the notation and assumptions of Lemma 1. If* $g_{L\setminus J_\ell}(\hat{\pi}^\ell) \leq 0$ *then the point* x^ℓ *solves the static quadratic program* (7)*.*

Proof. For x^ℓ to solve (7), the optimality condition (9) should hold for $\hat{\nu}^\ell \in N_L(x^\ell)$, i.e., (9b) should hold with $J_\ell = L$. Since $\hat{s}^\ell + \hat{\nu}^\ell = \mu_\ell(\hat{x}^k - x^\ell)$ has zero components in $L\setminus J_\ell$, from (9a) we see that $\hat{\nu}_{L\setminus J_\ell} = g_{L\setminus J_\ell}(\hat{\pi}^\ell)$ and the result follows. □

4. Selecting inequalities

Corollary 3 establishes that the condition

$$g_{L\setminus J_\ell}(\hat{\pi}^\ell) \leq 0$$

is sufficient for preserving the static bundle mechanism in the dynamic setting. Hence, if constraints are selected in a way that ensures eventual satisfaction of such condition, the corresponding dynamic method has chances of yielding a convergent algorithm.

To choose which constraints are to be dualized at each iteration we assume that a *Separation Oracle* is available. A Separation Oracle is a procedure *SepOr* that given $p \in \mathbb{R}^{\mathrm{p}}$ identifies inequalities violated by p, i.e., some index j such that $g_j(p) > 0$ for j in a subset of L. The output of the procedure is a set of indices which can be empty only if no inequality is violated. In Combinatorial Optimization, this corresponds to having an *exact separation* algorithm for the family of inequalities defined by the hard constraints.

Keeping in mind Corollary 3, only *smart* Separation Oracles will ensure convergence of the dynamic method. By "smartness" one may ask from the Separation Oracle to satisfy the following assumption:

$$\text{having the working set } J,\ SepOr(p) \subseteq J \Longrightarrow \text{ for all } l \in L\setminus J\ g_l(p) \leq 0. \quad (11)$$

In other words, as long as there remain inequalities in $L\setminus J$ violated by p, the separation procedure is able to identify one of such constraints.

The "maximum violation oracle" (MV_L) is a standard example of separation oracle found in the literature (for example in [Hel01]). MV_L defines $SepOr(p)$ as a subset of

$$\left\{ j \in L : g_j(p) = \max_{l \in L}\{g_l(p) > 0\} \right\}.$$

Since MV_L does not satisfy (11) and we aim at establishing convergence results that also cover the case of the maximum violation oracle, we relax (11) in the following manner:

$$\begin{array}{c} \text{given a fixed number } \beta > 0 \text{ and the working set } J, \\ SepOr(p) \subseteq J \Longrightarrow \text{ for all } l \in L\setminus J \\ \begin{cases} \text{either } g_l(p) \leq 0 \\ \text{or} \quad g_l(p) > 0 \text{ but there is } j \in J : g_l(p) \leq \beta g_j(p). \end{cases} \end{array} \quad (12)$$

Note that this weaker condition is satisfied by MV_L with $\beta = 1$.

Depending on the application, separating procedures different from MV_L may be preferable. The construction of an efficient separation procedure is an important question by itself and has been extensively studied in Combinatorial Optimization. In many cases, the particular structure of $\mathcal{Q}$ and g can be exploited so that the Separation Oracle reduces to solving a simple optimization problem. Nevertheless, there are families of inequalities for which no efficient exact separation procedure is known so far (for example, the path inequalities for the traveling salesman problem; see [BS06]). In order to gain in generality, in this work we make minimal assumptions on the separation procedure. Note in particular that assumption (12) allows for *any violated inequality* to be separated, requiring no special property from that inequality (such as being the maximum violated inequality). Furthermore, when the Separation Oracle returns many inequalities, (12) allows just to choose any nonempty *subset* of such inequalities. This is a feature of practical interest when using heuristic approaches, either because no exact separation procedure is available, or because there are inequalities that appear as "more promising" for some reason, intrinsic to the nature of the problem. For instance, any efficient approximation algorithm for the maximum violation oracle problem could be used to define a separation oracle that satisfies (12). Finally, even if an exact separation oracle is available, it may still be interesting to start the process with some fast heuristic method, to increase the overall efficiency.

We finish this section with some intuitive facts explaining why condition (12) should suffice to ensure convergence of a dynamic method (for simplicity, we dropped the iteration index ℓ):

A dynamic method with a reasonable management of relaxed constraints should eventually end up working with a fixed index set, say J. ¿From that point on, the dynamic dual phase behaves like a static method, working on the reduced set J. In particular, it drives to zero the J-component of $\hat{s} + \hat{\nu}$. Because f is a Lagrangian function and the relaxed constraints are affine, by (10b) *eventually $g_J(\hat{\pi}) = 0$.*

The dynamic method can only end up working with a fixed set J if eventually $SepOr(\hat{\pi}) \subseteq J$. Then, by (12)*, $g_{L\setminus J}(\hat{\pi}) \leq 0$.*

Since $g(\hat{\pi}) \leq 0$, all the hard constraints are satisfied at $\hat{\pi}$. By weak duality, $\hat{\pi} \in$ conv $\mathcal{Q}$ solves a convexified form of the primal problem. In turn, this implies that the dynamic method not only minimizes f on the set $\{x \geq 0 : x_{L\setminus J} = 0\}$, but also solves the full dual problem (2)*.*

Theorem 6 below states formally some of these facts in a rather general setting.

5. The algorithm

We now give the dynamic bundle procedure in full details. The algorithm requires the choice of some parameters, namely m, tol, and k_{comp}. The Armijo-like parameter m is used to measure if a decrease is significant (and, thus, to decide if a new stability center should be declared), the tolerance tol is used in the stopping test. Finally, k_{comp} denotes the first serious-step iteration triggering our *complementarity rule*, explained in Remark 4 below.

Algorithm 1 (Dynamic Bundle Method).

Initialization. Choose $m \in (0, 1)$, a tolerance $tol \geq 0$, and a nonnegative integer k_{comp}. Choose a non null $x^0 \geq 0$ and define $J_0 \supseteq \{j \leq n : x^0_j > 0\}$.

Make the dual calculations to obtain $p^0 = DualEval(x^0)$.
Set $k = 0$, $\hat{x}^0 := x^0$, $\hat{J}_0 := J_0$, $\ell = 1$ and $J_1 := J_0$.
Choose a positive parameter μ_1.
Define the oracle bundle $\mathcal{B}_1 := \{(C_0 := C(p^0), p^0)\}$.

Step 1. (Dual iterate generation) Solve the QP problem:

$$\left(x^\ell, \hat{s}^\ell, \hat{\nu}^\ell, \Delta_\ell, \alpha^\ell\right) = QPIt(\hat{x}^k, \check{f}_\ell, J_\ell, \mu_\ell)\,.$$

Make the dual computations

$$p^\ell = DualEval(x^\ell)$$

and compute $C_\ell := C(p^\ell)$ and $g(p^\ell)$.

Step 2. (Separation) For $\hat{\pi}^\ell = \sum_{i \in \mathcal{B}_\ell} \alpha_i^\ell p^i$ call the separation procedure to compute

$$I_\ell := SepOr(\hat{\pi}^\ell).$$

Step 3. (Descent test)
If $f(x^\ell) \leq f(\hat{x}^k) - m\Delta_\ell$ **then** declare a **serious step** and move the stabilization center: $\hat{x}^{k+1} := x^\ell$.
Otherwise, declare a **null step**.

Step 4. (Stopping test) If $\Delta_\ell \leq tol$ and $I_\ell \subseteq J_\ell$ **stop**.

Step 5. (Bundle Management) Choose a *reduced bundle* $\mathcal{B}_{red}$; see Remark 4 below for different options.
Define $\mathcal{B}_{\ell+1} := \mathcal{B}_{red} \cup \{(C_\ell, p^\ell)\}$.

Step 6. (Constraints Management)
If a **serious step** was declared,
If $k > k_{comp}$ and $I_\ell \not\subseteq J_\ell$, take $O_{k+1} := \emptyset$.
Otherwise, compute $O_{k+1} := \{j \in J_\ell : \hat{x}_j^{k+1} = 0\}$.
Define $\hat{J}_{k+1} := J_\ell \backslash O_{k+1}$, $J_{\ell+1} := \hat{J}_{k+1} \cup \Big(I_\ell \backslash J_\ell\Big)$, and set $k = k+1$.
If a **null step** was declared, define $J_{\ell+1} := J_\ell \cup \Big(I_\ell \backslash J_\ell\Big)$.

Loop and update Choose a positive $\mu_{\ell+1}$. Set $\ell = \ell + 1$ and go to Step 1. □

In this algorithmic pattern, there are two distinct parts: a *dual phase*, corresponding to Steps 1, 3, and 5; and a *primal phase*, corresponding to Steps 2 and 6. In particular, Step 6 is independent of the bundle method utilized for the dual phase computations. We comment on some important features of both phases.

Remark 4.

– *Constraint management (serious steps).* Let ℓ_k denote an iteration when a serious step is declared ($\hat{x}^{k+1} = x^{\ell_k}$). While $k \leq k_{comp}$, the new index set J_{ℓ_k+1} is defined by removing from J_{ℓ_k} those indices corresponding to zero components in $\hat{x}^{k+1}$. However, after generating k_{comp} serious steps, indices will be removed *only* if $I_{\ell_k} \subseteq J_{\ell_k}$. As a result, the relation

$$|I_\ell \backslash J_\ell||O_{k+1}| = 0 \quad \text{if } k > k_{comp} \text{ and } \hat{x}^{k+1} = x^\ell$$

holds. Due to its similarity with a complementarity condition in nonlinear programming, we call this relation *complementarity rule*. In practice, the parameter k_{comp} plays an important role. It gives the algorithm the flexibility to remove inequalities which are easily identified not to be active at the optimal primal point in the early iterations and, hence, possibly reducing the cardinality of the working set J_ℓ. In particular, if the algorithm makes an infinite number of serious steps,

$$I_{\ell_k} \subseteq J_{\ell_k} \text{ for an infinite number of serious iterates } \ell_k. \tag{13}$$

Otherwise, if the cardinality of the set $K = \{k : I_{\ell_k} \subseteq J_{\ell_k}\}$ is finite, the complementary rule would imply that we removed inequalities only a finite number of times, but since we add at least one inequality on each serious step $k \notin K$, there would be an infinite number of new inequalities, contradicting the fact that L is finite.

– *Constraint management (null steps).* Serious steps allow to remove indices from the working sets J_ℓ. By contrast, at null steps, the cardinality of sets J_ℓ increases and may become "too large". In order to discourage this undesirable phenomenon, serious steps could be favored by setting a very low value for the parameter m, say $m = 0.001$. Note however that even if the dual dimension grows at null steps, the dynamic method can at worst behave like a static bundle method, because L is finite. In particular, if there is an iteration $\hat{\ell}$ where the stability center $\hat{x}^{k(\hat{\ell})}$ is generated, followed by an infinite number of null steps, then for some iteration $\ell ast \geq \hat{\ell}$ the sets J_ℓ stabilize, i.e.,

$$\exists \bar{J} \subseteq L : J_\ell = \bar{J} \text{ and } I_\ell \subseteq \bar{J} \text{ for all } \ell \geq \ell ast. \tag{14}$$

Together with the assertion in (13), relation (14) guarantees that Step 6 satisfies the following property, crucial for convergence (cf. (17) in Theorem 6):

$$I_\ell \subset J_\ell \text{ for all } \ell \text{ in an infinite set of iterations } \mathcal{L}. \tag{15}$$

In Remark 10 below we discuss alternative variants of Step 6 for managing the constraints while ensuring satisfaction of (15).

– *Bundle management.* In Step 5 the bundle can be managed by using any standard mechanism for static bundle methods. Some typical examples to define the *reduced bundle* from Step 5 are:

- no bundle deletion: keeps all elements, $\mathcal{B}_{red} := \mathcal{B}_\ell$;
- bundle selection: keeps only active elements, $\mathcal{B}_{red} := \left\{ i \in \mathcal{B}_\ell : \alpha_i^\ell > 0 \right\}$;
- bundle compression: allows to discard active elements when their cardinality is too large. In fact, it is possible to keep in the reduced bundle just one element without impairing convergence, as long as the element kept is the so-called *aggregate* couple. In the context of our Lagrangian function f, this means that

$$\mathcal{B}_{red} \supseteq \left\{ (\hat{C}_\ell, \hat{\pi}^\ell) \right\} \text{ where we defined } \hat{C}_\ell := \sum_{i \in \mathcal{B}_\ell} \alpha_i^\ell C_i.$$

The aggregate couple represents in a condensed form bundle information that is relevant for subsequent iterations. Along the iterative bundle process, when the active bundle size $|\mathcal{B}_{red}|$ becomes too big, the aggregate couple may be inserted in the index set $\mathcal{B}_{\ell+1}$ to replace active elements that were deleted from $\mathcal{B}_{red}$ (if all active elements are kept there is no need of aggregation). After compression, bundle elements have the form

$$(C, p) \in \mathbb{R} \times conv\, \mathcal{Q},$$

corresponding either to past dual evaluations ($C = C(p^i)$ and $p = p^i \in \mathcal{Q}$) or to past compressions ($C = \hat{C}$ and $p = \hat{\pi} \in conv\, \mathcal{Q}$). In both cases the affine function $C - \langle g(p), \cdot\rangle$ entering the cutting-planes model (5) stays below f for any $x \geq 0$. Note that the results stated in Lemmas 1 and 2 still hold for compressed bundles. In particular, the relation in (10a) holds because C is a concave function; we refer to [BGLS06, Ch. 10] for more details. □

6. Asymptotic Convergence Analysis.

As usual when applying a dual approach, primal points generated by Algorithm 1 can (at best) solve a *convexified* form of the primal problem (1). For this reason, an important tool for our analysis will be the following primal problem:

$$\begin{cases} \max\limits_{p} C(p) \\ p \in conv\, \mathcal{Q} \\ g_j(p) \leq 0\,, j \in J, \end{cases} \qquad conv\,(1)_J,$$

which is $conv(1)$ with L replaced by J. Before giving our convergence result for Algorithm 1, we lay down some general background on dynamic dual methods.

6.1. General convergence theory

Our first result, relating solutions to dual problems on the set $\{x \geq 0 : x_{L\setminus J} = 0\}$ to solutions to (2), puts in evidence the importance of assumption (12), independently of the method chosen to solve the dual problem.

Theorem 5. *For the function f defined in (2), with g affine, let $x^\infty \in \mathbb{R}^n$ solve the problem*

$$\begin{cases} \min\limits_{x \geq 0} f(x) \\ x_{L\setminus J} = 0\,, \end{cases} \qquad (2)_J,$$

which is dual to problem $conv\,(1)_J$. The following hold:

(i) There exists a primal point $\hat{\pi}^\infty \in conv\, \mathcal{Q}$, that solves $conv\,(1)_J$ for which $f(x^\infty) = C(\hat{\pi}^\infty)$.

(ii) If, in addition, $SepOr(\hat{\pi}^\infty) \subseteq J$ and (12) holds, then $\hat{\pi}^\infty$ solves $conv\,(1)$ and x^∞ solves (2).

Proof. Since *conv* $\mathcal{Q}$ is compact, g is an affine function, and C is finite-valued on *conv* $\mathcal{Q}$, the *modified filling property*, as defined in [HUL93, Def. XII.3.2.1, vol. II] holds. Therefore the strong duality result in (i) follows from [HUL93, Prop. XII.3.2.2, vol. II], using the fact that g is an affine mapping.

To see item (ii), we first show that $\hat{\pi}^\infty$ is feasible for *conv* (1). We already know from (i) that $g_J(\hat{\pi}^\infty) \leq 0$. In addition, the optimality condition for *conv* $(1)_J$ implies that, for $\hat{s} \in \partial f(x^\infty)$ and $\hat{\nu} \in N_J(x^\infty)$, $\hat{s} + \hat{\nu} = 0$. In particular, $\hat{\nu}_{L\setminus J} = -\hat{s}_{L\setminus J} = g_{L\setminus J}(\hat{\pi}^\infty)$, so using condition (12), we obtain that $g_{L\setminus J}(\hat{\pi}^\infty) \leq 0$ (because $g_J(\hat{\pi}^\infty) \leq 0$). Optimality of both x^∞ and $\hat{\pi}^\infty$ follows from the fact that, by item (i), $f(x^\infty) = C(\hat{\pi}^\infty)$, using the weak duality relation (3) applied to problems *conv* (1)-(2). □

Theorem 5 also highlights an important role played by the primal point $\hat{\pi}^\ell$ in Algorithm 1. It suffices to call the separation procedure only at $\hat{\pi}^\ell$ to recognize optimality under (12). In Step 2, our dynamic method calls the Separation Oracle at $\hat{\pi}^\ell$. Nonetheless, the Separation Oracle could also be called at other points to generate additional inequalities to help speeding up the overall process: typically, since points in $\mathcal{Q}$ usually have a simpler structure than points in *conv* $\mathcal{Q}$, violated inequalities in $\mathcal{Q}$ can be found with simpler separation procedures.

Rather than exact solutions x^∞ and $\hat{\pi}^\infty$, as considered in Theorem 5, bundle algorithms generate sequences that asymptotically solve the desired problem. The next result addresses this situation in a general setting. More precisely, we consider dynamic methods generating a dual-primal sequence $\{(x^\ell, \hat{\pi}^\ell)\}$ such that for the dual phase

$$\boxed{\begin{array}{l} x^\ell \text{ is computed using a model function } \check{f}_\ell \text{ and an index set } J_\ell\,, \\ \text{not necessarily via (8), but (10) holds;} \end{array}} \qquad (16)$$

while for the primal phase

$$\boxed{\begin{array}{l} \text{the separation procedure satisfies (12),} \\ \text{and the constraint management strategy satisfies (15).} \end{array}} \qquad (17)$$

Theorem 6. *Consider a dual-primal dynamic algorithm applied to the minimization problem* (2)*, with C concave, g affine, and conv $\mathcal{Q}$ compact. Suppose that, as the algorithm loops forever, the dual phase satisfies* (16) *and the primal phase satisfies* (17)*. Suppose, in addition, that the following holds:*

$$\text{the dual sequence } \left\{x^\ell\right\}_{\ell\in\mathcal{L}} \text{ is bounded}; \qquad (18a)$$

$$\lim_{\mathcal{L}\ni\ell\to\infty} |(\hat{s}^\ell + \hat{\nu}^\ell)_{J_\ell}| = 0\,; \text{ and} \qquad (18b)$$

$$\exists x^\infty \geq 0 \text{ such that } \liminf_{\mathcal{L}\ni\ell\to\infty} \check{f}_\ell(x^\ell) \geq f(x^\infty). \qquad (18c)$$

Then any cluster point $\hat{\pi}^\infty$ of the primal sequence $\{\hat{\pi}^\ell\}_{\ell\in\mathcal{L}}$ solves problem conv (1) *while x^∞ solves* (2)*, with $C(\hat{\pi}^\infty) = f(x^\infty)$.*

Proof. Let $\mathcal{L}' \subseteq \mathcal{L}$ be an infinite iteration index set such that $\hat{\pi}^\ell \to \hat{\pi}^\infty$ as $\mathcal{L}' \ni \ell \to \infty$ (recall that *conv* $\mathcal{Q}$ is compact). Since g is a continuous function, $g(\hat{\pi}^\infty) = \lim_{\mathcal{L}' \ni \ell \to \infty} g(\hat{\pi}^\ell)$. Consider $\ell \in \mathcal{L}'$: for $j \in J_\ell$, by (10b), $g_j(\hat{\pi}^\ell) \leq -(\hat{s}^\ell_j + \hat{\nu}^\ell_j)$, while for $l \in L \backslash J_\ell$ by (15) and (12), either $g_l(\hat{\pi}^\ell) \leq 0$ or $g_l(\hat{\pi}^\ell) \leq \beta g_j(\hat{\pi}^\ell)$ for some $j \in J_\ell$. When passing to the limit and using (18b), we see that $g_j(\hat{\pi}^\infty) \leq 0$ for all $j \in L$, so $\hat{\pi}^\infty$ is a feasible point for *conv* (1).

Similarly to the argument in Theorem 5, we use (10) to show that $C(\hat{\pi}^\infty) \geq f(x^\infty)$ and then apply the weak duality property (3) to conclude that the dual-primal pair $(x^\infty, \hat{\pi}^\infty)$ is optimal:

$$\begin{aligned} C(\hat{\pi}^\ell) &\geq \check{f}_\ell(x^\ell) + \left\langle g(\hat{\pi}^\ell), x^\ell \right\rangle && \text{[by (10a)]} \\ &= \check{f}_\ell(x^\ell) - \left\langle \hat{s}^\ell + \hat{\nu}^\ell, x^\ell \right\rangle && \text{[by (10c)]} \\ &\geq \check{f}_\ell(x^\ell) - |(\hat{s}^\ell + \hat{\nu}^\ell)_{J_\ell}||x^\ell_{J_\ell}| && \text{[by } x^\ell_{L\backslash J_\ell} = 0 \text{ and Cauchy-Schwarz].} \end{aligned}$$

Passing to the limit as $\mathcal{L}' \ni \ell \to \infty$ and using (18), we obtain that $C(\hat{\pi}^\infty) \geq f(x^\infty)$ as desired. □

Some general comments on our assumptions are in order. The primal-phase condition (17) requires the availability of a "smart enough" Separation Oracle (for which property (12) holds) and a sound management of indices to be dropped out of the index set (satisfying asymptotically (15)).

Condition (18a) does not really depend on the chosen bundle method, but rather follows from properties of the primal problem (1), for example, from requiring satisfaction of some constraint qualification for g on $\mathcal{Q}$ (see (19) or (20) below).

The setting in Theorem 6 is general enough to allow dual iterates to be computed by many variants of bundle methods, not only the proximal bundle method presented here. Condition (16) essentially states that the dual phase of the method is approximately minimizing f on the set $\{x \geq 0 : x_{L\backslash J} = 0\}$. A non-exhaustive list of variants for which (16) is likely to hold is: the bundle trust region method [SZ92], the level bundle algorithm [LNN95], the generalized bundle [Fra02], the spectral bundle method [HK02], as well as the bundle, Newton, and variable metric nonsmooth methods described in [LV00]; see also [LV98] and [LV99]. With respect to the remaining conditions in the theorem, i.e., (18b) and (18c), all the bundle variants above have a convergence theory ensuring for the static version (i.e., with $J_\ell = L$ for all ℓ) that $\hat{s}^\ell + \hat{\nu}^\ell \to 0$, with $\check{f}(x^\ell) \to f(x^\infty)$. Note that the last asymptotic property is stronger than (18c). We chose this weaker form in order to potentially allow for inexact linearizations in the model functions (as in [Kiw95], [Sol03], [Kiw06], and references therein). We mention, however, that the convergence analysis is far from complete in this case: considering inexact bundle elements is not trivial and calls for substantial modifications in convergence proofs; we refer to [Kiw06] for the static case.

6.2. Convergence results for Algorithm 1

As usual in bundle methods, when the algorithm does not stop there are two mutually exclusive situations: either there is a last serious step followed by an infinite number of null steps, or there are infinitely many different serious steps. To show that the relations in (18) are satisfied when Algorithm 1 loops forever, we mainly follow [HUL93, Ch. XV.3, vol. II].

Lemma 7. *Consider Algorithm 1, applied to the minimization problem* (2)*, with* C *concave,* g *affine, and conv* $\mathcal{Q}$ *compact; and suppose the separation procedure used in Step 3 satisfies* (12)*. Suppose there is an iteration* $\hat{\ell}$ *where the stability center* $\hat{x} = \hat{x}^{k(\hat{\ell})}$ *is generated, followed by an infinite number of null steps, and let* $\ell ast \geq \hat{\ell}$ *be the iteration for which* (14) *holds.*
If $\mu_\ell \leq \mu_{\ell+1}$ *for* $\ell \geq \ell ast$ *and* $\displaystyle\sum_{\ell \geq \ell ast} \frac{\mu_\ell}{\mu_{\ell+1}^2} = +\infty$ *then there is an infinite set* $\mathcal{L}$ *of null steps such that* (15) *and* (18) *are satisfied. In particular, the relation* (18c) *holds with equality and written with* $x^\infty = \hat{x}$*.*

Proof. Satisfaction of (15) follows from (14). Moreover, since by (14) the sets J_ℓ stabilize, after iteration ℓast, the method becomes a static bundle method for problem $(2)_{\bar{J}}$, whose convergence analysis can be found in [HUL93, Ch.XV.3, vol. II]. In particular, since our assumptions on the sequence $\{\mu_\ell\}$ correspond, respectively, to conditions (3.2.7) and (3.2.8) in [HUL93, Theorem XV.3.2.4, vol. II], $\hat{x}$ minimizes f on the set $\{x \geq 0 : x_{L\setminus\bar{J}} = 0\}$. Validity of (18a), of (18b) written for $J_\ell = \bar{J}$, and of a stronger form of (18c), namely of $\liminf_{\ell \geq \ell ast} \check{f}_\ell(x^\ell) = f(\hat{x})$, can be found in [HUL93, pp. 311 and 312, vol. II]. □

The proof of Theorem 7 puts in evidence an interesting issue. As long as asymptotically we have $J_\ell \subseteq J_{\ell+1}$ at null steps, from the dual point of view it does not matter much how the index set I_ℓ is chosen, there will always be convergence and (18a) will hold. The situation is quite different at serious steps. In this case, for having bounded iterates, the dual problem should at the very least have a nonempty solution set. A sufficient condition for (2) to have minimizers is to require the function f to be *weakly coercive*, as defined in [Aus97, p. 9]; see also [Aus97, Theorem 5]. For our problem (2), f is weakly coercive if

$$\text{for all } d \geq 0 \quad \inf_{p \in \mathcal{Q}} \langle g(p), d \rangle \leq 0. \tag{19}$$

A stronger condition is to require f to be $0-$*coercive* on the nonnegative orthant, i.e., to require (19) to hold with strict inequality for all $d \neq 0$. It can be shown that f from (2) is $0-$ *coercive* if the Slater-like condition below is satisfied:

$$\begin{array}{c} \exists r \leq n+1, \text{ points } \tilde{p}^1, \tilde{p}^2, \ldots, \tilde{p}^r \in \mathcal{Q}, \text{ and convex multipliers } \tilde{\alpha}_1, \tilde{\alpha}_2, \ldots, \tilde{\alpha}_r \\ \text{such that} \quad \sum_{i \leq r} \tilde{\alpha}_i g_j(\tilde{p}^i) < 0 \quad \text{for all } j = 1, \ldots, n. \end{array} \tag{20}$$

In [Hel01], another sufficient condition ensuring 0-coercivity of the dual function therein (corresponding to a semidefinite relaxation) is given. Namely, existence of a strictly feasible point is assumed for (1). For affine constraints g, such condition is equivalent to (20) only when $\mathcal{Q}$ is convex. Depending on the primal problem considered, checking validity for condition (19) can be easier than ensuring (20), or vice versa.

Lemma 8. *Suppose the primal problem* (1) *satisfies either* (19) *or* (20)*, with* C *concave,* g *affine, and conv* $\mathcal{Q}$ *compact. Consider Algorithm 1 applied to the minimization problem* (2) *with separation procedure satisfying* (12)*. Suppose there are infinitely many serious steps and let* ℓ_k *denote an iteration index giving a serious step:* $\hat{x}^{k+1} = x^{\ell_k}$*. Then then there is an infinite set* $\mathcal{L}$ *of serious steps such that* (15) *holds. Furthermore, if* $\mu_{\ell_k} \leq \mu_{max}$ *for all* k*,* (18) *is satisfied for the whole sequence of serious steps, with* x^∞ *a cluster point of the sequence* $\{x^{\ell_k}\}$*.*

Proof. Satisfaction of (15) follows from (13). To show (18a), first note that by [Aus97, Theorem 5] satisfaction of (19) or (20) implies that f is 0-coercive. Moreover, by construction, we have

$$\check{f}_{\ell_k}(x^{\ell_k}) + \frac{1}{2}\mu_{\ell_k}|x^{\ell_k} - \hat{x}^k|^2 \leq \check{f}_{\ell_k}(\hat{x}^k) \leq f(\hat{x}^k)$$

or, equivalently using (6), $\Delta_{\ell_k} \geq \frac{1}{2}\mu_{\ell_k}|x^{\ell_k} - \hat{x}^k|^2 \geq 0$. Next, since the descent test holds, $f(\hat{x}^{k+1}) \leq f(\hat{x}^k) - m\Delta_{\ell_k}$. Therefore, the sequence $\{f(\hat{x}^k)\}$ is bounded and so is $\{x^{\ell_k} = \hat{x}^k\}$, because f is 0-coercive.
Since the optimal value $\bar{f}$ in (2) is finite, by summation over the infinite set of serious step indices, we obtain that $0 \leq m\sum_k \Delta_{\ell_k} \leq f(\hat{x}^0) - \bar{f}$. Thus, the series of nominal decreases is convergent and $\Delta_{\ell_k} \to 0$. By (6), this means that (18c) is satisfied with equality for any cluster point of the (bounded) sequence $\{x^{\ell_k}\}$. Finally, condition (18b) follows from combining the inequality $\Delta_{\ell_k} \geq \frac{1}{2}\mu_{\ell_k}|x^{\ell_k} - \hat{x}^k|^2$ with the relation

$$\hat{x}^{k+1} - \hat{x}^k = x^{\ell_k} - \hat{x}^k = \frac{1}{\mu_{\ell_k}}(\hat{s}^{\ell_k} + \hat{\nu}^{\ell_k})$$

from Lemma 1, and using the fact that $\mu_{\ell_k} \leq \mu_{max}$. □

We end this section with a convergence result gathering the main statements in Lemmas 7 and 8. Its proof is straightforward from Theorem 6, using (10) in Lemma 2 together with Lemmas 7 and 8. We mention that Theorem 9 holds for any of the bundle management strategies described in Remark 4.

Theorem 9. *Suppose the primal problem* (1) *satisfies either* (19) *or* (20)*, with* C *concave,* g *affine, and conv* $\mathcal{Q}$ *compact. Consider Algorithm 1 with separation procedure satisfying* (12) *applied to the minimization problem* (2) *and suppose it loops forever. Suppose, in addition, that at serious steps* $\mu_{\ell_k} \leq \mu_{max}$*, and at null steps*

$$\mu_\ell \leq \mu_{\ell+1}, \text{ with } \sum_{\{\ell:\ell \text{ and } \ell+1 \text{ are null steps}\}} \frac{\mu_\ell}{\mu_{\ell+1}^2} = +\infty$$

if infinitely many successive null steps occur. Then the dual-primal sequence $\{(x^\ell, \hat{\pi}^\ell)\}$ *has a cluster point* $(x^\infty, \hat{\pi}^\infty)$ *such that* x^∞ *solves* (2)*,* $\hat{\pi}^\infty$ *solves conv* (1)*, and* $C(\hat{\pi}^\infty) = f(x^\infty)$*.* □

Remark 10. As properly pointed out by a thorough referee, there are different possibilities for the constraint management strategy defining Step 6 in Algorithm 1. More specifically, any strategy yielding satisfaction of (15) could be used. In particular, from Lemmas 7 and 8, we see that an infinite sequence of steps results in $\hat{s}^\ell + \hat{\nu}^\ell \to 0$. Step 6 blocks any removal at null steps and uses the parameter k_{comp} to trigger the complementarity condition and stop dropping indices from the working index sets at serious steps. A more flexible strategy would be to prevent removals only *asymptotically*, for infinite sequences. To detect when the algorithm enters such an *asymptotic tail*, one could simply check if $|\hat{s}^\ell + \hat{\nu}^\ell|$ is smaller than a prescribed tolerance, say tol_{comp}, given at the initialization step:

Alternative Step 6. (Constraints Management)
If a **serious step** was declared,
If $|\hat{s}^\ell + \hat{\nu}^\ell| \le tol_{comp}$ and $I_\ell \not\subseteq J_\ell$, take $O_{k+1} := \emptyset$.
Otherwise, compute $O_{k+1} := \{j \in J_\ell : \hat{x}_j^{k+1} = 0\}$.
Define $\hat{J}_{k+1} := J_\ell \backslash O_{k+1}$, $J_{\ell+1} := \hat{J}_{k+1} \cup \Big(I_\ell \backslash J_\ell \Big)$, and set $k = k+1$.
If a **null step** was declared,
If $|\hat{s}^\ell + \hat{\nu}^\ell| \le tol_{comp}$ take $O_{\ell+1} := \emptyset$.
Otherwise, compute $O_{\ell+1} := \{j \in J_\ell : x_j^{\ell+1} = 0\}$.
Define $J_{\ell+1} := \Big(J_\ell \backslash O_{\ell+1} \Big) \cup \Big(I_\ell \backslash J_\ell \Big)$.

Several variants of the same idea are possible, for example for serious steps one could trigger the complementarity condition when Δ_{ℓ_k} becomes sufficiently small, and similarly for null steps.

Note also that the static setting, i.e., a classical bundle method, is covered by our algorithmic pattern, simply by taking $J_0 = L$ and defining in Step 6 the index sets as $J_{\ell+1} = L$. For this *constraints management strategy*, condition (15) is trivially satisfied. □

In our final section we address the question of when the dynamic bundle procedure has finite termination.

7. Finite termination.

In this section we address the question of finite termination of the Algorithm 1. For a polyhedral dual function f, if the stopping parameter is set to $tol = 0$, and the bundle management is either "no bundle deletion" or "bundle selection" (recall Remark 4), we provide a positive answer for that question. We proceed somewhat similarly to [Kiw85, Ch. 2.6], using the asymptotic dual results to obtain a contradiction and, thus, showing finite termination.

Our main assumption is that there exists a finite number q of primal points,

$$\{p^1, p^2, \ldots, p^q\} \subset \mathcal{Q},$$

such that the dual function can be written as

$$f(x) = \max_{i \le q} \left\{ C(p^i) - \left\langle g(p^i), x \right\rangle \right\}, \tag{21}$$

i.e., the dual function is polyhedral.

Although this assumption is made on the dual function, there are many primal conditions that ensure the form (21) for f. For example, if the set $\mathcal{Q}$ itself is finite (which is the case in many combinatorial problems), or if C is affine and $\mathcal{Q}$ is a compact polyhedral set. Therefore, (21) provides us with an abstract framework that can be easily verified for many primal problems.

Condition (21) implies that at each given x^i there are at most q different maximizers p^i as in (4), yielding a finitely generated subdifferential

$$\partial f(x^i) = conv\left\{-g(p^i) : p^i \text{ from (4) and } i \le q\right\}.$$

Likewise, bundle elements corresponding to past dual evaluations, i.e,

$$\Big(C_i = C(p^i), p^i \in \mathcal{Q}\Big), \text{ where } i \le q,$$

can only take a finite number of different values. This is not the case for aggregate elements, which can take an infinite number of values, simply because they have the expression

$$\Big(\hat{C} = \sum_i \alpha_i C(p^i), \hat{\pi} = \sum_i \alpha_i p^i\Big),$$

where $\hat{\pi} \in conv\,\mathcal{Q}$. This is the underlying reason why we in our next result we cannot handle the "bundle compression" strategy from Remark 4.

Theorem 11. *Suppose the primal problem* (1) *satisfies either* (19) *or* (20)*, with g affine, $conv\,\mathcal{Q}$ compact, and a dual function of the form* (21)*. Consider Algorithm 1 applied to the minimization problem* (2) *with separation procedure satisfying* (12)*. Suppose $tol = 0$ and that Step 5 of the algorithm always sets the bundle management strategy to be either "no bundle deletion" or "bundle selection". If at null steps $\mu_\ell = \mu_{k(\ell)}$, while at serious steps $\mu_{\ell_k} \le \mu_{max}$, then the algorithm stops after a finite number of iterations having found a primal convex point $\hat{\pi}^{\ell ast}$, solution to $conv$* (1)*, with $x^{\ell ast}$ solving* (2).

Proof. For contradiction purposes, suppose Algorithm 1 loops forever and note that our assumptions on $\{\mu_\ell\}$ satisfy the conditions in Lemmas 7, 8, and Theorem 9. To begin, suppose that there is a last serious step $\hat{x}$ followed by infinitely many null steps and let $\hat{\mu}$ denote the corresponding proximal parameter. Note first that, since $(C(p^\ell), p^\ell) \in \mathcal{B}_{\ell+1}$ for any bundle management strategy, having $x^{\ell+1} = x^\ell$ implies that $f(x^{\ell+1}) = f(x^\ell) = C(p^\ell) - \langle g(p^\ell), x^{\ell+1}\rangle$. But since by construction $\check{f} \le f$ and $\check{f}_{\ell+1}(x^{\ell+1}) \ge C(p^\ell) - \langle g(p^\ell), x^{\ell+1}\rangle$, we conclude that $\check{f}_{\ell+1}(x^{\ell+1}) = f(x^{\ell+1})$ and, hence, $\Delta_{\ell+1} = f(\hat{x}) - f(x^{\ell+1})$. As a result, the non satisfaction of the descent test in Step 3 of Algorithm 1 yields the inequality $f(x^{\ell+1}) \ge f(\hat{x})$. On the other hand, by (8), $f(x^{\ell+1}) + \frac{1}{2}\hat{\mu}|x^{\ell+1} - \hat{x}|^2 = \check{f}_{\ell+1}(x^{\ell+1}) + \frac{1}{2}\hat{\mu}|x^{\ell+1} - \hat{x}|^2 \le f(\hat{x}) \le f(x^{\ell+1})$, so $\hat{x} = x^{\ell+1}$. In this case the algorithm would eventually stop ($\Delta_{\ell+1} = 0$ and (14) holds), contradicting our

starting assumption. Thus, infinite null steps occur only with $x^{\ell+1} \neq x^\ell$.
For $\ell \geq \ell ast$, consider the following rewriting of problem (8):

$$\begin{cases} \min_{r\in\mathbb{R},x\in\mathbb{R}^n} r + \frac{1}{2}\hat{\mu}|x-\hat{x}|^2 \\ r \geq C(p^i) - \langle g(p^i), x\rangle \text{ for } i \in \mathcal{B}_\ell \\ x_{\bar{J}} \geq 0 \text{ and } x_{L\setminus\bar{J}} = 0, \end{cases} \tag{22}$$

and denote its optimal value $\mathcal{O}_\ell := \check{f}_\ell(x^\ell) + \frac{1}{2}\hat{\mu}|x^\ell - \hat{x}|^2$. Relation (3.2.9) in [HUL93, p. 311, vol. II],

$$\mathcal{O}_{\ell+1} \geq \mathcal{O}_\ell + \frac{1}{2}\hat{\mu}|x^{\ell+1} - x^\ell|^2,$$

together with the fact that $x^{\ell+1} \neq x^\ell$, imply that the values of $\mathcal{O}_\ell$ are strictly increasing. The assumption that $\mathcal{Q}$ is finite implies that $\mathcal{B}_\ell$ contains at most q different pairs (C_i, p^i). As a result, there is a finite number of different feasible sets in (22) for $\ell \geq \ell ast$, contradicting the fact that the (infinite) values of $\mathcal{O}_\ell$ are strictly increasing.

Next we assume there is an infinite number of serious steps. Since Lemma 8 applies with our assumptions, (18b) is satisfied for the whole set of serious iterations, i.e., by $\mathcal{L} := \{\ell_k : 0 \leq k \leq +\infty\}$. Moreover, using (13), we can take an infinite subsequence $\mathcal{L}' \subset \mathcal{L}$ such that $I_{\ell_k} \subset J_{\ell_k}$ for all $\ell_k \in \mathcal{L}'$. Consider $\ell_k \in \mathcal{L}'$. By (9), $\hat{s}^{\ell_k} \in \partial\check{f}_{\ell_k}(\hat{x}^{k+1})$ and $\hat{\nu}^{\ell_k} \in N_{J_{\ell_k}}(\hat{x}^{k+1})$. Since $\mathcal{Q}$ is finite there is only a finite number of different combinations of J_{ℓ_k} and $\mathcal{Q}_{\ell_k} := \{p^i \in \mathcal{Q} : i \in \mathcal{B}_{\ell_k}\}$. Therefore, see for example [Kiw91, Lemma 4.1], there exists $\rho > 0$ such that $|\hat{s}^{\ell_k} + \hat{\nu}^{\ell_k}| < \rho$ implies that $\hat{s}^{\ell_k} + \hat{\nu}^{\ell_k} = 0$. As a result, using(8), $\hat{x}^{k+1} = x^{\ell_k+1} = \hat{x}^k$. For this value of $\hat{x}^{k+1}$, the descent test in Step 3 of Algorithm 1 (which must hold because ℓ_k gave a serious step) becomes $f(\hat{x}^k) \leq f(\hat{x}^k) - m\Delta_{\ell_k}$. This inequality is only possible if $\Delta_{\ell_k} = 0$, or, using in (6) that $\hat{x}^{k+1} = \hat{x}^k$, if

$$f(\hat{x}^{k+1}) = \check{f}_{\ell_k}(\hat{x}^{k+1})\,. \tag{23}$$

Let k' be the first index such that $\hat{s}^{\ell_{k'}} + \hat{\nu}^{\ell_{k'}} = 0$, i.e. such that $\hat{x}^{k'+1}$ minimizes $\check{f}_{\ell_{k'}}$ on the set $\{x \geq 0 : x_{L\setminus J_{\ell_{k'}}} = 0\}$. Combining the relation $\hat{\nu}^{\ell_{k'}} = -\hat{s}^{\ell_{k'}}$ with (13) and (12) we see that for every $l \in L\setminus J_{\ell_{k'}}$ there is an index $j \in J_{\ell_{k'}}$ such that

$$\hat{\nu}_l^{\ell_{k'}} = -\hat{s}_l^{\ell_{k'}} = g_l(\hat{\pi}^{\ell_{k'}}) \leq \beta g_j(\hat{\pi}^{\ell_{k'}}) = -\beta\hat{s}_j^{\ell_{k'}} = \beta\hat{\nu}_j^{\ell_{k'}} \leq 0,$$

so $\hat{x}^{k'+1}$ solves the problem $\min_{x\geq 0} \check{f}_{\ell_{k'}}(x)$ by Corollary 3.

In particular, for the cluster point x^∞ from Lemma 8, solution to (2), it holds that $\check{f}_{\ell_{k'}}(\hat{x}^{k'+1}) \leq \check{f}_{\ell_{k'}}(x^\infty)$. Since $\check{f} \leq f$ by construction, we see that

$$\check{f}_{\ell_{k'}}(\hat{x}^{k'+1}) \leq f(x^\infty), \tag{24}$$

and by (23), this means that $f(x^\infty) \geq f(\hat{x}^{k'+1})$, i.e, the relation is satisfied with equality, because x^∞ solves (2). Therefore, the algorithm would have $\Delta_{\ell_{k'}} = 0$ with $I_{\ell_{k'}} \subseteq J_{\ell_{k'}}$, and the stopping test would be activated. Since $x^{\ell_{k'}}$ solves (2), the corresponding primal point $\hat{\pi}^{\ell_{k'}}$ solves *conv* (1), by Theorem 5. □

It is worth mentioning that finite termination results in the (static) bundle literature need to modify the descent test in Step 3 by setting the Armijo-like parameter m equal to 1. Such requirement is used for example in [Kiw85, Ch. 2.6] and [Kiw91] to show that there can only be a finite number of serious steps when the function f is polyhedral, with the same assumptions on the bundle management strategy, i.e., either no deletion or bundle selection. Since the static case is covered by the dynamic setting, Theorem 11 extends the known results of finite termination to include the case $m \in (0, 1)$.

8. Concluding Remarks

We have given general convergence results for using bundle methods in a dynamic setting. Although the dynamic approach has already been considered in other recent works, the theoretical analysis presented in this paper not only provides justification to the general dynamic bundle methodology, but also gives implementable rules for removing inequalities that are sound both computationally and theoretically. More precisely, by means of a constructive complementarity rule we ensure convergence without artificial assumptions on the sequence of iterates. The complementarity rule is general enough to allow nontrivial resettings of inequalities preventing, at the same time, eventual cycling behavior.

The convergence analysis strongly relies on the assumption that the separation procedure is *smart* enough. It is important to notice that any dynamic dual method will naturally be limited by the quality of the available Separation Oracle. Our results require the separation procedure to be good enough, in the sense of (12), but the method still provides enough flexibility to combine the Separation Oracle with fast heuristics capable of speeding up the process. To evaluate the impact of these assumptions, in [BS06] we analyze the behavior of Algorithm 1 for two combinatorial problems. The first test set is the integer programming formulation of linear ordering problem, [GJR84]. For this case, (12) is satisfied, because there is only a polynomial number of dynamically dualized constraints that can be completely enumerated in an easy and efficient manner. The second test set is the traveling salesman problem, in a dual formulation that is stronger than the classical Held and Karp bound, and uses r-Regular t-Paths Inequalities, following the approach in [BL02]. For this case, there is not known exact separation method and satisfaction of (12) cannot be ensured (i.e., we need to rely on heuristics). For our comparisons, we use a dynamic subgradient method, as in [BL02] and [Luc92]. Although the numerical results in [BS06] use a simple implementation requiring vast improvement in terms of computational times, we observe an impressive superiority of the dynamic bundle method over subgradient methods in terms of duality gaps, with better performance when (12) is satisfied, as expected.

Acknowledgements. We are grateful to Krzysztof C. Kiwiel for his careful reading and constructive comments. We also would like to thank three anonymous referees and the associate editor for many suggestions that improved the contents of this paper.

References

[Aus97] A. Auslender. How to deal with the unbounded in optimization: Theory and algorithms. In *Mathematical Programming*, volume 79, pages 3–18, 1997.

[BA00] F. Barahona and R. Anbil. The volume algorithm: producing primal solutions with a subgradient method. *Math. Program.*, 87(3, Ser. A):385–399, 2000.

[BC81] E. Balas and N. Christofides. A restricted Lagrangian approach to the traveling salesman problem. *Math. Program.*, 21(1): 19–46, 1981.

[BdMV05] F. Babonneau, O. du Merle, and J.-P. Vial. Solving large scale linear multicommodity flow problems with an active set strategy and proximal-ACCPM. *Operations Research*, 2005.

[Bea90] J. E. Beasley. A Lagrangian heuristic for set-covering problems. *Naval Res. Logist.*, 37(1):151–164, 1990.

[BGLS06] J.F. Bonnans, J.Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal. *Numerical Optimization. Theoretical and Practical Aspects*. Universitext. Springer-Verlag, Berlin, 2006 (second edition). xiv+490 pp.

[BL02] A. Belloni and A. Lucena. Improving on the Help and Karp Bound for the STSP via Lagrangian Relaxation. Working paper, 2002.

[BMS02] L. Bahiense, N. Maculan, and C. Sagastizábal. The Volume Algorithm revisited. Relation with Bundle Methods. *Math. Program., Ser. A*, 94(1):41–69, 2002.

[BS06] A. Belloni and C. Sagastizábal. Numerical assessment of dynamic bundle methods in combinatorial optimization, 2006. In preparation.

[CB83] N. Christofides and J. E. Beasley. Extensions to a Lagrangean relaxation approach for the capacitated warehouse location problem. *European J. Oper. Res.*, 12(1):19–28, 1983.

[CG59] E. Cheney and A. Goldstein. Newton's method for convex programming and Tchebycheff approximations. *Numerische Mathematik*, 1:253–268, 1959.

[EGM94] L. F. Escudero, M. Guignard, and K. Malik. A Lagrangian relax-and-cut approach for the sequential ordering problem with precedence relationships. *Ann. Oper. Res.*, 50:219–237, 1994. Applications of combinatorial optimization.

[Erm66] Ju. M. Ermol′ev. Method for solving nonlinear extremal problems. *Kibernetika (Kiev)*, 2(4):1–17, 1966.

[FG99] A. Frangioni and G. Gallo. A bundle type dual-ascent approach to linear multicommodity min cost flow problems. *INFORMS Journal On Computing*, 11(4):370–393, 1999.

[FGRS06] I. Fischer, G. Gruber, F. Rendl, and R. Sotirov. Computational experience with a bundle approach for semidefinite cutting plane relaxations of max-cut and equipartition. *Math. Programming*, 105(2-3, Ser. B):451–469, 2006.

[FLR06] A. Frangioni, A. Lodi, and G. Rinaldi. New approaches for optimizing over the semimetric polytope. *Math. Programming*, 104(2-3, Ser. B):375–388, 2006.

[Fra02] A. Frangioni. Generalized bundle methods. *SIAM Journal on Optimization*, 13(1):117–156, 2002.

[Gav85] B. Gavish. Augmented Lagrangian based algorithms for centralized network design. *IEEE Transactions on Communication*, 33:1247–1257, 1985.

[GJR84] M. Grötschel, M. Jünger, and G. Reinelt. A cutting plane algorithm for the linear ordering problem. *Oper. Res.*, 32(6):1195–1220, 1984.

[Hel01] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. The sharpest cut, 233–256, MPS/SIAM Ser. Optim., SIAM, Philadelphia, PA, 2004.

[HFK01] M. Hunting, U. Faigle, and W. Kern. A Lagrangian relaxation approach to the edge-weighted clique problem. *European J. Oper. Res.*, 131(1):119–131, 2001.

[HK02] C. Helmberg and K. C. Kiwiel. A spectral bundle method with bounds. *Math. Programming*, 23(2, Ser. A):173–194, 2002.

[HUL93] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms*. Number 305-306 in Grund. der math. Wiss. Springer-Verlag, 1993. (two volumes).

[HWC74] M. Held, Ph. Wolfe, and H.P. Crowder. Validation of subgradient optimization. *Math. Programming*, 6:62–88, 1974.

[Kel60] J. E. Kelley. The cutting plane method for solving convex programs. *J. Soc. Indust. Appl. Math.*, 8:703–712, 1960.

[Kiw85] K.C. Kiwiel. *Methods of descent for nondifferentiable optimization*. Springer-Verlag, Berlin, 1985.

[Kiw91] K.C. Kiwiel. Exact penalty functions in proximal bundle methods for constrained convex nondifferentiable minimization. *Math. Programming*, 52(2, Ser. B):285–302, 1991.

[Kiw95] K.C. Kiwiel. Approximations in Proximal Bundle Methods and Decomposition of Convex Programs. *Journal of Optimization Theory and Applications*, 84:529–548, 1995.

[Kiw06] K. C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM J. Optim.*, 16:1007–1023, 2006.

[LB96] A. Lucena and J. E. Beasley. Branch and cut algorithms. In *Advances in linear and integer programming*, volume 4 of *Oxford Lecture Ser. Math. Appl.*, pages 187–221. Oxford Univ. Press, New York, 1996.

[LNN95] C. Lemaréchal, A. Nemirovskii, and Yu. Nesterov. New variants of bundle methods. *Math. Program.*, 69:111–148, 1995.

[Luc92] A. Lucena. Steiner problem in graphs: Lagrangean relaxation and cutting-planes. *COAL Bulletin*, 21(2):2–8, 1992.

[LV98] L. Lukšan and J. Vlček. A bundle-Newton method for nonsmooth unconstrained minimization. *Math. Programming*, 83(3, Ser. A):373–391, 1998.

[LV99] L. Lukšan and J. Vlček. Globally convergent variable metric method for convex nonsmooth unconstrained minimization. *J. Optim. Theory Appl.*, 102(3), 1999.

[LV00] L. Lukšan and J. Vlček. NDA: Algorithms for nondifferentiable optimization. Technical report, Institute of Computer Science, Academy of Sciences of the Czech Republic, 2000.

[RS03] A. Ruszczynski and A. Shapiro (eds.). *Stochastic Programming.* Number 10 in "Handbooks in Operations Research and Management Science". Elsevier Science, Amsterdam, 2003.

[RS06] F. Rendl and R. Sotirov. Bounds for the quadratic assignment problem using the bundle method. *Math. Programming*, Ser. B, 2006. Accepted for publication.

[Sho85] N. Shor. *Minimization methods for non-differentiable functions.* Springer-Verlag, Berlin, 1985.

[Sol03] M. V. Solodov. On approximations with finite precision in bundle methods for nonsmooth optimization. *Journal of Optimization Theory and Applications*, 119:151–165, 2003.

[SZ92] H. Schramm and J. Zowe. A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results. *SIAM Journal on Optimization*, 2(1):121–152, 1992.