Università di Pisa

Dipartimento di Informatica

Technical Report

Standard Bundle Methods: Untrusted Models and Duality

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
Largo B.Pontecorvo 3, 56127 Pisa, Italia, frangio@di.unipi.it

June 14, 2018

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

Standard Bundle Methods: Untrusted Models and Duality

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
Largo B.Pontecorvo 3, 56127 Pisa, Italia, frangio@di.unipi.it

June 14, 2018

Abstract

We review the basic ideas underlying the vast family of algorithms for nonsmooth convex optimization known as "bundle methods". In a nutshell, these approaches are based on constructing models of the function, but lack of continuity of first-order information implies that these models cannot be trusted, not even close to an optimum. Therefore, many different forms of stabilization have been proposed to try to avoid being led to areas where the model is so inaccurate as to result in almost useless steps. In the development of these methods, duality arguments are useful, if not outright necessary, to better analyze the behaviour of the algorithms. Also, in many relevant applications the function at hand is itself a dual one, so that duality allows to map back algorithmic concepts and results into a "primal space" where they can be exploited; in turn, structure in that space can be exploited to improve the algorithms' behaviour, e.g. by developing better models. We present an updated picture of the many developments around the basic idea along at least three different axes: form of the stabilization, form of the model, and approximate evaluation of the function.

Keywords: *Nonsmooth optimization, bundle methods, stabilization, decomposition, Lagrangian relaxation, duality, inexact function evaluation, incremental approach, survey*

1 Introduction

We will describe the general ideas behind a large class of algorithms for the convex minimization problem

$$f_* = \min\{ f(\boldsymbol{x}) \ : \ \boldsymbol{x} \in X \} \ , \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is proper and convex but possibly nondifferentiable. Problem (1) is quite general because of the "minimal" assumptions on how f is provided: any computational procedure (an *oracle*) that, given $\boldsymbol{x}$, returns the value $f(\boldsymbol{x})$ and information about the first-order behaviour of f at $\boldsymbol{x}$ under the form of a subgradient $\boldsymbol{z} \in \partial f(\boldsymbol{x})$ (both can actually be *approximated*, cf. §5). As far as the feasible set X is concerned, the usual assumption is that, roughly speaking, it is not making the problem significantly more complex than what the unconstrained version would be; details are given in §4.3, but on first reading one may imagine X as defined by a "small" set of explicitly known linear/conic constraints. To simplify the notation, for most of this work we will take $X = \mathbb{R}^n$; the modifications required to extend the ideas to the constrained case are, usually, simple enough as to be better introduced separately from the main analysis. We immediately remark, however, that allowing for constraints is important in that it makes it possible to deal with extended-valued f, i.e., $\operatorname{dom} f \subset \mathbb{R}^n$. In the simplest case, if only feasible iterates are produced and, say, $X \subset \operatorname{int} \operatorname{dom} f$, then f is just never evaluated at points $\boldsymbol{x}$ where $f(\boldsymbol{x}) = \infty$. It is actually possible to allow this to happen, but the oracle for f then has to provide appropriate information. In other words, we can view (1) as the unconstrained minimization of the *essential objective* $f_X = f + \imath_X$, where $\imath_X$ is the indicator function of X; then, besides an oracle for the finite-valued f, we will need a—necessarily, somewhat different—oracle for the extended-valued $\imath_X$. For most this work f will therefore be intended as finite-valued, with the other case discussed in §4.3.

The basic idea behind all *Bundle Methods* (BM) is that, being them iterative algorithms, they will

construct a sequence $\{ \boldsymbol{x}^i \}$ of iterates, hopefully converging towards some optimum $\boldsymbol{x}_*$ of (1). The oracle will be called at the points $\boldsymbol{x}^i$, producing the corresponding sequence of pairs $\{ (f(\boldsymbol{x}^i) , \boldsymbol{z}^i \in \partial f(\boldsymbol{x}^i)) \}$. Unlike algorithms for smooth optimization, that can work keeping information about a very restricted set of iterates—possibly even only the last one—BM have to resort to ideally collect and store *all* the previously generated information to work, although *compression* and *selection* procedures can usually be implemented (cf. §3.2). For a number of reasons to become apparent in due time, one customarily replaces $f(\boldsymbol{x}^i)$ with $\alpha^i = \langle \boldsymbol{z}^i, \boldsymbol{x}^i \rangle - f(\boldsymbol{x}^i)$ to define the (lower) *bundle* $\mathcal{B} = \{ (\boldsymbol{z}^i , \alpha^i) \}$. Then, the *cutting plane model*

$$\check{f}_{\mathcal{B}}(\boldsymbol{x}) = \max \left\{ \langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b \ : \ b \in \mathcal{B} \right\} \tag{2}$$

(with the useful shorthand "$b \in \mathcal{B}$" for "$(\boldsymbol{z}^b , \alpha^b) \in \mathcal{B}$") is a global *lower model* for f, i.e., $\check{f}_{\mathcal{B}} \leq f$. Upon first reading one may assume $b = i$; however, in general not all pairs in $\mathcal{B}$ are directly related with an iterate, as we shall see, whence the different index. Also, since $\mathcal{B}$ changes at each iteration it must be denoted as $\mathcal{B}^i$ which, if nothing else, justifies using a different index for its elements; we will try to simplify notation as much as possible by using, e.g., $\check{f}^i$ in place of $\check{f}_{\mathcal{B}^i}$. Note that (2) does not use (and hence $\mathcal{B}$ does not need to store) the original iterates $\boldsymbol{x}^i$, which is already a sufficient rationale for introducing the α^b; however, $\check{f}_{\mathcal{B}}$ is not the only possible (lower) model of f, and some of them actually do require storing the iterates (cf. §4.4). It is in general useful to avoid as much as possible to detail which (lower) model one uses, so that different ones can be employed (cf. §4); we will therefore generically indicate the model as $\underline{f}_{\mathcal{B}}$, although $\underline{f}_{\mathcal{B}} = \check{f}_{\mathcal{B}}$ is by far the most common choice.

With $\underline{f}_{\mathcal{B}}$ at hand, the obvious idea is to directly use it to guide the selection of the new iterate. That is, the iterative scheme

$$\boldsymbol{x}^i \in \operatorname{argmin} \left\{ \underline{f}^i(\boldsymbol{x}) \ : \ \boldsymbol{x} \in X \right\} \ , \tag{3}$$

reminiscent of the most successful algorithms for nonlinear optimization, immediately springs to mind. Of course, the new pair $(\boldsymbol{z}^i , \alpha^i = \langle \boldsymbol{z}^i, \boldsymbol{x}^i \rangle - f(\boldsymbol{x}^i))$ is then added to $\mathcal{B}^i$; on first reading one may assume that no information is ever removed from $\mathcal{B}^i$. With $\underline{f}^i = \check{f}^i$ this is the *Cutting Plane Method* (CMP) [60], whose attractive feature is that (3) can be written as

$$(\boldsymbol{x}^i , v^i) \in \operatorname{argmin} \left\{ v \ : \ v \geq \langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b \ \ b \in \mathcal{B}^i \ \ , \ \ \boldsymbol{x} \in X \right\} \ , \tag{4}$$

i.e., an LP if X is a polyhedron and in general a problem that looks "easy enough" to solve, at least if $|\mathcal{B}^i|$ is "not too large". The formulation also highlights how the natural space for the *Master Problem* (MP) (3)/(4) is the *epigraphical space* of f, with the extra variable v accounting for f-values (and $v^i = \underline{f}^i(\boldsymbol{x}^i)$). It is not surprising that the CPM is globally convergent, given that any convex function is the supremum of all its affine minorants; the proof, however, is short and instructive enough to be worth reporting.

Theorem 1 *If the level sets of the initial model $\check{f}^1$ are bounded, then $\{ \boldsymbol{x}^i \}$ in the CPM (weakly) converges to an optimal solution $\boldsymbol{x}_*$ of* (1).

Proof. As $\mathcal{B}^{i+1} \supseteq \mathcal{B}^i$, $\check{f}^i$ is monotonically nondecreasing in i, hence so are its level sets. Thus, them being bounded for $i = 1$ means they are always so, which makes (3) always well defined. Since $\check{f}^i \leq f$, this means that $f_* \geq v^i > -\infty$, and $\{ v^i \}$ is clearly nondecreasing as well. Then, the nonincreasing *record value* $f^i_{rec} = \min\{ f(\boldsymbol{x}^j) \ : \ j = 1, \ldots, i \}$ can be used to define the nonincreasing gap $g^i = f^i_{rec} - v^i \geq 0$. The aim is proving that $g^i \to 0$, which, via $f^i_{rec} \geq f_* \geq v^i$, immediately implies $f^i_{rec} \to f_*$, and therefore that, extracting subsequences if necessary, $\{ \boldsymbol{x}^i \} \to \boldsymbol{x}_*$: in fact, $f^1 \geq f^i_{rec} \geq v^i$, i.e., $\boldsymbol{x}^i \in \operatorname{lev}(v^i , f^i) \subseteq \operatorname{lev}(v^1 , f^1)$, hence $\{ \boldsymbol{x}^i \}$ is a bounded sequence. This implies that $\{ \boldsymbol{z}^i \}$ is also bounded, as the image of a compact set under the subdifferential mapping is compact [57, Proposition XI.4.1.2]. Hence, assume $g^i \geq \varepsilon > 0$: for each $j < i$, $f(\boldsymbol{x}^j) \geq f^i_{rec}$ and $\check{f}^i(\boldsymbol{x}^i) \geq f(\boldsymbol{x}^j) + \langle \boldsymbol{z}^j, \boldsymbol{x}^i - \boldsymbol{x}^j \rangle$, which gives $0 > -\varepsilon \geq \langle \boldsymbol{z}^j, \boldsymbol{x}^i - \boldsymbol{x}^j \rangle$. Taking a subsequence if necessary $\| \boldsymbol{x}^i - \boldsymbol{x}^j \| \to 0$; since $\| \boldsymbol{z}^j \|$ is bounded the right-hand side has to converge to zero, yielding the desired contradiction. ∎

A nice feature of the above proof is that constraints $\boldsymbol{x} \in X$ do not even need to be mentioned; a compact X is actually advantageous, in that compactness of $\operatorname{lev}(\check{f}^1 , \cdot)$ is clearly no longer required ($\operatorname{lev}(\check{f}^1 + \imath_X , \cdot)$ are surely compact). But for this aspect, even a cursory glance at the proof immediately suggest that the prototypical CPM is fraught with computational issues. First, it requires $\mathcal{B}$ to start "large enough" so that the model $\check{f}_{\mathcal{B}}$ is bounded below and (3) is well-defined, which is not trivial unless X is compact. Furthermore, there is no apparent way to control the size of $\mathcal{B}^i$ by removing "outdated" information. Already keeping compactness of the level sets while removing elements from $\mathcal{B}^i$ is nontrivial.

Even worse, there seem to be no way to detect whether an iterate $\boldsymbol{x}^i$ belongs or not to the convergent subsequence crucial in the argument. Indeed, it is easy to prove that "apparently reasonable" removals can lead to cycling, as the following example shows.

Example 1 *Consider Figure 1, where f is the pointwise maximum of the three linear functions (a), (b) and (c), to be minimized over $X = [\, x_a \,,\, x_b \,]$ (compact). With $\mathcal{B}^1 = \{\, (c) \,\}$, assume* (3) *returns $x^1 = x_a$, yielding $\mathcal{B}^2 = \{\, (a) \,,\, (c) \,\}$. Now assume* (3) *returns $x^2 = x_b$, so that (b) is added to $\mathcal{B}^2$. In this moment it would seem harmless to delete (a) from $\mathcal{B}^1$: the linearization has been obtained in x_a, hence "very far" from the current x^2, and it is not active (it does not contribute to defining $\check{f}^2(x^2)$). However, doing so opens the possibility that subsequently $x^3 = x_a$ with (b) being removed from $\mathcal{B}^3$, yielding a cycle. The example may seem to hinge on the fact that the linearization (c) belongs to $\mathcal{B}$ without having been produced by the oracle, and therefore without having produced the corresponding function value which contributes to the record value. This may actually happen (cf. §5), but one may easily extend the example by adding another dimension and having (c) as the intersection of two linearizations, computed (exactly) at different points.*

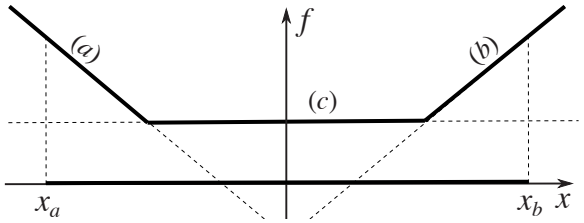


Figure 1: Example of the CPM cycling

Besides illustrating the difficulty in managing $\mathcal{B}$, the previous example also shows what is perhaps the most damning characteristic of the CPM: the approach is inherently *unstable*, with subsequent iterates possibly "very far" from each other. This is known to cause slow convergence, as clearly illustrated by the following experiment. A problem is solved by the CPM, with arbitrary initial iterate ($\boldsymbol{x}^1 = \mathbf{0}$) and $\mathcal{B}^1 = \emptyset$, and the optimal solution $\boldsymbol{x}_*$ is recorded. Then, the problem is solved again, this time with $\boldsymbol{x}^1 = \boldsymbol{x}_*$ and adding to the MP in (3) the constraint $\|\boldsymbol{x} - \boldsymbol{x}_*\|_\infty \leq \delta$ for some δ, but still taking $\mathcal{B}^1 = \emptyset$. The results are reported in Table 1, where "r.it." is the ratio between the number of iterations required by the CPM with the added constraint, for the given value of δ, and these of the initial CPM. To avoid unboundedness problems at early iterations, and for extra fairness, the MP of the CPM is actually solved with an extra constraint $\|\boldsymbol{x}\|_\infty \leq$ 1e+4; since $\|\boldsymbol{x}_*\|_\infty \approx 1$, this does not impact the correctness of the CPM.

Table 1: A conceptual experiment illustrating instability of the CPM

δ	1e+4	1e+2	1e+0	1e−2	1e−4	1e−5	1e−6
r.it.	1.07	1.12	0.86	0.77	0.56	0.19	0.04

Although these results are for a specific instance (the Lagrangian dual of a small-scale randomly generated nonempty and bounded LP), they are quite typical. Knowledge of $\boldsymbol{x}_*$, when only used to choose $\boldsymbol{x}^1 = \boldsymbol{x}_*$, is basically useless: the CPM will not perform significantly less iterations, and may easily do more. Restricting the search in a box around $\boldsymbol{x}_*$ can improve convergence, but only if the box is small enough. With a very small box, improvements of about two orders of magnitude are not unusual. All this starkly contrast with efficient algorithms for smooth optimization; ran with a starting point close to $\boldsymbol{x}_*$, these would converge extremely fast due to the use of second-order models having very good approximations of the curvature information at the optimum. A piecewise-linear $\underline{f}_{\mathcal{B}}$ such as $\check{f}_{\mathcal{B}}$ has no inherent curvature information, and therefore has to construct it piecemeal by accruing first-order information in $\mathcal{B}$. It is possible to add "poorman's" second-order information to $\underline{f}_{\mathcal{B}}$ (cf. §4.4), but this has not so far improved performances in general. BM with reliable second-order-type models have been proposed, but they require considerably more sophisticated theory [77–79]; besides, they are implicitly based on the assumption that some second-order information exists that can be extracted, which may not be the case in all applications (f, not only $\underline{f}_{\mathcal{B}}$, can well be polyhedral, cf. §3.3). Therefore, we will concentrate here on the case where $\underline{f}_{\mathcal{B}}$ is an "unstable" model like $\check{f}_{\mathcal{B}}$ can be expected to be, with the corresponding unwelcome consequences: iterates do not have any locality properties and can "swing wildly" across the search space, meaning that often the new pair $(\, \boldsymbol{z}^i \,,\, \alpha^i \,)$, when added to $\mathcal{B}^i$, conveys little useful information, failing to effectively drive $\boldsymbol{x}^{i+1}$ towards $\boldsymbol{x}_*$. As a consequence, overall

convergence of the CPM can be rather slow (although possibly with a surprising twist towards the end, cf. §2.1). This is why the CPM is more or less heavily modified, yielding the large family of (standard) BM described in this work.

The structure of this work is as follows. In Section 2 we discuss different forms of stabilization, all using the "primal" view of the problem (1), which try to address the issues illustrated above. Each time that a MP is formulated, a dual problem is implicitly defined; making it explicit is often quite useful for understanding the nuances of the approaches and improving their implementation, besides suggesting even different forms of stabilization, as discussed in Section 3. Section 4 presents the other, orthogonal approach that can significantly improve the practical convergence rate of a BM: exploiting specific structures in f to develop specialized models. Finally, since the cost of computing f can be considerable in some applications, another way of improving the practical efficiency of BM is allowing to perform this computation only approximately, which is discussed in Section 5. Section 6 briefly reviews a number of issues that have not been addressed in this work and draws some conclusions.

2 Stabilization

The previous discussion has illustrated the need for *stabilizing* the CPM, i.e., ensuring that the iterates do not stray too far from a properly chosen point. However, in general the "right" point—ideally $\boldsymbol{x}_*$—is unknown, and therefore has to be estimated and revised iteratively. Hence, together with the sequence $\{\, \boldsymbol{x}^i \,\}$ of iterates one has to consider the sequence $\{\, \bar{\boldsymbol{x}}^i \,\}$ of *stability centers* which, as we shall see, is actually the one that matters most in terms of convergence properties of the algorithm. It is quite natural (although not strictly necessary [2]) to assume that the stability centers are chosen among the iterates, i.e., $\{\, \bar{\boldsymbol{x}}^i \,\} \subseteq \{\, \boldsymbol{x}^i \,\}$; a convenient consequence is that typically $\underline{f}^i(\bar{\boldsymbol{x}}^i) = f(\bar{\boldsymbol{x}}^i)$. Several different variants of BM correspond to different ways of ensuring that $\boldsymbol{x}^i$ is "near enough" to $\bar{\boldsymbol{x}}^i$. As the example has illustrated, having a "good" $\bar{\boldsymbol{x}}^i$ is not, by itself, enough: one also have to properly estimate "how near" $\boldsymbol{x}^i$ has to be kept. While one can expect the answer "as near as possible" to be correct when $\bar{\boldsymbol{x}}^i = \boldsymbol{x}_*$, in general this is not so, and an excessive stabilization is as detrimental as an insufficient one (cf. Figure 2). Hence, each BM will also have some *stabilization parameters* controlling this aspect, again with a different meaning for each different variant.

2.1 Trust-region stabilization

A simple approach closely mimics our conceptual example by solving the *stabilized* MP

$$\boldsymbol{x}^i \in \operatorname{argmin} \left\{\, \underline{f}^i(\boldsymbol{x}) \,:\, \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\| \leq \delta^i \,\right\} \;, \tag{5}$$

where the iterate is kept in a *Trust Region* (TR) around the current stability center; the (single, as in most cases) stabilization parameter is δ^i, the radius of the trust region. Usually the norm in (5) is the L_∞ one, because then the natural "explicit form" of (5)

$$(\, \boldsymbol{x}^i \,,\, v^i \,) \in \operatorname{argmin} \left\{\, v \,:\, v \geq \langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b \;\; b \in \mathcal{B}^i \;\;,\;\; \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\| \leq \delta^i \,\right\} \tag{6}$$

is an LP; this justifies the "BOXSTEP" name originally given to the *Trust-Region* BM (TRBM) [76], although the exact form of the TR is largely immaterial. Of course, rules to update $\bar{\boldsymbol{x}}^i$ and δ^i need be defined. For the latter, a simple boundedness condition $0 < \underline{\delta} \leq \delta^i \leq \bar{\delta} < \infty$ is sufficient. The former can be done in a natural way with an Armijo-type condition:

$$f(\boldsymbol{x}^i) \leq f(\bar{\boldsymbol{x}}^i) + m(\, \underline{f}^i(\boldsymbol{x}^i) - f(\bar{\boldsymbol{x}}^i) \,) \quad \equiv \quad \Delta f^i \leq m \Delta^i \tag{7}$$

where $m \in (0,1)$ is fixed and $\Delta^i = \underline{f}^i(\boldsymbol{x}^i) - f(\bar{\boldsymbol{x}}^i) = v^i - f(\bar{\boldsymbol{x}}^i) < 0$, $\Delta f^i = f(\boldsymbol{x}^i) - f(\bar{\boldsymbol{x}}^i)$ are, respectively, the improvement estimated by the model and the actual one due to moving from $\bar{\boldsymbol{x}}^i$ to $\boldsymbol{x}^i$. If $\Delta^i = 0$, then $\bar{\boldsymbol{x}}^i$ is optimal for (1): in fact, $\bar{\boldsymbol{x}}^i$ is then optimal for the MP (although this does not necessarily mean that $\bar{\boldsymbol{x}}^i = \boldsymbol{x}^i$, as the MP can have multiple optimal solutions, cf. Example 1). Hence, $\bar{\boldsymbol{x}}^i$, which is in the *interior* of the TR, is also optimal for (3) where the TR constraint is removed, which immediately implies the result. As a conseqience, $\Delta^i \leq \varepsilon$ is a convenient approximate stopping condition for the method, although one has to be careful that a small δ^i necessarily implies a small Δ^i. Whenever (7) holds the, *tentative point* $\boldsymbol{x}^i$ is "substantially better" than $\bar{\boldsymbol{x}}^i$, and one may reasonably set $\bar{\boldsymbol{x}}^{i+1} = \boldsymbol{x}^i$; this is usually called a *Serious Step* (SS). Leaving the stability center unchanged, i.e., $\bar{\boldsymbol{x}}^{i+1} = \bar{\boldsymbol{x}}^i$, is instead called a

Null Step (NS). Clearly, (7) ensures that $\{ f(\bar{\boldsymbol{x}}^i) \}$ is a decreasing sequence, and in fact one typically uses $f(\bar{\boldsymbol{x}}^i)$ in place of f^i_{rec} (although the latter may be slightly better). The role of NS is instead to ensure that $\underline{f}^i$ is improved "in the neighbourhood of $\bar{\boldsymbol{x}}^i$", with the aim to ultimately attaining an accurate enough model so as to achieve descent. All in all, the method can be easily proven to be convergent.

Theorem 2 *If the level sets of f are bounded, then $\{ f(\bar{\boldsymbol{x}}^i) \} \to f_*$.*

Proof. Clearly $\{ \bar{\boldsymbol{x}}^i \} \subset \mathrm{lev}(f, f(\bar{x}^1))$ and therefore by the boundedness assumption it admits at least an accumulation point $\bar{\boldsymbol{x}}_\infty$; we want to prove that $f_\infty = f(\bar{\boldsymbol{x}}_\infty) = f_*$. The proof is divided into two distinct parts, according to the fact that $\{ \bar{\boldsymbol{x}}^i \}$ is or not a *finite* sequence.

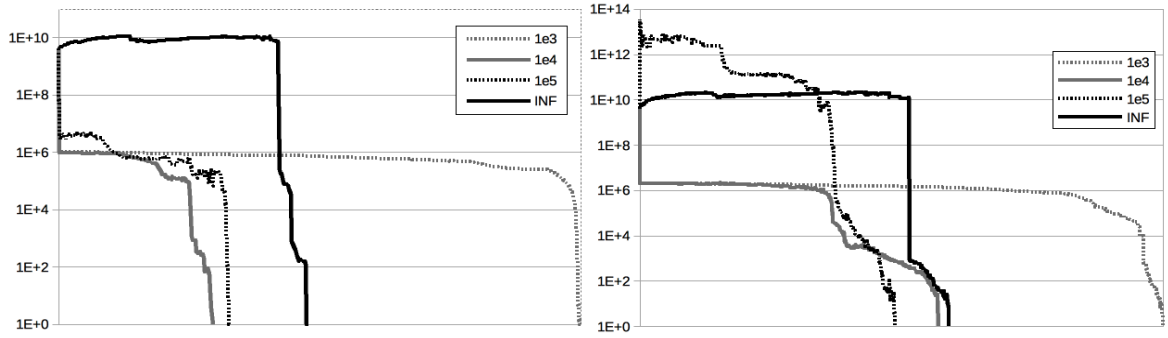
Assume that the sequence is finite: there is a last SS, after which only NS are done with the fixed stability center $\bar{\boldsymbol{x}}_\infty$. Then, because $\delta^i \le \bar{\delta} < \infty$, one is actually applying the CPM to (1) with the compact set $X := X \cap \{ \boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x} - \bar{\boldsymbol{x}}_\infty\| \le \bar{\delta} \}$. Therefore, by Theorem 1 (extracting subsequences if necessary) $\{ \boldsymbol{x}^i \} \to \boldsymbol{x}_\delta$, with $\boldsymbol{x}_\delta$ an optimal solution to that problem. If $f(\boldsymbol{x}_\delta) = f(\bar{\boldsymbol{x}}_\infty)$ then $\bar{\boldsymbol{x}}_\infty$ is an optimal solution as well, and reasoning as before therefore an optimal solution of (1). Assume by contradiction that $f(\boldsymbol{x}_\delta) - f(\bar{\boldsymbol{x}}_\infty) = \Delta_\infty < 0$ instead. From the proof of Theorem 1, $g^i = f^i_{rec} - v^i \to 0$; since $f^i_{rec} \to f(\boldsymbol{x}_\delta)$, $v^i \to f(\boldsymbol{x}_\delta)$ as well. Hence, both $\Delta^i = v^i - f(\bar{\boldsymbol{x}}_\infty) \to \Delta_\infty$ and $f(\boldsymbol{x}^i) - f(\boldsymbol{x}_\infty) \to \Delta_\infty$: since $m < 1$, this contradicts the fact that (7) never holds.

Let us now turn to the case where $\{ \bar{\boldsymbol{x}}^i \}$ is an infinite sequence, converging (extracting subsequences if necessary) to $\bar{\boldsymbol{x}}_\infty$. Clearly, (7) then implies that $\Delta^i \to 0$. For all i and any fixed optimal solution $\boldsymbol{x}_*$ to (1) define $\Gamma^i = f_* - f(\bar{\boldsymbol{x}}^i) \le 0$, and assume that $\Gamma_\infty = f_* - f(\bar{\boldsymbol{x}}_\infty) < 0$. As $\Gamma^i \le \Gamma_\infty < 0$, clearly, $\|\bar{\boldsymbol{x}}^i - \boldsymbol{x}_*\| \ge \varepsilon$ for all i and some $\varepsilon > 0$. Define $\boldsymbol{x}^i(\alpha) = \alpha \boldsymbol{x}_* + (1-\alpha)\bar{\boldsymbol{x}}^i$: by convexity, $f(\boldsymbol{x}^i(\alpha)) \le f(\bar{\boldsymbol{x}}^i) + \alpha\Gamma^i$. Also, let $\bar{\alpha}^i = \max\{ \alpha : \|\boldsymbol{x}^i(\alpha) - \bar{\boldsymbol{x}}^i\| \le \delta^i \}$: since $\delta^i \ge \underline{\delta} > 0$ and $\|\bar{\boldsymbol{x}}^i - \boldsymbol{x}_*\|$ is bounded away from 0, then $\bar{\alpha}^i$ is also bounded away from 0. But since $\boldsymbol{x}^i(\bar{\alpha}^i)$ is feasible for (5), for which $\boldsymbol{x}^i$ is the optimal solution, and $\underline{f}^i \le f$, one has $\underline{f}^i(\boldsymbol{x}^i) \le \underline{f}^i(\boldsymbol{x}^i(\bar{\alpha}^i)) \le f(\boldsymbol{x}^i(\bar{\alpha}^i)) \le f(\bar{\boldsymbol{x}}^i) + \bar{\alpha}^i\Gamma^i$. Hence, $\Delta^i = \underline{f}^i(\boldsymbol{x}^i) - f(\bar{\boldsymbol{x}}^i) \le \bar{\alpha}^i\Gamma^i$; the right-hand side is bounded away from zero, contradicting $\Delta^i \to 0$. ∎

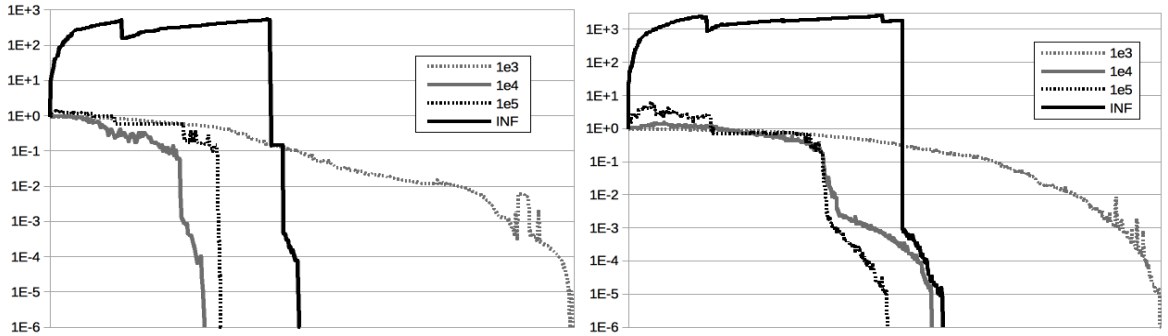
The above proof purposely used direct and elementary arguments and is obtained under unnecessarily strict conditions. For instance, boundedness of the level sets is incompatible with $f_* = -\infty$, which instead happens in applications. Also, one may want more freedom about the size of the trust region, say allowing $\delta^i \to 0$ as $\bar{\boldsymbol{x}}^i \to \boldsymbol{x}_*$. These extensions are possible, and the proof can be simplified in the process, using appropriate tools (cf. §3.4). Yet, the proof already clearly illustrates the basic machinery underlying many of BM convergence arguments. In particular, it is subdivided into two almost entirely distinct cases: that of finitely many SS, and that of infinitely many ones. In the former case, the algorithm becomes a standard CPM on the restricted feasible region and converges to an optimal solution of this problem: this has to be a global optimum, for otherwise at some point the descent condition (7) is triggered. In the latter case, the algorithm (restricted to the SS sub-sequence) is a standard descent one, and it has to converge because whenever $\bar{\boldsymbol{x}}^i$ is "far" from $\boldsymbol{x}_*$, the descent Δ^i predicted by the model cannot vanish. This almost complete separation is also apparent from the fact that the two conditions on δ^i are separately required: $0 < \underline{\delta} \le \delta^i$ is needed for SS to ensure that $\boldsymbol{x}^i - \bar{\boldsymbol{x}}^i$ does not vanish, impairing global convergence of the $\{ \bar{\boldsymbol{x}}^i \}$ sequence, whereas $\delta^i \le \bar{\delta} < \infty$ is needed to ensure that $\{ \boldsymbol{x}^i \}$ during a sequence of consecutive NS actually remains inside a finite TR around the stability center. In some sense the separation is positive: for instance, it tells that one may entirely reset $\mathcal{B}^i$ after any SS, as accumulation of information is only required to make sequences of consecutive NS to work (not that this is a good idea in practice, cf. §3.2). However, in general this disconnect makes it harder to prove properties of the method, such as global efficiency estimates.

Of course, practitioners would be more interested in the practical effect of stabilization. An illustration is given in Figure 2 for two specific problems. The figure compares how the distance from the optimal solution and the relative gap evolve during the CPM (INF) and the TRBM with three different (fixed) values of δ (1e+3, 1e+4, 1e+5).

The plots have several notable features, starting from the rather peculiar behaviour of the CPM. For the vast majority of the iterations, the algorithm seems to be making no progress: many of the first iterates $\boldsymbol{x}^i$ are far *worse* than the initial one $\boldsymbol{x}^1$, and there seems to be little, if any, sign of progress towards an optimum. However, information is indeed accrued during these iterations, and suddenly a tipping point is reached where the convergence behaviour drastically changes, becoming surprisingly quick at the end. Stabilizing may avoid the initial worsening of the iterations; even if it does not (right, $\delta = 1$+e5), it typically results in the "quick tail" ensuing sooner. Stronger stabilization may (left) or not



(a) Evolution of $\|\boldsymbol{x}^i - \boldsymbol{x}_*\|$



(b) Evolution of $(f(\boldsymbol{x}^i) - f_*)/f_*$

Figure 2: Convergence plots of TRBM with different (fixed) values for δ

(right) result in better performances: a weaker stabilization may result in worse iterates at first, but a faster convergence overall. Indeed, it is clearly possible to over-stabilize ($\delta = 1$+e3): the algorithm has then a much smoother convergence profile, but ultimately requires many more iterations. This is not surprising, in that a small TR intuitively corresponds to the fact that the algorithm behaves basically as a pure descent method (cf. §3.1): excessive stabilization does not allow to exploit the fact that the model $\underline{f}_{\mathcal{B}}$ is "global" instead of "local", and therefore potentially—provided that $\mathcal{B}$ contains enough information—capable of leading the iterate towards the global optimum, which is what happens in the "quick tail". All in all, the plots clearly show that "the right amount of stabilization" can have a positive impact; unfortunately, in general little can be said a-priori about how much stabilization is the right amount. This also depends on which stabilization device is employed, of which the TR is but one.

2.2 Proximal stabilization

The *Proximal BM* (PBM) replaces the TR with a penalty, as in

$$\boldsymbol{x}^i = \operatorname{argmin} \left\{ \underline{f}^i(\boldsymbol{x}) + \tfrac{\mu^i}{2} \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\|_2^2 \right\} \ , \tag{8}$$

where the stabilization parameter is now μ^i. The penalty term also ensures that $\boldsymbol{x}^i$ will not be "too far" from $\bar{\boldsymbol{x}}^i$, although the radius of the TR is only indirectly determined. Indeed, (8) could be viewed as the Lagrangian relaxation of (5) w.r.t. the TR constraint if the L_2 norm were used in the latter, and in principle given a δ^i one could always choose μ^i such that the two MP give the same solution, and vice-versa [57, Proposition XV.2.2.3]. The equivalence is only theoretical, since finding the value of μ^i equivalent to a given δ^i (or vice-versa) is not straightforward; not that there would be any reason for wanting to, since finding the "right" values of the two stabilization parameters in practice is roughly equally difficult. It is not entirely surprising, however, that in practice the PBM is sometimes found to be more efficient than the TRBM (e.g., [12, 43]). Indeed, the quadratic penalty term acts as a "poorman's Hessian", adding some (admittedly, very rough) second-order information to the piecewise $\check{f}_{\mathcal{B}}$; an in-depth computational evaluation of the practical behaviour of the PBM can be found e.g. in [15]. However, (8) is a QP, which may be more costly than the LP (5), potentially negating the advantage due to a faster convergence speed [40]. Yet, this can be partly counterbalanced (or even reversed [43]) by developing specialized QP algorithms that exploit the structure of the MP and its typical usage pattern [34].

An advantage of the stabilizing term is that it makes it easier to answer to an interesting question, i.e., "what would the master problem achieve if the model $\underline{f}_{\mathcal{B}}$ were exact?" That is, consider the *Moreau–Yosida regularization* ϕ_μ of f, perhaps better written in terms of the *displacement* $\boldsymbol{d}$ from $\bar{\boldsymbol{x}}$:

$$\phi_\mu(\bar{\boldsymbol{x}}) = \min \left\{ f(\bar{\boldsymbol{x}} + \boldsymbol{d}) + \tfrac{\mu}{2}\|\boldsymbol{d}\|_2^2 \right\} \ . \tag{9}$$

This is an interesting object with useful properties, starting from $\phi_\mu \leq f$ (trivial since $\boldsymbol{d} = 0$ is feasible in (9)). The unique optimal solution $\boldsymbol{d}_*$ of (9) satisfies

$$0 \in \partial[\ f(\bar{\boldsymbol{x}} + \cdot) + \tfrac{\mu}{2}\|\cdot\|_2^2\ ](\boldsymbol{d}_*) \quad \Longleftrightarrow \quad -\mu\boldsymbol{d}_* \in \partial f(\boldsymbol{x}) \ \text{ with } \ \boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{d}_* \tag{10}$$

(note that we ignore the dependence of $\boldsymbol{d}_*$ and $\boldsymbol{x}$ on both $\bar{\boldsymbol{x}}$ and μ for notational simplicity); in other words, $\boldsymbol{z}_* = -\mu\boldsymbol{d}_*$ is a (very specific) subgradient at $\boldsymbol{x}$, and $\boldsymbol{x}$ itself is obtained by starting at $\bar{\boldsymbol{x}}$ and moving of a step $1/\mu$ along $-\boldsymbol{z}_*$. Therefore, $\boldsymbol{x}$ might appear to be produced by a subgradient-type approach, were it not that $\boldsymbol{z}_*$ is a subgradient at the *destination* $\boldsymbol{x}$ rather than at the starting point $\bar{\boldsymbol{x}}$. Yet, it turns out that moving from $\bar{\boldsymbol{x}}$ to $\boldsymbol{x}$ actually *is* a step of a *gradient* method: indeed, [57, Corollary XI.3.4.1] shows that ϕ_μ is differentiable, with $\nabla\phi_\mu(\bar{\boldsymbol{x}}) = \boldsymbol{z}_*$ [57, Theorem XV.4.1.4]. Note that this depends on *smoothness* of the stabilizing term rather than, as one may guess, its strong coercivity, i.e., uniqueness of $\boldsymbol{d}_*$. Hence, $\boldsymbol{d}_* = \boldsymbol{0}$ implies that $\bar{\boldsymbol{x}}$ is both a minimum of f and of ϕ_μ: indeed, minimizing ϕ_μ is equivalent to (1) [57, Theorem XV.4.1.7], with the obvious advantage that ϕ_μ is smooth. Thus, if (9) were efficiently solvable—which it isn't, as computing just one $\boldsymbol{d}_*$ for given $\bar{\boldsymbol{x}}$ and μ is as difficult as solving (1)—then one may run a *Proximal Point Algorithm* (PPA), simply obtained by always setting $\bar{\boldsymbol{x}}^{i+1} = \bar{\boldsymbol{x}}^i + \boldsymbol{d}_*^i = \boldsymbol{x}^i$. With only minor requirements on μ^i—it must not to grow too fast, which would be analogous to $\delta^i \to 0$ very fast in the TRBM—and some technical conditions, the PPA can be shown to be a convergent algorithm. We will not go into the details of the convergence proof, which can be found e.g. in [57, §XV.4.2], besides noting that the fact that one can always take the pre-determined step $1/\mu^i$ in a gradient method and still converge is not surprising considering that $\nabla\phi_\mu$ is Lipschitz continuous with constant μ (cf. again [57, Theorem XV.4.1.4]). Said otherwise, by necessity $\phi_\mu(\boldsymbol{x}) < \phi_\mu(\bar{\boldsymbol{x}})$ (or $\boldsymbol{d}_* = 0$ and the algorithm terminates), hence the step is surely a descent one; the devious trick here is that the stepsize μ is chosen beforehand, and *the function* ϕ_μ changes to reflect the choice, i.e., in such a way that $\nabla\phi_\mu(\bar{\boldsymbol{x}})$ provides the desired descent. However, all this is only conceptual, in that ϕ_μ is not readily available. What is relevant is rather the interpretation of the PBM in terms of a PPA: basically, if "the model were perfect", i.e., $\underline{f}_{\mathcal{B}} = f$, then each iteration would result in a SS. In other words, the PBM can be seen as an approximated—but implementable, as opposed to conceptual—variant of the PPA, where sequences of consecutive NS aim at computing $\nabla\phi_\mu(\bar{\boldsymbol{x}})$ "accurately enough", so that finally a SS can be performed. This ties in well with the standard structure of convergence proofs, where sequences of consecutive NS and the sequence of SS are analysed separately.

Besides being aesthetically pleasing, these results are also the basis of practical algorithmic developments, comprised some related to the real crux of the (P)BM, which is appropriately (and dynamically) choosing the stabilization parameter. These are based on the idea that (9) could be generalized to

$$\phi_H(\bar{\boldsymbol{x}}) = \min \left\{ f(\bar{\boldsymbol{x}} + \boldsymbol{d}) + \tfrac{1}{2}\boldsymbol{d}^T H \boldsymbol{d} \right\} \tag{11}$$

depending on a whole matrix parameter $H \succ 0$, the standard Moreau–Yosida regularization then being just the special case for $H = \mu I$. Clearly it would be attractive to have H providing a better depiction of the second-order behaviour of f at $\bar{\boldsymbol{x}}$ that what the "poorman's matrix" μI can do. Of course, the Hessian cannot be used, but one may nonetheless consider quasi-Newton formulæ. Say, with $\bar{\boldsymbol{z}}_i \in \partial f(\bar{\boldsymbol{x}}_i)$ and $\boldsymbol{z}_i \in \partial f(\boldsymbol{x}_i)$, it would be natural to select H_{i+1} so that the standard *quasi-Newton equation*

$$H_{i+1}\Delta\boldsymbol{x}_i = \Delta\boldsymbol{z}_i \tag{12}$$

is satisfied with $\Delta\boldsymbol{x}_i = \boldsymbol{x}_i - \bar{\boldsymbol{x}}_i$ and $\Delta\boldsymbol{z}_i = \boldsymbol{z}_i - \bar{\boldsymbol{z}}_i$, which is what one would do if f were differentiable; which it isn't, and this entirely breaks the theory upon which (12) relies. Yet, $\phi^i = \phi_{H^i}$ is differentiable, and therefore (12) would make sense with $\Delta\boldsymbol{z}_i = \nabla\phi^i(\boldsymbol{x}_i) - \nabla\phi^i(\bar{\boldsymbol{x}}_i)$, were it not for the fact that exactly computing gradients of ϕ^i requires solving a problem as difficult as the original (1). Here, however, one can cleverly exploit (10), immediately generalized as $\nabla\phi_H(\bar{\boldsymbol{x}}) = -H\boldsymbol{d}_* \in \partial f(\boldsymbol{x})$, to *find* $\boldsymbol{d}_*$—and therefore $\bar{\boldsymbol{x}}$—*given* $\boldsymbol{z} \in \partial f(\boldsymbol{x})$. Indeed, $-H\boldsymbol{d}_* = \boldsymbol{z}$ gives $(\bar{\boldsymbol{x}} - \boldsymbol{x}) = H^{-1}\boldsymbol{z}$, i.e., $\boldsymbol{z} = \nabla\phi_H(\boldsymbol{x} + H^{-1}\boldsymbol{z})$. In plain words, once a subgradient of f is known at any point $\boldsymbol{x}$, one can easily compute the point $\bar{\boldsymbol{x}}_H(\boldsymbol{z})$ such that $\boldsymbol{z} = \nabla\phi_H(\bar{\boldsymbol{x}}_H(\boldsymbol{z}))$. This *reversal operation* [73] suggests to use (12) indeed with $\Delta\boldsymbol{z}_i = \boldsymbol{z}_i - \bar{\boldsymbol{z}}_i = \nabla\phi^i(\bar{\boldsymbol{x}}^i(\boldsymbol{z}^i)) - \nabla\phi^i(\bar{\boldsymbol{x}}^i(\bar{\boldsymbol{z}}^i))$, but also with $\Delta\boldsymbol{x}_i = \bar{\boldsymbol{x}}^i(\boldsymbol{z}^i) - \bar{\boldsymbol{x}}^i(\bar{\boldsymbol{z}}^i)$ (with the obvious notation). This may give rise to various quasi-Newton approaches depending on the way in which (12) is approached, say with

typical rank-one updates. We will not delve in these details, to be found in [73] and references therein, except for the specific case where H is constrained to have the "poorman's" form μI. This means that there is no hope that (12) be satisfied except in a least-squares sense, which yields

$$\mu_{i+1} = \|\Delta \boldsymbol{z}_i\|_2^2 \,/\, \langle \Delta \boldsymbol{z}_i, \Delta \boldsymbol{x}_i \rangle \ . \tag{13}$$

Of course, for (13) to make sense one must ensure that $\langle \Delta \boldsymbol{z}_i, \Delta \boldsymbol{x}_i \rangle > 0$, which can be done by an appropriate *curved search*, i.e., solving the MP with iteratively changing μ until the condition is attained [73]; this a natural enough approach for BM, already proposed e.g. in [88]. Under rather strong conditions (f differentiable or strongly convex), fast convergence (respectively, superlinear or two-step-superlinear) can be proven. Perhaps more importantly, the approach seems to improve practical performances w.r.t. other proposed strategies [62]. Also, the idea can be extended to making use of other available information for even better managing μ [85].

Yet, this does not imply that effective μ-management is completely understood. Formula (13) requires a specific care to ensure that $\langle \Delta \boldsymbol{z}_i, \Delta \boldsymbol{x}_i \rangle > 0$, and the theory is developed under the assumption that μ is only updated at SS, whereas intuitively being able to increase μ after a few unsuccessful NS could also be useful. Furthermore, all these approaches [62, 73, 85] are based on "local" behaviour of f, i.e., they do not explicitly depend on how "far" $\bar{\boldsymbol{x}}_i$ is from $\boldsymbol{x}_*$ ($f(\bar{\boldsymbol{x}}_i)$ from f_*), which may lead to sequences of "short" steps that slow down convergence (cf. $\delta =$ 1e+3 in Figure 2). Although more "global" strategies can be devised [38], other stabilization approaches seem to be inherently better suited in this respect, as discussed next.

2.3 Level stabilization

The idea of *level stabilization* is in some sense opposite to that of the previous approaches. In general, the issue is that $\underline{f}_{\mathcal{B}}$ is "too optimistic" a model of f, in that it dramatically underestimates the true value of f in a large part of the space. This lures the MP to points $\boldsymbol{x}^i$ such that $\underline{f}_{\mathcal{B}}(\boldsymbol{x}^i) \ll f(\bar{\boldsymbol{x}})$, often "unreasonably so", while $f(\boldsymbol{x}^i) \gg f(\bar{\boldsymbol{x}})$. The TRBM tries to se the TR in such a way as to exclude the points where $\underline{f}_{\mathcal{B}}(\boldsymbol{x}) \ll f(\bar{\boldsymbol{x}})$, while the PBM tries to limit their appeal by penalizing them on the basis of the distance from $\bar{\boldsymbol{x}}$. In these cases, the amount of descent that the model will estimate for the next iteration, as measured by $\Delta^i = \underline{f}^i(\boldsymbol{x}^i) - f(\bar{\boldsymbol{x}}^i) < 0$, is a complex function of the stabilization parameters (δ^i and μ^i). A different approach is to fix beforehand how much descent the model should attain, which clearly has an intuitive appeal in the context of a descent method, i.e., to work in the level set $\mathrm{lev}(\, \underline{f}_{\mathcal{B}} \,, \, l \,)$ for some given *level parameter* $l < f(\bar{x})$. Such a set, however, may well be "large" (even unbounded), and therefore there has to be some way pick a specific point in there. In the spirit of BM, the intuitive idea is just that of keeping "close" to the stability center, which leads to the MP

$$\boldsymbol{x}^i = \mathrm{argmin} \left\{ \, \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\| \ : \ \underline{f}^i(\boldsymbol{x}) \leq l^i \, \right\} \ . \tag{14}$$

An advantage of the resulting *Proximal Level BM* (PLBM) approach is that the stabilization parameter, l^i, has the scale of function values, which may make it easier to choose. For instance, if the optimal value f_* is *known*, then obviously l^i has to belong to the interval $[\, f(\bar{\boldsymbol{x}}^i) \,, \, f_* \,]$ (actually, $[\, f^i_{rec} \,, \, f_* \,]$). The simple strategy of fixing any $\lambda \in (0, 1]$ and choosing $l^i = \lambda f(\bar{\boldsymbol{x}}^i) + (1 - \lambda) f_*$ then works *even with very relaxed assumptions on the choice of* $\bar{\boldsymbol{x}}^i$, such as by always doing SS ($\bar{\boldsymbol{x}}^{i+1} = \boldsymbol{x}^i$) even if (7) does not hold, and even keeping $\bar{\boldsymbol{x}}^i$ (possibly, $\notin X$) fixed [70]. The proof is somewhat technical and is not repeated here; what is relevant is that knowledge of f_* is not really required, as it can be replaced by its lower bound v^i obtained by solving the original un-stabilized MP (3) (assumed finite). Solving the MP of the CPM but *not* directly using its optimal solution as the next iterate is an interesting algorithmic concept, of which we will see other applications (cf. §2.5); here, it is rather the optimal value v^i that is used to compute the value of l^i, after which (14) provides $\boldsymbol{x}^i$. Of course, the disadvantage is having to solve two (related but different) MPs, which is not appealing in the case where they are rather costly (e.g., [97]). However, in some applications the cost of computing f far outweighs the MP cost, and therefore this approach may be competitive in that it provides a clear and principled way to choose the stabilization parameter, as opposed to the heuristic ones common for the TRBM and PBM, possibly resulting in better practical convergence.

In case one is not willing to compute v^i, or unable to do so (say, because (3) is unbounded below), the alternative is to choose l^i arbitrarily. The possible troubling consequence is that (14) may be empty, but this is actually not an issue; since $\underline{f}^i \leq f$, this means that $l^i < f_*$. Hence, if this happens the algorithm

has found a provably correct lower bound on f_*, which can then be used in place of f_*/v^i to set the next target; clearly, then $l^{i+1} > l^i$, hopefully making (14) feasible. This is one of the specific traits of the PLBM, i.e., that it can provide valid lower approximations to f_*; in some cases this can be helpful. Actually, also TRBM and PBM may do this, since their next iterate $\boldsymbol{x}^i$ may in fact coincide with the optimal solution of (3), which is easy to detect; for instance, for TRBM this happens if $\boldsymbol{x}^i$ is in the interior of the TR. However, in these methods the occurrence is incidental and does not impact on the algorithm, while in the PLBM it is a crucial aspect. Hence, together with NS and SS, the convergence analysis for the PLBM has to cater for these *Level Steps* (LS); yet, this is easy. In fact, if, say, $l^{i+1} = \lambda f(\bar{\boldsymbol{x}}^i)+(1-\lambda)l^i$ whenever a LS happens, infinitely many LSs result in $l^i \to f(\bar{\boldsymbol{x}}^i)$, which means that $f(\bar{\boldsymbol{x}}^i) \to f_*$. Once this case is dealt with, the remaining analysis is analogous to the case where f_*/v^i are available, and not dissimilar from those of the TRBM and PBM.

Indeed, an interesting recent development is the *Doubly-Stabilized BM* (DSBM) of [26], which has *both* proximal and level stabilization, i.e., MP

$$\boldsymbol{x}^i = \operatorname{argmin} \left\{ \underline{f}^i(\boldsymbol{x}) + \mu^i \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\|_2^2 \,:\, \underline{f}^i(\boldsymbol{x}) \le l^i \right\} \ . \tag{15}$$

Two stabilization parameters are not necessarily more difficult to tune than one; actually, the converse may happen. Indeed, at any given iteration one among μ^i and l^i is "irrelevant": the obtained $\boldsymbol{x}^i$ is either that of (8) (a *proximal iteration*), or that of (14) (a *level iteration*), and it is easy (cf. (34)) to tell which of the two it is. Hence, the somewhat "more principled" level parameter l^i, which can exploit information about f_*, can be used to select the desired amount of descent, while μ^i can be used to select a "good" $\boldsymbol{x}^i$ in the (possibly, large) set $\operatorname{lev}(\underline{f}_{\mathcal{B}}, l^i)$; the results of [26] are encouraging. Convergence theory is hardly much different from that of PBM: once the case of infinitely many LS is ruled out, the algorithm is (almost, barring some fine details) exactly a PBM.

One may, however, argue that there is actually no need for the level stabilization in order to tune μ^i exploiting information about f_*. Firstly, any known guaranteed lower bound $l \le f_*$—such as v^i from (3), or directly obtained by the problem, cf. §3—can be directly incorporated into $\underline{f}_{\mathcal{B}}$ under the form of the "flat" linearization $(\boldsymbol{0}, l) \in \mathcal{B}$. This incurs in hardly any MP cost and it means that $\boldsymbol{x}^i$ will automatically exploit this information, which is indeed useful in practice; for instance, surely $v^i \ge l$. Furthermore, one might design μ-updating strategies that take into account this information and, say, try to ensure that $\underline{f}^i(\boldsymbol{x}^i) \le l^i = \lambda f(\bar{\boldsymbol{x}}^i) + (1-\lambda)l$ exactly as in the DSBM. Somewhat different, but related, strategies can use information about the fact that $\boldsymbol{x}^i$ is, or not, "close" to a minimizer of $\underline{f}^i$ to properly increase or decrease μ^i (cf. e.g. [35]). Current consensus is that the l-updating strategies of PLBM are more robust, in particular in the constrained case ($X \ne \mathbb{R}^n$), while the PBM may be more efficient, especially in the unconstrained case; thus, the DSBM makes sense, as would any μ-updating strategy "simulating" it. All this highlights how proper tuning of the proximal parameters is still quite an open issue, and an area of active research. This also justifies why there is, among practitioners, a latent distrust of stabilization techniques, partly justifying the development of the alternative approaches of §2.5.

2.4 Center-based approaches

Another class of BM are based on the idea that, instead of aiming for the "extreme" point $\boldsymbol{x}^i$ minimizing $\underline{f}_{\mathcal{B}}$, one should target the "center" of a *localization set* $\mathcal{L}(g,l) = \{ (\boldsymbol{x}, v) \,:\, g(x) \le v \le l \} \subset \mathbb{R}^{n+1}$, which is the epigraphical version of the level set, for appropriately chosen g and l. For instance, the polyhedron

$$\mathcal{L}^i = \mathcal{L}(\check{f}^i, f^i_{rec}) = \left\{ (\boldsymbol{x}, v) \,:\, \langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b \le v \le f^i_{rec} \ \ b \in \mathcal{B}^i \right\}$$

(cf. (4)) is clearly the best possible outer approximation—with the known data—of the epigraphical extension of the set of the optimal solutions $\mathcal{L}_* = \mathcal{L}(f, f_*) = \{ (\boldsymbol{x}, v) \,:\, f(\boldsymbol{x}) = v = f_* \}$; it is defined by the linearizations in $\mathcal{B}^i$, plus the *hat cut* $v \le f^i_{rec}$. This is obviously related with level-based BM, whose MP has feasible set is $\mathcal{L}(\underline{f}^i, l^i)$ (identical but for the hat cut). Each time the oracle is called at some $\boldsymbol{x}$ in (the projection of) $\mathcal{L}^i$, the generated information can be used to *cut away* some part of $\mathcal{L}^i$, obtaining a smaller $\mathcal{L}^{i+1}$. Indeed, if $f(\boldsymbol{x}) > \underline{f}^i(\boldsymbol{x})$ then the corresponding new linearization will at least cut away the point $(\boldsymbol{x}, \underline{f}^i(\boldsymbol{x})) \in \mathcal{L}^i$, while if $f(\boldsymbol{x}) < f^i_{rec}$ then the hat cut will be lowered; barring blatantly obvious bad choices of $\boldsymbol{x}$, at least one of the conditions must happen, and both potentially can. The idea is then to select $\boldsymbol{x}$ so that as "much as possible" of $\mathcal{L}^i$ is cut away at each iteration; intuitively, this corresponds to choosing $\boldsymbol{x}^i$ in "the center" of $\mathcal{L}^i$. Among the possible definitions of center of a polyhedron, a widely

used one is the *Analytic Center* (AC): the minimum of the *logarithmic barrier function*

$$(\boldsymbol{x}^i , v^i) = \operatorname{argmin} \left\{ -\log(f^i_{rec} - v) - \textstyle\sum_{b \in \mathcal{B}^i} \log(v - \langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b) \right\} \tag{16}$$

upon which Interior-Point (IP) methods are based. It can be alternatively defined as the point $(\boldsymbol{x}^i , v^i) \in \mathcal{L}^i$ that *maximizes the product of the slacks* of the constraints; using it as the next iterate gives the *Analytic Center CPM* (ACCPM). Clearly, this means that $\mathcal{L}^i$ must have a nonempty interior and must be bounded; the latter is in general nontrivial, exactly as in the CPM. Due to the relationships with IP methods, the (approximate) computation of the AC can be performed by means of extremely well-understood and efficient methods. Also, the known methods can be adapted to efficiently update $(\boldsymbol{x}^i , v^i)$ to $(\boldsymbol{x}^{i+1} , v^{i+1})$ when a new linearization enters $\mathcal{B}^i$ and/or the hat cut changes, a nontrivial feat because the former is typically no longer feasible, even less interior [49]. The upshot is that ACCPM has favourable worst-case complexity estimates, and usually a regular convergence profile.

As other methods explicitly constructed to optimize the worst-case, however, ACCPM is not always very fast in practice. One issue is that, as discussed in §2.1, when enough information has been accrued $\underline{f}^i$ can be quite accurate a model (especially if f itself is polyhedral), and therefore its optimum can be a promising point where to call the oracle; ACCPM not using it may lead to missing out on the "fast tail" of the CPM. There are also some specific issues due to the fact that the AC of a polyhedron depends from its *algebraic representation* rather than from its true geometry. For instance, if a linearization $(\boldsymbol{z}^b , \alpha^b)$ is generated multiple times (which happens in applications), this skews the AC to be "farther" from that. Conversely, the hat cut $v \leq f^i_{rec}$ is the only inequality limiting v from above; as $|\mathcal{B}^i|$ grows the influence of the many cuts "pushing up v from below" may overwhelm that of the hat cut, which therefore tends to become almost active. Both cases may slow down the convergence, which is based on keeping $(\boldsymbol{x}^i , v^i)$ firmly in the interior of $\mathcal{L}^i$, but specific adaptations can be devised to counter these effects [29]. Also, the issue of compactness of $\mathcal{L}^i$ can be faced by the *Proximal ACCPM* (PACCPM), a "doubly-stabilized" version [4] where a standard proximal term a-la (8) (thus introducing a proximal center $\bar{\boldsymbol{x}}$ and a proximal parameter μ) is added to (16), which is claimed to further improve the performances of the approach. Anyway, ACCPM has not been widely adopted; this is likely due, above and beyond any other reason, to the need of specific sophisticated implementations for efficiently solving (16), which cannot therefore benefit from the regular advances of general-purpose LP/QP solvers.

It is possible to avoid the need of specialized approaches to solve the MP by using the *Chebychev Center* (CC) instead, i.e., the center of the largest ball inside $\mathcal{L}^i$. For a generic polyhedron $\mathcal{P} = \{ \boldsymbol{y} \in \mathbb{R}^k : \langle \boldsymbol{a}_h, \boldsymbol{y} \rangle \leq b_h \;\; h \in H \}$, the CC is the optimal solution of the LP

$$(\boldsymbol{y} , \sigma) = \operatorname{argmax} \left\{ \sigma : \langle \boldsymbol{a}_h, \boldsymbol{y} \rangle + \|\boldsymbol{a}_h\| \sigma \leq b_h \;\; h \in H \right\} \tag{17}$$

(assuming it has any, which obviously requires $\mathcal{P}$ to be compact). When applied to $\mathcal{L}^i$, $\boldsymbol{y} = (\boldsymbol{x} , v)$ ($\mathbb{R}^k = \mathbb{R}^{n+1}$) and the hat cut $v \leq f^i_{rec}$ gives rise to a constraint of the form $v + \sigma \leq f^i_{rec}$, which is necessarily active in the optimal solution [81, Proposition 2.1]; this allows to substitute away v for $f^i_{rec} - \sigma$, which together with $\nu = -\sigma$ yields

$$(\boldsymbol{x}^i , \nu^i) \in \operatorname{argmin} \left\{ \nu : \nu \geq \tfrac{\langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b - f^i_{rec}}{1 + \sqrt{\|\boldsymbol{z}^b\|^2 + 1}} \;\; b \in \mathcal{B}^i \right\} . \tag{18}$$

The notation is chosen to highlight the similarity with the MP (4) of the CPM: besides translating the right-hand side by f^i_{rec} (which is routinely done, cf. (21)), each constraint is just scaled by a factor depending only on $\|\boldsymbol{z}^b\|$. Using $\boldsymbol{x}^i$ from (18) , which already can have a stabilization effect as in ACCPM, gives the *Chebychev Center CPM* (C^3PM) [58]. The modern take to the approach [81] views (18) as the finitely sampled version of the *Elzinga-Moore-Ouorou function*

$$\Psi(l) = \inf \left\{ \nu : \nu \geq \tfrac{\langle \boldsymbol{z}, \boldsymbol{x} - \boldsymbol{y} \rangle + f(\boldsymbol{y}) - l}{1 + \sqrt{\|\boldsymbol{z}\|^2 + 1}} \;\; \boldsymbol{y} \in \mathbb{R}^n \;\; \boldsymbol{z} \in \partial f(\boldsymbol{y}) \right\} ; \tag{19}$$

note that in (19) $\boldsymbol{x}$ and ν are the variables upon which the minimization is performed (as in (18)), whereas $\boldsymbol{y}$ and $\boldsymbol{z}$ serve to index the infinitely many linear constraints. The function $\Psi(l)$ gives the negative of the radius of the largest sphere inscribed into $\mathcal{L}(f, l)$, and therefore is a *merit function* for (1): $\Psi(l) \leq 0$, and $\Psi(l) = 0$ if and only if $l = f_*$. Therefore, (1) is equivalent to finding l such that $\Psi(l) = 0$. One can then make Ψ a multivariate function by just setting $\Psi(\boldsymbol{x}) = \Psi(f(\boldsymbol{x}))$; this could be seen as awkward, were it not that it makes it possible to add a proximal term, yielding the MP

$$\min \left\{ \nu + \tfrac{\mu^i}{2} \|\boldsymbol{x} - \bar{\boldsymbol{x}}^i\|_2^2 : \nu \geq \tfrac{\langle \boldsymbol{z}^b, \boldsymbol{x} \rangle - \alpha^b - f(\bar{\boldsymbol{x}}^i)}{1 + \sqrt{\|\boldsymbol{z}^b\|^2 + 1}} \;\; b \in \mathcal{B}^i \right\} , \tag{20}$$

which has the usual advantage to have a finite solution even if $\mathcal{L}(\underline{f}^i, f(\bar{\boldsymbol{x}}^i)$ is unbounded. Clearly, this is for the C^{3}PM what the PBM is for the CPM; in other words, (20) with an "infinitely large" $\mathcal{B}^i$ a-la (19) defines the *Elzinga-Moore-Moreau-Yosida regularization*, which is to $\Psi(\boldsymbol{x})$ what the Moreau-Yosida regularization ϕ_μ (cf. (9)) is to f. Such a function has minima where $\Psi(\boldsymbol{x}) = 0$, and therefore minimizing it is equivalent to (1) [81]. It is not surprising, then, that the *Proximal* C^{3}PM (PC^{3}PM) algorithm that minimizes it is very close to the PBM, down to the fine details of the solution of the MP (which is indeed identical save for the scaling factor $1 + \sqrt{\|\boldsymbol{z}^b\|^2 + 1}$), and whose convergence analysis proceeds in the same way. Interestingly, a variant of the approach [81, §5] exists where a second LP is solved to compute the current maximum radius of the sphere, and this is used for tuning the proximal parameter μ^i, analogously to how the PLBM solves (3) to tune l^i. This is not incidental: that *target radius method* can be interpreted as a PLBM with a specific rule to define l^i [23].

Hence, both centers-based approaches, like the PLBM, benefit from adding a second proximal stabilizing device. While double stabilization has been reported to be superior to the (singly-stabilized) PBM in some cases [26, 81], this is not yet firmly established for all relevant applications.

2.5 Approximate CPM approaches

All previous stabilization approaches are based on modifying the MP of CPM, although in some cases that is *also* solved to help tuning the stabilization parameter(s). Another take is to keep the MP unchanged, but deal with its solution differently.

A first idea is solving (3) only *approximately*, a simple way of doing this being to employ a *subgradient-type* method (a "poorman's version" of the PBM, cf §3.1), whose slow convergence and lack of effective stopping criteria mean that it is typically ran with a fixed number of iterations, reaching only an approximately optimal solution. This is actually natural enough when (1) itself is the dual of the problem one is actually interested in solving; as this is discussed in §3.3 we refrain from further delving into the subject now, pointing e.g. to [90] for details. Perhaps more interesting is the recent resurgence of ACCPM-type methods under the moniker of "*Primal-Dual Column Generation Technique*" (PDCGT) [51]. The idea is again that of stopping "way before" the optimal solution of (3) is achieved, except doing this with an IP approach rather than with a subgradient-type one. This exploits the fact that IP methods approximately follow the *central path* of the polyhedron, which starts from the AC—exactly $\boldsymbol{x}^i$ of (16), if the hat cut is included in the formulation—and goes to the $\boldsymbol{x}^i$ of (3). Hence, by construction they produce a sequence of "well centred" iterates in $\mathcal{L}^i$, except in the v-dimension (that is minimized); thus, by stopping the IP method early on—which is also convenient computationally, as IP iterations are costly—one can obtain well-centred solution "in between" the AC and the CPM iterate. Since (feasible) primal-dual IP methods (unlike, say, subgradient-type methods) allow to measure the quality of the current iterate, the stabilization parameter can just be the gap ε below which the MP computation is terminated. A large ε produces iterates close to the AC, while a small ε produces iterates close to that of the standard CPM, which can be beneficial in the "fast tail" of the CPM when $\mathcal{B}^i$ is a "good" model. As in ACCPM, however, for efficiency reasons nontrivial warm-starting strategies are needed each time the IP method is re-started after a new linearization is included in $\mathcal{B}^i$ [50].

PDCGT tracks the iterative solution, via an IP method, of the MP from the AC of $\mathcal{L}^i$ to the CPM iterate $\boldsymbol{x}^i$ of (3), and "stops somewhere in the middle". The In-Out Approach (IOA) [11] takes a similar stance in a simpler way, using the previous iterate—which doubles as a stability center $\bar{\boldsymbol{x}}^i$—in lieu of the AC. That is, the optimal solution $\boldsymbol{x}^i$ of (3) is obtained (which does not depend on $\bar{\boldsymbol{x}}^i$), and then f is computed as $\bar{\boldsymbol{x}}^{i+1} = (1 - \lambda^i)\bar{\boldsymbol{x}}^i + \lambda^i \boldsymbol{x}^i$ for some $\lambda^i \in (0, 1]$. The "In-Out" moniker derives from the fact that $(\, \bar{\boldsymbol{x}}^i \,,\, f(\bar{\boldsymbol{x}}^i) \,)$ is *inside* epi f, whereas $(\, \boldsymbol{x}^i \,,\, v^i \,)$ belongs to $\mathcal{L}^i$ which is an outer approximation, and therefore it is very likely *outside* epi f (if not, the algorithm terminates). By taking only a partial step towards $\boldsymbol{x}^i$ from $\bar{\boldsymbol{x}}^i$, the IOA tries to remain inside the epigraph. If this actually happens, then $f(\bar{\boldsymbol{x}}^{i+1}) \leq f(\bar{\boldsymbol{x}}^i) + \lambda^i \Delta^i$, where as usual $\Delta^i = v^i - f(\bar{\boldsymbol{x}}^i) < 0$; hence, a significant decrease of f is attained, a-la (7). Otherwise, the newly obtained linearization $(\, \boldsymbol{z}^i \,,\, \alpha^i \,)$ cuts away a part of $\mathcal{L}^i$. Clearly, the CPM is the special case where $\lambda^i = 1$ uniformly, and therefore convergence can be proven similarly to the CPM whenever λ^i does not become too small; the simple condition $\lambda^i \geq \underline{\lambda} > 0$ is used in [11]. As most other stabilization approaches, the IOA requires ways to dynamically tune the stabilization parameter λ^i. The recent large computational study in [83] deals with these aspects and proposes further variants where the next iterate is chosen along the *deflected* direction $(\boldsymbol{x}^i - \bar{\boldsymbol{x}}^i) + \beta^i \boldsymbol{z}^i$, where $\boldsymbol{z}^i$ is the subgradient

at $\bar{\boldsymbol{x}}^i$. The IOA method is shown to be competitive with ones using piecewise-linear penalty terms/trust regions (cf. §3.4).

It is worth remarking that the IOA is also related with the version of the PLBM where the MP of the CPM is solved, *prior* to (14), to compute the lower bound v^i out of which l^i is obtained. Indeed, the PLBM would obtain the same $\boldsymbol{x}^i$ as the IOA if it was using the (*upper*, cf. 5.2) model $\underline{f}^i$ such that $\underline{f}^i(\boldsymbol{x}) = (1-\lambda)f(\bar{\boldsymbol{x}}^i) + \lambda v^i$ if $\boldsymbol{x} = (1-\lambda)\bar{\boldsymbol{x}}^i + \lambda \boldsymbol{x}^i$, and $\underline{f}^i(\boldsymbol{x}) = \infty$ otherwise, instead as the cutting-plane one, in (14). To the best of our knowledge, this connection has never been explicitly made before.

A significant perceived benefit of both the PDCGM and the IOA for practitioners is that there is no need to modify the MP of the CPM; this may (or may not) also make them more efficient, since, say, an LP is solved instead of a QP like (8)/(14), and the LP does not have the extra bounds of (5). Of course, the approaches also share the issue of the CPM of requiring (3) to have a solution in the first place (e.g., X compact). Hybridizing them with a proximal/trust region approach, a-la [4], could solve this issue, but would do away with the benefit of working with an unsullied MP. Yet, in particular for the IP method used by PDCGM, the addition of a simple quadratic term in the objective function may not make it any significantly more difficult to solve, and conceivably even less so (cf. e.g. [17]). To the best of our knowledge, this has not been tested yet.

While the above recount summarizes many of the (simple) BM approaches in the literature, the discussion is purposely limited to the "primal" description of the problem. In many relevant applications (and, in fact, in general) the "dual" aspect is as much, if not more, important. Indeed, (1) itself can be a dual problem, whose aim is to help solving a primal one. Furthermore, the dual description is also useful to understand and implement the approaches themselves; this is the subject of the next section.

3 Duality

As everywhere in convex analysis, duality is inescapable: even if one were trying to purposely avoid it, as we did in the previous section, it would still be there. In our case, this starts from the fact that every BM solves one (or more) MP, which is a convex program and therefore it has a dual. Most often, MPs are LPs or QPs, and therefore their duals are also straightforward to compute. Doing so is actually beneficial, both because the dual may be simpler to solve, and because it reveals details of the method that can be important to understand and improve it. Also, in some applications (1) is itself the dual of the problem one is actually interested to solve, and therefore the dual of the MP (and of (1)) is related to it. This section is devoted to discussing all these issues and their main conceptual and algorithmic consequences.

3.1 Dual forms of the Master Problem

For discussing the dual forms of the MP, it is useful to introduce the *translated* model $\underline{f}_{\mathcal{B},\boldsymbol{x}}(\boldsymbol{d}) = \underline{f}_{\mathcal{B}}(\boldsymbol{x}+\boldsymbol{d}) - f(\boldsymbol{x})$ w.r.t. a point $\boldsymbol{x}$ (typically, the stability center $\bar{\boldsymbol{x}}$). This is a model of the *translated function* $f_{\bar{\boldsymbol{x}}}(\boldsymbol{d}) = f(\bar{\boldsymbol{x}}+\boldsymbol{d}) - f(\bar{\boldsymbol{x}})$ such that $f_{\bar{\boldsymbol{x}}}(\boldsymbol{0}) = 0$, with $\underline{f}_{\mathcal{B},\bar{\boldsymbol{x}}}(\boldsymbol{0}) = 0$ if any pair having $\boldsymbol{x}_b = \bar{\boldsymbol{x}}$ belongs to $\mathcal{B}$, as it usually (but not always) happens. A *displacement* (cf. (9)) $\boldsymbol{d}$ such as $\underline{f}_{\mathcal{B},\bar{\boldsymbol{x}}}(\boldsymbol{d}) < 0$ indicates a point $\boldsymbol{x} = \bar{\boldsymbol{x}} + \boldsymbol{d}$ where $\underline{f}_{\mathcal{B}}(\boldsymbol{x}) < f(\bar{\boldsymbol{x}})$, a crucial property throughout all of §2 (e.g., (7) and (20)). The effect of translation on $\mathcal{B}$ is trivial: it only amounts at replacing the α^b—the intercepts of the linearizations in the "default stability center" $\boldsymbol{0}$—with the *linearization errors*

$$\alpha^b(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}}) - [\, f(\boldsymbol{x}^b) + \langle \boldsymbol{z}^b, \bar{\boldsymbol{x}} - \boldsymbol{x}^b \rangle \,] = \alpha^b - \langle \boldsymbol{z}^b, \bar{\boldsymbol{x}} \rangle + f(\bar{\boldsymbol{x}}) \tag{21}$$

(just apply the definition to $f_{\bar{\boldsymbol{x}}}$). By convexity, $\alpha^b(\bar{\boldsymbol{x}}) \geq 0$, and

$$\boldsymbol{z}^b \in \partial_{\alpha^b(\bar{\boldsymbol{x}})} f(\bar{\boldsymbol{x}}) \tag{22}$$

where the ε-subdifferential $\partial_\varepsilon f(\bar{\boldsymbol{x}})$ contains all ε-subgradients of f at $\bar{\boldsymbol{x}}$, i.e., $\boldsymbol{z} \in \mathbb{R}^n$ such that $f(\boldsymbol{x}) \geq f(\bar{\boldsymbol{x}}) + \langle \boldsymbol{z}, \boldsymbol{x} - \bar{\boldsymbol{x}} \rangle - \varepsilon$ for all $\boldsymbol{x} \in \mathbb{R}^n$. Therefore, $\alpha^b(\bar{\boldsymbol{x}})$ is a measure of "how close" $\boldsymbol{z}^b$ is to be a subgradient of f at $\bar{\boldsymbol{x}}$. Although the definition (21) uses the original iterates $\boldsymbol{x}^b$, it is not necessary to store them to re-compute the linearization errors when $\bar{\boldsymbol{x}}$ changes to any other $\boldsymbol{x}$, since they can be updated using the *information transport property*

$$\alpha^b(\boldsymbol{x}) = \langle \boldsymbol{z}^b, \bar{\boldsymbol{x}} - \boldsymbol{x} \rangle + \alpha^b(\bar{\boldsymbol{x}}) + (\, f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}) \,) \tag{23}$$

(just write (21) for $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ and simplify out common terms). Since usually $\bar{\boldsymbol{x}}$ is clear from the context, for the sake of notational simplicity we will use α^b as much as possible. Doing so, the MP of the CPM using the translated model $\underline{f}_{\mathcal{B},\bar{\boldsymbol{x}}}$ is formally identical to (4), save that its optimal value need be increased by $f(\bar{\boldsymbol{x}})$, to account for the translation in $f/\underline{f}_{\mathcal{B}}$, to recover the original objective value. This provides a neat interpretation for its (linear) dual, i.e.,

$$[-]\ \min\left\{\ \textstyle\sum_{b\in\mathcal{B}^i}\alpha^b\theta^b\ :\ \sum_{b\in\mathcal{B}^i}\boldsymbol{z}^b\theta^b=\boldsymbol{0}\ \ ,\ \ \sum_{b\in\mathcal{B}^i}\theta^b=1\ \ ,\ \ \theta^b\geq 0\quad b\in\mathcal{B}^i\ \right\}\quad[-f(\bar{\boldsymbol{x}}^i)]\ . \tag{24}$$

Note that ordinarily (24) would be a maximization one with coefficients $-\alpha^b$ in the objective function; the change of sign reveals the problem as that of constructing $\boldsymbol{0}$ as convex combination of the $\boldsymbol{z}^b$ using "as much accurate as possible" information w.r.t. the point $\bar{\boldsymbol{x}}$ (although the latter actually changes nothing in this problem), also accounting for the fact that the offset has to be changed in sign, too. This intuitive interpretation can be stated exactly. It is crucial that the dual variables θ^b are convex combinators; since this will be quite common, we will denote by Θ the unitary simplex of appropriate dimension. The fact that $\boldsymbol{\theta}\in\Theta$ implies that

$$\textstyle\sum_{b\in\mathcal{B}}\boldsymbol{z}^b\theta^b=\boldsymbol{z}(\boldsymbol{\theta})\in\partial f_{\alpha(\boldsymbol{\theta})}(\bar{\boldsymbol{x}})\qquad\text{where}\qquad\alpha(\boldsymbol{\theta})=\sum_{b\in\mathcal{B}}\alpha^b\theta^b\ . \tag{25}$$

This can be obtained combining $\partial_\varepsilon\underline{f}_{\mathcal{B}}(\bar{\boldsymbol{x}})\subseteq\partial_\varepsilon f(\bar{\boldsymbol{x}})$ (use [57, Proposition XI.1.3.1.(vii)] together with $\underline{f}^i\leq f$ and $\underline{f}^i(\bar{\boldsymbol{x}})=f(\bar{\boldsymbol{x}})$) and

$$\partial_\varepsilon\underline{f}^i(\bar{\boldsymbol{x}})=\left\{\ \boldsymbol{z}=\textstyle\sum_{b\in\mathcal{B}^i}\boldsymbol{z}^b\theta^b\ :\ \boldsymbol{\theta}\in\Theta\ \ ,\ \ \sum_{b\in\mathcal{B}^i}\alpha^b\theta^b\leq\varepsilon\ \right\}$$

[57, Example XI.5.3]. The dual (24) can therefore be described in plain words as the problem of finding the smallest ε such that $\boldsymbol{0}\in\partial_\varepsilon\underline{f}^i(\bar{\boldsymbol{x}})$.

This interpretation carries over to the PBM: the explicit, translated form of (8) and its (quadratic) dual are, respectively

$$(\ \boldsymbol{d}^i\ ,\ v^i\ )=\operatorname{argmin}\left\{\ v+\tfrac{\mu^i}{2}\|\boldsymbol{d}\|_2^2\ :\ v\geq\langle\boldsymbol{z}^b,\boldsymbol{d}\rangle-\alpha^b\ \ b\in\mathcal{B}^i\ \right\}\ [+f(\bar{\boldsymbol{x}}^i)] \tag{26}$$

$$\boldsymbol{\theta}^i=\operatorname{argmin}\ \left\{\ \tfrac{1}{2\mu^i}\left\|\textstyle\sum_{b\in\mathcal{B}^i}\boldsymbol{z}^b\theta^b\right\|_2^2+\sum_{b\in\mathcal{B}^i}\alpha^b\theta^b\ :\ \boldsymbol{\theta}\in\Theta\ \right\}\ [-f(\bar{\boldsymbol{x}}^i)]\ . \tag{27}$$

The dual optimal solution $\boldsymbol{\theta}^i$ gives, via (25), the *aggregated linearization* ($\bar{\boldsymbol{z}}^i=\boldsymbol{z}(\boldsymbol{\theta}^i)$, $\bar{\alpha}^i=\alpha(\boldsymbol{\theta}^i)$) such that $\bar{\boldsymbol{z}}^i\in\partial_{\bar{\alpha}^i}f(\bar{\boldsymbol{x}}^i)$; the complementary slackness conditions tie that to the optimal solution of (26) as

$$\boldsymbol{d}^i=-(1/\mu^i)\bar{\boldsymbol{z}}^i\qquad,\qquad v^i=\langle\bar{\boldsymbol{z}}^i,\boldsymbol{d}^i\rangle-\bar{\alpha}^i=-(1/\mu^i)\|\bar{\boldsymbol{z}}^i\|_2^2-\bar{\alpha}^i\ . \tag{28}$$

Thus, the dual problem requires finding an ε-subgradient $\bar{\boldsymbol{z}}^i$, obtained as a convex combination of previously obtained ones, which has *both* a small norm *and* a small ε, with the relative weight of the two objective functions dictated by μ^i. In other words, (27) is the *augmented Lagrangian* of (24) w.r.t. the constraint requiring $\boldsymbol{z}^i$ to be $\boldsymbol{0}$; it is therefore not surprising, then, that the former always have a(n unique) solution, whereas the latter can be empty. Furthermore, the next iterate is then $\boldsymbol{x}^i=\bar{\boldsymbol{x}}^i+\boldsymbol{d}^i=\bar{\boldsymbol{x}}^i-(1/\mu^i)\bar{\boldsymbol{z}}^i$, i.e., is obtained by doing a step $1/\mu^i$ along the approximated subgradient; this strongly links the PBM with (approximated, deflected) subgradient-type methods, cf. §3.2 and [19].

The fact that $\bar{\boldsymbol{z}}^i\in\partial_{\bar{\alpha}^i}f(\bar{\boldsymbol{x}}^i)$ has several useful consequences. For instance, it immediately candidates it at being used as an alternative source of approximated subgradients of f to be used in (13) [85]. More importantly, however, it provides the stopping criterion of the method: $\bar{\boldsymbol{z}}^i=\boldsymbol{0}$ and $\bar{\alpha}^i=0$ imply that $\bar{\boldsymbol{x}}^i$ is optimal. In practice one therefore stops when $\|\bar{\boldsymbol{z}}^i\|$ and $\bar{\alpha}^i$ are both "small". One can either use two distinct thresholds for the two quantities, or join both in a single criterion

$$\|\bar{\boldsymbol{z}}^i\|_2^2/\bar{\mu}+\bar{\alpha}^i\leq\varepsilon\ , \tag{29}$$

where $\bar{\mu}$ is a scaling factor and ε is the final (absolute) accuracy required. This still requires to properly chose $\bar{\mu}$, but at least $\bar{\mu}$ and μ^i should be related; this means that (29) can be exploited to on-line tune μ^i, as discussed below.

However, what the above development mainly reveals is that BM have to properly balance two contrasting objectives: getting a "small" $\|\bar{\boldsymbol{z}}^i\|$, and getting a "small" $\bar{\alpha}^i$. The CPM goes all the way towards the first, which basically means completely ignoring the "quality" of the first-order information w.r.t. $\bar{\boldsymbol{x}}^i$, with the known negative practical consequences. The opposite approach is well represented by the following variant of MP [57, §XI.2.4]:

$$\min\ \left\{\ \left\|\textstyle\sum_{b\in\mathcal{B}}\boldsymbol{z}^b\theta^b\right\|_2^2\ :\ \sum_{b\in\mathcal{B}}\alpha^b\theta^b\leq\varepsilon^i\ \ ,\ \ \boldsymbol{\theta}\in\Theta\ \right\}\ . \tag{30}$$

One can see (27) as the Lagrangian relaxation of (30) having $1/\mu^i$ as Lagrangian multiplier, and therefore this would yield a BM with basically the same relationship to the PBM as the LBM has, except in the dual. In principle, for any given ε^i one could find a μ^i giving the same solution. The effect of a small ε^i in (30)—equivalently, a small μ^i in (30)—is therefore to discard all the first-order information with "large" α^b, so that the new iterate only takes into account information that is "quite accurate" at $\bar{\boldsymbol{x}}^i$. Indeed, (30) can be seen as a minor variant of the MP of *ε-descent* methods [57, Chap. IX], where $\mathcal{B}^i$ is exclusively used to construct an inner approximation of $\partial_{\varepsilon^i} f(\bar{\boldsymbol{x}}^i)$; then, (30) becomes the problem of finding the *steepest ε-descent direction* for the model, i.e., the least-norm vector in $\partial_{\varepsilon^i} \underline{f}^i(\bar{\boldsymbol{x}})$. Choosing the right value of the stabilization parameter ε^i—similarly to δ^i, μ^i, l^i, λ^i, ...—is crucial, since pure steepest descent methods have a rather bad practical behaviour even in the smooth case.

The issue with all BM is therefore to find the right value of the stabilization parameter(s) so as on one hand to include as much as possible non-local information to avoid the pitfalls of the steepest descent direction, and on the other hand not to trust the model too far beyond the region where it actually provides a reasonable depiction of the function's behaviour. For the PBM, this can be described in terms of finding the right point along the *proximal trajectory*, the family of solutions of (8) as a function of μ^i, which is a piecewise linear function, easily computed incrementally by solving a sequence of linear programs [45] or by sensitivity analysis techniques [34]. Exploring the proximal trajectory allows one to figure out how $\|\boldsymbol{z}^i\|$ and α^i change as μ^i does, and therefore can be the basis for handling μ^i.

Although similar relationship between the stabilization parameter and the locality of the used information should hold for other forms of BM, the different shape of the MP makes them less obvious to see. For reasons to become apparent in due time we postpone the discussion on the TRBM on §3.4. For the LBM, the explicit form of (14) and its dual are, respectively,

$$\boldsymbol{d}^i \in \operatorname{argmin} \left\{ \|\boldsymbol{d}\|_2^2/2 \,:\, l \geq \langle \boldsymbol{z}^b, \boldsymbol{d} \rangle - \alpha^b \;\; b \in \mathcal{B}^i \right\} \tag{31}$$

$$\boldsymbol{\theta}^i \in \operatorname{argmin} \left\{ \; \left\| \textstyle\sum_{b \in \mathcal{B}^i} \boldsymbol{z}^b \theta^b \right\|_2^2 /2 + \textstyle\sum_{b \in \mathcal{B}^i} (l + \alpha^b) \theta^b \;:\; \boldsymbol{\theta} \geq \boldsymbol{0} \right\} \tag{32}$$

(where α^b has to be intended as $\alpha^b(\bar{\boldsymbol{x}})$). Here again $\boldsymbol{d}^i = -\boldsymbol{z}(\boldsymbol{\theta}^i)$ holds as in (25), but $\boldsymbol{\theta}^i$ does not necessarily belong to Θ. Yet, the fact that necessarily $\boldsymbol{d}^i \neq \boldsymbol{0}$ implies that $\boldsymbol{\theta}^i \neq \boldsymbol{0}$ as well; thus, $\boldsymbol{\theta}^i / \langle \boldsymbol{\theta}^i, \boldsymbol{u} \rangle \in \Theta$ ($\boldsymbol{u}$ being the vector of all ones). In other words, $\boldsymbol{d}^i$ is still a scaled multiple of a convex combination of the $\boldsymbol{z}^b$, although the stepsize is no longer clearly related to the stabilization parameter. With a "small" l^i, (32) will have an incentive to only use $\boldsymbol{z}^b$ with "small" α^b (locally accurate information), whereas with a "large" l^i the role of the α^b becomes marginal. Also, note that, despite being a QP, (32) can be unbounded below as the objective function is not necessarily strictly convex (in fact, (31) can be empty). Similarly, the explicit form of the MP (15) of the DSBM and its dual are

$$(\, \boldsymbol{d}^i \,,\, v^i \,) = \qquad \min \left\{ \, v + \tfrac{\mu^i}{2} \|\boldsymbol{d}\|_2^2 \,:\, v \geq \langle \boldsymbol{z}^b, \boldsymbol{d} \rangle - \alpha^b \;\; b \in \mathcal{B}^i \;\;,\;\; l^i \geq v \right\} \tag{33}$$

$$(\, \boldsymbol{\theta}^i \,,\, \rho^i \,) \in \quad \operatorname*{argmin} \begin{array}{l} \frac{1}{2\mu^i} \left\| \sum_{b \in \mathcal{B}^i} \boldsymbol{z}^b \theta^b \right\|_2^2 + \sum_{b \in \mathcal{B}^i} \alpha^b \theta^b + l^i \rho \\ \sum_{b \in \mathcal{B}^i} \theta^b - \rho = 1 \;\;,\;\; \rho \geq 0 \;\;,\;\; \theta^b \geq 0 \qquad b \in \mathcal{B}^i \end{array} \;. \tag{34}$$

By complementary slackness, $\rho^i > 0$ implies $v^i = l^i$: in this case, (34) coincides with (32), in that $\rho^i = \langle \boldsymbol{\theta}^i, \boldsymbol{u} \rangle$ and therefore the objective function is identical, save for a constant term and the scaling factor μ^i on the quadratic term. If, instead, $v^i < l^i$ then $\rho^i = 0$ and (34) coincides with (27). Thus, ρ^i can be used to devise strategies to adjust l^i and/or μ^i [26]; this is but one of the many important uses of dual information, as discussed in the next section.

3.2 Algorithmic uses of duality

The dual concepts introduced in the previous section have many uses in the definition and analysis of BM. In particular, if $\boldsymbol{\theta}^i \in \Theta$ then the aggregated pair $(\, \bar{\boldsymbol{z}}^i = \boldsymbol{z}(\boldsymbol{\theta}^i) \,,\, \bar{\alpha}^i = \alpha(\boldsymbol{\theta}^i) \,)$ satisfies $\bar{\boldsymbol{z}}^i \in \partial_{\bar{\alpha}^i} f(\bar{\boldsymbol{x}}^i)$, and therefore can be inserted into $\mathcal{B}^i$. This is free for the PBM and the DSBM when $\rho^i = 0$; for the LBM, or the DSBM when $\rho^i > 0$, a simple scaling is needed, and an analogous technique can be used for PC^{3}PM.

The aggregated pair $(\, \bar{\boldsymbol{z}}^i \,,\, \bar{\alpha}^i \,)$ has not been obtained at any iterate $\boldsymbol{x}^i$, but this is not an issue; $\bar{\alpha}^i = \bar{\alpha}^i(\bar{\boldsymbol{x}}^i)$ can be updated via (23) when $\bar{\boldsymbol{x}}^i$ changes as all the other ones in $\mathcal{B}^i$. The important result is that $(\, \bar{\boldsymbol{z}}^i \,,\, \bar{\alpha}^i \,)$ can actually *substitute* all other information: if one were to set $\mathcal{B}^{i+1} = \{ (\, \bar{\boldsymbol{z}}^i \,,\, \bar{\alpha}^i \,) \}$, then $(\, \boldsymbol{d}^{i+1} \,,\, v^{i+1} \,) = (\, \boldsymbol{d}^i \,,\, v^i \,)$ in (26). Of course, one does not really want the solution to remain the

same, in particular if a NS is being performed; this is not so because of the new information $(\boldsymbol{z}^i, \alpha^i)$ computed by evaluating $f(\boldsymbol{x}^i)$. It is easy to prove that even if one takes the minimal stance $\mathcal{B}^{i+1} = \bar{\mathcal{B}}^i = \{(\bar{\boldsymbol{z}}^i, \bar{\alpha}^i), (\boldsymbol{z}^i, \alpha^i)\}$—called the *poorman's bundle*—the PBM is still convergent; that is, an infinite sequence of consecutive NS will result in $\|\bar{\boldsymbol{z}}^i\| \to 0$ and $\bar{\alpha}^i \to 0$. The proof is simple and instructive enough to be worth reporting: it is based on the fact that (27) with $\bar{\mathcal{B}}^i$ is the simple problem

$$\min \left\{ h^i(\theta) = \tfrac{1}{2\mu^i}\|\theta\bar{\boldsymbol{z}}^i + (1-\theta)\boldsymbol{z}^i\|_2^2 + \theta\bar{\alpha}^i + (1-\theta)\alpha^i \;:\; \theta \in [0,1] \right\} , \tag{35}$$

whose optimal solution has the following closed-form expression:

$$\theta_*^i \;=\; \min\left\{1\,,\; \max\left\{0\,,\; \tfrac{\alpha^i - \bar{\alpha}^i - \langle \boldsymbol{z}^i, \bar{\boldsymbol{z}}^i - \boldsymbol{z}^i\rangle/\mu^i}{\|\bar{\boldsymbol{z}}^i - \boldsymbol{z}^i\|_2^2/\mu^i}\right\}\right\} . \tag{36}$$

Since $h^i(1)$ is the optimal value of (27) at iteration i, one only has to show that $h^i(1) - h^i(\theta_*^i)$ decreases enough. This hinges on the fact that (7) *not* holding can be rewritten, by means of some simple algebra (cf. (28))

$$\Delta f^i > -mv^i = -m\big(-(1/\mu^i)\|\bar{\boldsymbol{z}}^i\|_2^2 - \bar{\alpha}^i\big) \geq -mh^i(1) , \tag{37}$$

from which it is easy to derive

$$h^i(1) - h^i(\theta_*^i) \geq \tfrac{(1-m)h^i(1)}{2} \min\left\{1\,,\; \tfrac{(1-m)h^i(1)}{\|\bar{\boldsymbol{z}}^i - \boldsymbol{z}^i\|_2^2/\mu^i}\right\} . \tag{38}$$

By (38), the optimal value of (27) is decreasing, and it must necessarily converge to zero during an infinite sequence of consecutive NS (at least if μ^i is not dramatically mishandled, e.g. just kept bounded away from 0).

Thus, the PBM is convergent provided that $(\bar{\boldsymbol{z}}^i, \bar{\alpha}^i)$ is still a feasible solution of (34) (in the $(\boldsymbol{z}, \alpha)$-space) at iteration $i+1$, and, of course, $(\boldsymbol{z}^i, \alpha^i) \in \mathcal{B}^{i+1}$. This immediately suggests the two standard forms of *bundle management*: i) ensure that $(\bar{\boldsymbol{z}}^i, \bar{\alpha}^i) \in \mathcal{B}^{i+1}$ (*compression*), ii) ensure that $b \in \mathcal{B}^{i+1}$ for all the $b \in \mathcal{B}^i$ such that $\theta^{i,b} > 0$ (*selection*). Note that, by Carathéodory's theorem, there always exist a $\boldsymbol{\theta}^i$ with at most $n+1$ positive variables; hence, both strategies yield a finite bound over the size of $\mathcal{B}$, a significant advantage—at least in theory—over the non-stabilized CPM.

Not unexpectedly, the practical side of bundle management is considerably more nuanced. For once, (38) only refers to the "tail" of the algorithm, where $\bar{\boldsymbol{x}}^i$ has reached (very close to) some optimal $\boldsymbol{x}_*$ and the PBM "only" have to *prove* this by driving both $\|\bar{\boldsymbol{z}}^i\|$ and $\bar{\alpha}^i$ to 0. This all but ignores the "cruise" phase where $\boldsymbol{x}^i$ is closing in to $\boldsymbol{x}_*$. For that, (7) would imply a reasonably fast convergence *in the number of SS*, with of course the issue of how many NS occur between two consecutive SS. Even ignoring this, the rate of convergence implied by (38) is sublinear, i.e., rather slow. This is one of the main reasons why iteration complexity of the PBM is $O(1/\varepsilon^3)$ [2, 61], even worse than the $O(1/\varepsilon^2)$ that any black-box algorithm of this type necessarily has to have. This is so bad a convergence rate as to make it completely impractical to obtain anything more than moderately accurate solutions. Indeed, the PBM with "extreme" aggregation $\mathcal{B}^{i+1} = \bar{\mathcal{B}}^i$ is a minor variant of a *deflected* subgradient-type method [22]; in particular, it is closely related [7] with the so-called *Volume Algorithm* [8], that had spurred considerable interest in combinatorial optimization circles at the turn of the millenium. It had actually been known already [1] that these subgradient-type methods have—in theory—a working stopping criterion, which is important in some applications. However, (38) reveals how the advantage is only theoretical: in practice, convergence of subgradient methods is so slow that the only feasible stopping criterion is a limit on the number of iterations. Although they can still be attractive in some applications, this is only true under very mild requirements on the required accuracy (say, 1e-3 to 1e-4 relative) [41].

It is revealing to contrast this behaviour with that of the CPM as numerically illustrated in §2.1. There, although the algorithm has an erratic behaviour in the "cruise" phase (apparently failing to exhibit any convergence at all), the "tail" of the process is pleasingly fast. This is due to the fact that once enough information is accrued in $\mathcal{B}$ to make $\underline{f}_{\mathcal{B}}$ a good enough model at some optimal solution, the algorithm can efficiently close in to that. Such accumulation of information in $\mathcal{B}$ is essentially destroyed by extreme aggregation $\mathcal{B}^{i+1} = \bar{\mathcal{B}}^i$: although the process remains generally convergent, the speed can be as abysmal in practice as (38) predicts. In other words, extreme aggregation can hurt a BM precisely in what could otherwise be a strong point of its. Similarly, discarding a pair $(\boldsymbol{z}^b, \alpha^b)$ as soon as $\theta^{i,b} = 0$ may considerably hurt performances; more appropriate (heuristic) rules are to discard it after that the multiplier has been zero for some (say, 20) consecutive iterations. In some tests, the "fast tail phase" has proven rather delicate, being impaired by even mildly aggressive selection rules or by imposing even seemingly loose limits on the maximum size of $\mathcal{B}$ [43]. Hence, at least in some applications it is better

to shoulder the substantial burden of solving MP with a large $\mathcal{B}$ than trying to keep the latter small, as any reduction in MP cost is largely outweighed by the corresponding decrease of convergence speed.

Unfortunately, all these aspects are only characterized experimentally; all convergence arguments—and efficiency estimates—on PBM hinge on extreme aggregation. The complexity estimate can actually be improved to—the still sublinear—$O(\log(1/\varepsilon)(1/\varepsilon))$ with further assumptions on f (in particular, strong coercivity at the unique optimum) [28], but still the same bound holds for $\bar{\mathcal{B}}^i$ and for any arbitrarily large $\mathcal{B}^i$; hence, the theoretical worst-case analysis seems unable to capture some important aspects of the practical behaviour of BM, (fortunately) substantially underestimating convergence speed. This is not helped by the fact that convergence arguments, as discussed in §2.1, deal with the sequence of SS and with sub-sequences of consecutive NS between two SS as two loosely related processes; after a SS is declared the algorithm can basically be restarted from scratch, as the arguments allow to completely change $\mathcal{B}$ then. One recent effort to devise a convergence analysis of the PBM as an unique process is based on (in principle) avoiding the dichotomic distinction between SS and NS [2]. This hinges on the introduction of the—apparently weird—*merit function* $\zeta_\mu(x) = 2f(x) - \phi_\mu(x)$, with ϕ_μ the Moreau–Yosida regularization (9). The only nice properties of ζ_μ are that $\zeta_\mu \geq f$ and $\zeta_\mu(x) = f(x) \iff x$ is optimal for (1); otherwise, the function is nondifferentiable and nonconvex. However, its upper approximation $\zeta_{\mathcal{B},\mu} \geq \zeta_\mu$ obtained by replacing f with $\underline{f}_{\mathcal{B}}$ in (8) is *precisely* computed by solving (27), *comprised* the constant term "$-f(\bar{\boldsymbol{x}})$" that is usually ignored in the analysis of the PBM. Once $\boldsymbol{x}^i$ is produced by the MP and $f(\boldsymbol{x}^i)$ and $\boldsymbol{z}^i$ are computed by the oracle, it is possible to define the problem of minimizing $\zeta_{\mathcal{B},\mu}(\boldsymbol{x})$ for $\boldsymbol{x} \in [\bar{\boldsymbol{x}}^i, \boldsymbol{x}^i]$. Actually, doing so would require knowing the value of f at all points of the interval; this can be replaced by an *upper model* of f on the interval, typically $\lambda f(\bar{\boldsymbol{x}}^i) + (1-\lambda) f(\boldsymbol{x}^i)$ for $\boldsymbol{x} = \boldsymbol{x}(\lambda) = \lambda \bar{\boldsymbol{x}}^i + (1-\lambda)\boldsymbol{x}^i$. This allows to define a further upper approximation of $\zeta_{\mathcal{B},\mu}$, and $\bar{\boldsymbol{x}}^{i+1}$ can be easily chosen as the minima of this function on the interval $[\bar{\boldsymbol{x}}^i, \boldsymbol{x}^i]$; doing so one can prove that eventually $\zeta_\mu(\bar{\boldsymbol{x}}^i) - f(\bar{\boldsymbol{x}}^i) \to 0$, i.e., global convergence. This is potentially interesting in that $\bar{\boldsymbol{x}}^{i+1}$ can be chosen "in between" $\bar{\boldsymbol{x}}^i$ and $\boldsymbol{x}^i$, thereby generalizing the PBM at least insomuch as $f(\bar{\boldsymbol{x}}^{i+1}) > f(\bar{\boldsymbol{x}}^i)$ can happen. Unfortunately, the approach—at least with the natural upper model—turns out to actually only do either SS or NS. Also, the efficiency analysis still uses arguments very close to (36), and therefore it does not seem of being any better able of properly evaluating the effect of information accrual. Besides, the practical efficiency of the method does not seem to be much different from that of the original PBM.

All in all, it can be argued that the currently available convergence and efficiency analyses fail to properly capture some aspects of the BM that are important in practice. Yet, the dual viewpoint is crucial for the understanding and the implementation of BM; this is even more so in the case where (1) is itself a dual problem, as discussed next.

3.3 Duality in the original problem

One important motivation for (1) is the case where

$$f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{b} \rangle + \max \left\{ \langle \boldsymbol{c} - \boldsymbol{x}A, \boldsymbol{u} \rangle \ : \ \boldsymbol{u} \in U \right\} \ , \tag{39}$$

i.e., f is the Lagrangian function of the problem

$$\max \left\{ \langle \boldsymbol{c}, \boldsymbol{u} \rangle \ : \ A\boldsymbol{u} = \boldsymbol{b} \ , \ \boldsymbol{u} \in U \right\} \ , \tag{40}$$

w.r.t. the explicit constraints. Customarily U is assumed compact, so that f is finite-valued; this is mainly to save on details, with extensions discussed in §4.3. Similarly, linearity of objective function and constraints can be relaxed with most of the results carrying over, albeit at the cost of considerably more cumbersome notation [72].

Evaluating f at some iterate $\boldsymbol{x}^i$ requires solving the *Lagrangian relaxation* (39) of (40). Any of its *optimal solutions* $\boldsymbol{u}^i$ gives $f(\boldsymbol{x}^i) = \langle \boldsymbol{c} - \boldsymbol{x}^i A, \boldsymbol{u}^i \rangle + \langle \boldsymbol{x}^i, \boldsymbol{b} \rangle$ and $\boldsymbol{z}^i = \boldsymbol{b} - A\boldsymbol{u}^i$; note that this yields $\alpha^i = \langle \boldsymbol{z}^i, \boldsymbol{x}^i \rangle - f(\boldsymbol{x}^i) = -\langle \boldsymbol{c}, \boldsymbol{u}^i \rangle$, a suggestive enough result. Indeed, the dual (24) of the MP of the CPM then (heavily exploiting linearity) becomes

$$\max \left\{ c\big(\textstyle\sum_{b \in \mathcal{B}^i} \boldsymbol{u}^b \theta^b \big) \ : \ A\big(\textstyle\sum_{b \in \mathcal{B}^i} \boldsymbol{u}^b \theta^b \big) = \boldsymbol{b} \ , \ \boldsymbol{\theta} \in \Theta \right\} \quad . \tag{41}$$

Hence, one is in fact considering the convex set $U_{\mathcal{B}} = \mathrm{conv}(\{ \boldsymbol{u}^b \ : \ b \in \mathcal{B} \})$, which would be an inner approximation of U if the latter were convex, and is solving (40) with U replaced by $U_{\mathcal{B}}$. Clearly, "with

an infinitely large $\mathcal{B}$" one would be solving the *convexified relaxation* of (40)

$$\max\{\ \langle \boldsymbol{c},\boldsymbol{u}\rangle\ :\ A\boldsymbol{u}=\boldsymbol{b}\ ,\ \ \boldsymbol{u}\in \mathrm{conv}(U)\ \}\ , \tag{42}$$

equivalent to (40) if U is convex, and in some sense its best possible convex relaxation otherwise. Then, (41) is the *inner approximation* (a restriction) of (42) corresponding to the finite subset of solutions collected so far. The optimal value of (41) is thus a lower bound on that of (42), just as that of (3) is a lower bound on f_*; indeed, the *Lagrangian Dual* (LD) (1) of (40) is equivalent to its convexified relaxation (42), a celebrated result [72] with many useful consequences [37]. This allows to give interesting interpretations to the results of §3.1, starting with the fact that the linearization error (21) becomes $\alpha^b(\bar{\boldsymbol{x}}) = \langle \boldsymbol{c}-\bar{\boldsymbol{x}}A, \boldsymbol{u}(\bar{\boldsymbol{x}})\rangle - \langle \boldsymbol{c}-\bar{\boldsymbol{x}}A, \boldsymbol{u}^b\rangle$, where $\boldsymbol{u}(\bar{\boldsymbol{x}})$ is (any one of) the optimal solution of (39) with $\boldsymbol{x}=\bar{\boldsymbol{x}}$; basically, how much sub-optimal is the solution $\boldsymbol{u}^b$ w.r.t. the optimal one $\boldsymbol{u}(\bar{\boldsymbol{x}})$ with the *Lagrangian costs* (sometimes called *reduced costs*) $\boldsymbol{c}-\bar{\boldsymbol{x}}A$ corresponding to the current point $\bar{\boldsymbol{x}}$.

Hence, the LD (1) of (42)/(40)—which is the same, as the LD cannot distinguish a problem from its convexified relaxation—provides a way to solve (42) by iteratively accumulating solutions $\boldsymbol{u}^i \in U$ and explicitly constructing (the relevant part of) its feasible region. If, say, U is a finite set, then $conv(U)$ is a polyhedron and only the finite set of its extreme points is required to fully represent it; hence, the LP (41) with a (possibly, very) large $\mathcal{B}$ is actually equivalent to (42). This is known as the *Dantzig-Wolfe reformulation* of (42), and it is well-known that solving the LD by the CPM is equivalent to solving (42) by the *Dantzig-Wolfe decomposition algorithm* [37]. The Dantzig-Wolfe reformulation has "few" $(n+1)$ constraints, but in principle exponentially many variables (columns in the LP); thus the Dantzig-Wolfe decomposition algorithm is also referred to as *Column Generation* (although the latter concept is in some sense slightly more general) [27]. Stabilizing the CPM is therefore also known as stabilizing the Column Generation [11, 12]. For instance, for the PBM one can re-write (27) as

$$\max\{\ \textstyle\sum_{b\in\mathcal{B}^i}(\boldsymbol{c}\boldsymbol{u}^b)\theta^b + \langle \bar{\boldsymbol{x}},\boldsymbol{z}\rangle - \frac{1}{2\mu^i}||\,\boldsymbol{z}\,||_2^2\ :\ A\left(\textstyle\sum_{b\in\mathcal{B}^i}\boldsymbol{u}^b\theta^b\right)-\boldsymbol{b}=\boldsymbol{z}\ \ ,\ \ \boldsymbol{\theta}\in\Theta\ \}\ ,$$

or, even more tellingly, as

$$\max\{\ \langle \boldsymbol{c},\boldsymbol{u}\rangle + \langle \bar{\boldsymbol{x}},\boldsymbol{b}-A\boldsymbol{u}\rangle - \tfrac{1}{2\mu^i}||\,A\boldsymbol{u}-\boldsymbol{b}\,||_2^2\ :\ \boldsymbol{u}\in U^i\ \}\ . \tag{43}$$

(with $U^i = U_{\mathcal{B}^i}$). Thus, the PBM can be read from the viewpoint of (42) as an *augmented Lagrangian* combined with an *inner linearization* approach where U is substituted by its approximation U^i. The aggregated pair $(\,\bar{\boldsymbol{z}}^i\,,\,\bar{\alpha}^i\,)$ is then associated with the point

$$\bar{\boldsymbol{u}}^i = \boldsymbol{u}(\boldsymbol{\theta}^i)\in\mathrm{conv}(U)\quad\text{with}\quad \boldsymbol{u}(\boldsymbol{\theta}) = \textstyle\sum_{b\in\mathcal{B}}\boldsymbol{u}^b\theta^b \tag{44}$$

by $\bar{\boldsymbol{z}}^i = \boldsymbol{b}-A\bar{\boldsymbol{u}}^i$ and $\bar{\alpha}^i = \langle \boldsymbol{c}-\bar{\boldsymbol{x}}A, \boldsymbol{u}(\bar{\boldsymbol{x}})\rangle - \langle \boldsymbol{c}-\bar{\boldsymbol{x}}A, \bar{\boldsymbol{u}}^i\rangle$; convergence of the PBM can be read as the fact that $\{\,\bar{\boldsymbol{u}}^i\}\to\boldsymbol{u}_*$, with the latter optimal to (42) (an easy but instructive connection to formally prove).

It is, however, useful to delve a bit further into the equivalence between (1) and (42)—with the f of (39)—as doing so requires to introduce useful concepts, primarily the *Fenchel's conjugate* $f^*(\boldsymbol{z}) = \sup_{\boldsymbol{x}}\{\,\langle \boldsymbol{z},\boldsymbol{x}\rangle - f(\boldsymbol{x})\,\}$ of f. The function f^* is convex by definition, even if f is not, and closed under very mild assumptions on f. The bi-conjugate f^{**} is the (closed) *convex envelope* of f, i.e., the smallest (in set-inclusion sense) closed convex function g such that $\mathrm{epi}(g)\supseteq\mathrm{epi}(f)$; clearly, if f is closed convex then $f^{**}=f$. Geometrically, f^* characterizes all the affine functions supporting $\mathrm{epi}(f)$, i.e., basically its (approximate) subgradients; indeed, a fundamental property of f^* is

$$\boldsymbol{z}\in\partial_\varepsilon f(\boldsymbol{x}) \Longleftrightarrow \boldsymbol{x}\in\partial_\varepsilon f^*(\boldsymbol{z}) \Longleftrightarrow f(\boldsymbol{x})+f^*(\boldsymbol{z})\le\langle \boldsymbol{z},\boldsymbol{x}\rangle+\varepsilon \tag{45}$$

for each $\varepsilon\ge 0$ [57, Proposition XI.1.2.1]. Coupled with *Fenchel's inequality* $\langle \boldsymbol{z},\boldsymbol{x}\rangle\le f(\boldsymbol{x})+f^*(\boldsymbol{z})$ for all $\boldsymbol{z}$, $\boldsymbol{x}$, this gives $\langle \boldsymbol{z},\boldsymbol{x}\rangle = f(\boldsymbol{x})+f^*(\boldsymbol{z}) \Longleftrightarrow \boldsymbol{z}\in\partial f(\boldsymbol{x}) \Longleftrightarrow \boldsymbol{x}\in\partial f^*(\boldsymbol{z})$; the immediate consequence is that $\alpha^b = \langle \boldsymbol{z}^b,\boldsymbol{x}^b\rangle - f(\boldsymbol{x}^b) = f^*(\boldsymbol{z}^b)$ $(=-\langle \boldsymbol{c},\boldsymbol{u}^b\rangle$ for (39)). Also, since $(f(\cdot+\bar{\boldsymbol{x}}))^*(\boldsymbol{z}) = f^*(\boldsymbol{z})-\langle \boldsymbol{z},\bar{\boldsymbol{x}}\rangle$ and $(f(\cdot)-v)^*(\boldsymbol{z}) = f^*(\boldsymbol{z})+v$, one has for the translated function $f_{\bar{\boldsymbol{x}}}(\boldsymbol{d}) = f(\bar{\boldsymbol{x}}+\boldsymbol{d})-f(\bar{\boldsymbol{x}})$ that $f^*_{\bar{\boldsymbol{x}}}(\boldsymbol{z}) = f^*(\boldsymbol{z})-\langle \boldsymbol{z},\boldsymbol{x}\rangle+f(\bar{\boldsymbol{x}})$ $(\ge 0$ by Fenchel's inequality). Hence, $\alpha^b(\bar{\boldsymbol{x}}) = \alpha^b - \langle \boldsymbol{z}^b,\bar{\boldsymbol{x}}\rangle + f(\bar{\boldsymbol{x}}) = f^*_{\bar{\boldsymbol{x}}}(\boldsymbol{z}^b)$ (cf. (21)). Thus, clearly all dual problems of §3.1 are dealing with $f^*/f^*_{\bar{\boldsymbol{x}}}$, but when f is (39) they are also dealing with (42). The link is made explicit by the (opposite of the) *value function* of (42)

$$\begin{aligned}
\nu(\boldsymbol{z}) &= -\max\{\ \langle \boldsymbol{c},\boldsymbol{u}\rangle\ :\ \boldsymbol{b}-A\boldsymbol{u}=\boldsymbol{z}\ ,\ \boldsymbol{u}\in\mathrm{conv}(U)\ \}\ , \\
\text{in fact}\quad \nu^*(\boldsymbol{x}) &= \max\ \{\ \langle \boldsymbol{z},\boldsymbol{x}\rangle + \max\{\ \langle \boldsymbol{c},\boldsymbol{u}\rangle\ :\ \boldsymbol{b}-A\boldsymbol{u}=\boldsymbol{z}\ ,\ \boldsymbol{u}\in\mathrm{conv}(U)\ \}\ \} \\
&= \max\{\ \langle \boldsymbol{c},\boldsymbol{u}\rangle + \langle \boldsymbol{x},\boldsymbol{b}-A\boldsymbol{u}\rangle\ :\ \boldsymbol{u}\in\mathrm{conv}(U)\ \}\ =\ f(\boldsymbol{x})\ .
\end{aligned} \tag{46}$$

With the obvious $f^*(\boldsymbol{0}) = -f_*$, this confirms that the LD (1) of (40) is equivalent to its convexified

relaxation (42): $\nu(\mathbf{0}) = -f_*$, with the change in sign only due to the insistence on minimization typical of convex optimization. Linearity of the objective function is by no means a crucial ingredient: with a generic objective function $c(\boldsymbol{u})$ in (40), the LD (1) is equivalent to $\max\{\, \tilde{c}(\boldsymbol{u}) \;:\; A\boldsymbol{u} = \boldsymbol{b} \,\}$, where $\tilde{c} = (c \;+\; \imath_U)^{**}$. The result easily extends to inequality constraints $A\boldsymbol{u} \leq \boldsymbol{b}$, yielding sign constraints $\boldsymbol{x} \geq 0$ in (1); the generic nonlinear case $A(\boldsymbol{u}) \leq b$ requires considerably more complex notation, even in the convex case, although the results are in the same vein [72].

Thus, the conjugate f^* allows to express in a general way primal/dual relationships that would seem to be specific of the Lagrangian case (39). In particular, one can consider the (apparently weird) problem

$$\min\{\, f^*(\boldsymbol{z}) \;:\; \boldsymbol{z} = \mathbf{0} \,\} \tag{47}$$

as the dual of (1). This is quite a reasonable dual: its optimal value is $(-)\; f_*$, and it deals with dual objects, as $\boldsymbol{z} \in \operatorname{dom} f^*$ if and only if $\boldsymbol{z} \in \partial f(\boldsymbol{x})$ for some point $\boldsymbol{x}$ (cf. (45)). Furthermore, the Lagrangian relaxation of (47) w.r.t. the constraints "$\boldsymbol{z} = \mathbf{0}$", using $\bar{\boldsymbol{x}}$ as Lagrangian multipliers, is

$$\inf\{\, f^*(\boldsymbol{z}) - \langle \boldsymbol{z}, \bar{\boldsymbol{x}} \rangle \,\} = (f^*)^*(\bar{\boldsymbol{x}}) = f(\bar{\boldsymbol{x}}) \;\; . \tag{48}$$

Thus, the minimization in (48) is the equivalent to the maximization in (39): a Lagrangian relaxation that has to be solved to find the optimum $\boldsymbol{z}$ (respectively $\boldsymbol{u}$), which is (provides) the subgradient. All interpretation of, say, the PBM as an inner linearization approach, where the computation of $f(\boldsymbol{x}^i)$ provides a new point $\boldsymbol{u}^i$ that enlarges $U_{\mathcal{B}}$, can be recast in terms of generating an inner approximation of $\operatorname{epi} f^*$, without a need for any special structure in f. This is described in some detail in the next section, in which conjugacy arguments—in particular *Fenchel's duality*—are used to devise more general stabilization devices than the proximal and trust region ones. Yet, it is clear that the dual interpretation of BM is particularly useful for their applications to Lagrangian optimization (cf. [5, 6, 11, 12, 15, 16, 20, 31, 33, 38–40, 42, 43, 46, 55, 67, 71, 83, 89, 91, 98] among the many others), because then generated dual information has a direct and crucial algorithmic use (e.g., [8, 9, 14, 21, 30, 32, 37, 44]).

3.4 Generalized stabilization

As the previous section showed, devising and analysing BM requires—or at least significantly benefits from—considering the dual aspects of all involved concepts, starting from the MP. This would seem to make it harder to use less simple stabilizing terms, like trust-region constraints in any norm that is not L_1, L_2 or L_∞ or penalty functions that are not either piecewise-linear or convex quadratic, just because then the dual of the MP cannot be obtained with familiar LP or QP duality. Yet, it is intuitively clear that these should in principle work as well (if not better) than the simple ones. Furthermore, there can be good reasons for wanting to use different stabilizing terms, which requires being able to express dual relationships beyond LP and QP. Being this a convex setting Lagrangian duality would seem to be the natural recourse, but its max/min form is more cumbersome than closed-form duals with only dual variables. The alternative is *Fenchel's Duality*, mirably expressed by

$$\inf_{\boldsymbol{x}} \{\; f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}) \;\} = -\inf_{\boldsymbol{z}} \{\; f_1^*(\boldsymbol{z}) + f_2^*(-\boldsymbol{z}) \;\} \quad , \tag{49}$$

which holds under mild assumptions (f_1 and f_2 closed convex and the intersection of their domains nonempty). Note the "$-\boldsymbol{z}$", which comes from the standard form of the conjugate of a sum

$$(\, f_1(\cdot) + f_2(\cdot)\,)^*(\mathbf{0}) = \inf_{\boldsymbol{z}_1, \boldsymbol{z}_2} \{\; f_1^*(\boldsymbol{z}_1) + f_2^*(-\boldsymbol{z}_2) \;:\; \boldsymbol{z}_1 + \boldsymbol{z}_2 = \mathbf{0} \;\}$$

and that could not be noticed in §3.1 because the stabilizing terms are radially symmetric ($\|\boldsymbol{z}\| = \|-\boldsymbol{z}\|$). Thus, one may consider the Generalized BM (GBM) with a *generalized stabilization term* D_μ [36], i.e., the MP

$$\boldsymbol{d}^i \in \operatorname{argmin} \{\; \underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}) + D_{\mu^i}(\boldsymbol{d}) \;\} \quad , \tag{50}$$

and immediately derive its (Fenchel's) Dual

$$\bar{\boldsymbol{z}}^i \in \operatorname{argmin} \{\; (\underline{f}^i)^*(\boldsymbol{z}) + \langle \boldsymbol{z}, \bar{\boldsymbol{x}} \rangle + D^*_{\mu^i}(-\boldsymbol{z}) \;\} \;\; . \tag{51}$$

This becomes more familiar when using $\underline{f}_{\mathcal{B}} = \check{f}_{\mathcal{B}}$, as for (2) one has

$$\check{f}^*_{\mathcal{B}}(\boldsymbol{z}) \quad = \quad \min \{\; \textstyle\sum_{b \in \mathcal{B}} \alpha^b \theta^b \;:\; \sum_{b \in \mathcal{B}} \boldsymbol{z}^b \theta^b = \boldsymbol{z} \;,\; \boldsymbol{\theta} \in \Theta \;\} \tag{52}$$

which, with (say) $D_\mu(\boldsymbol{d}) = \mu\|\boldsymbol{d}\|_2^2/2 \equiv D_\mu^*(\boldsymbol{z}) = \|\boldsymbol{z}\|_2^2/(2\mu)$ immediately reproduces (27). For the Lagrangian case (39), (51) becomes (cf. (43))

$$\bar{\boldsymbol{u}}^i \in \operatorname{argmin}\left\{ \langle \boldsymbol{c}, \boldsymbol{u}\rangle + \langle \bar{\boldsymbol{x}}, \boldsymbol{b} - A\boldsymbol{u}\rangle - D_{\mu^i}^*(\, A\boldsymbol{u} - \boldsymbol{b}\,) \;:\; \boldsymbol{u} \in U^i \right\}$$

(again, note the change of sign in $\boldsymbol{z} = \boldsymbol{b} - A\boldsymbol{u}$), or, in "explicit form"

$$\min\left\{ \textstyle\sum_{b\in\mathcal{B}^i} (\boldsymbol{c}\boldsymbol{u}^b)\theta^b + \langle \bar{\boldsymbol{x}}, \boldsymbol{z}\rangle + D_{\mu^i}^*(\,-\boldsymbol{z}\,) \;:\; A\left(\textstyle\sum_{b\in\mathcal{B}^i} \boldsymbol{u}^b\theta^b\right) - \boldsymbol{b} = \boldsymbol{z} \;\;,\;\; \boldsymbol{\theta} \in \Theta \right\} \;;$$

a *generalized Augmented Lagrangian* of (42), with D_μ^* in the second-order term. Conjugacy arguments allow to derive primal-dual relationships that do not depend on the choice of D_μ (or $\underline{f}_{\mathcal{B}}$, for that matter), such as

$$\begin{aligned}
-\bar{\boldsymbol{z}}^i \in \partial D_{\mu^i}(\boldsymbol{d}^i) \quad &\text{and} \quad \boldsymbol{d}^i \in \partial D_{\mu^i}^*(-\bar{\boldsymbol{z}}^i) \\
\bar{\boldsymbol{z}}^i \in \partial \underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}^i) \quad &\text{and} \quad \bar{\boldsymbol{x}} + \boldsymbol{d}^i \in \partial(\underline{f}^i)^*(\bar{\boldsymbol{z}}^i) \\
\underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}^i) + (\underline{f}^i)^*(\bar{\boldsymbol{z}}^i) = \langle \bar{\boldsymbol{z}}^i, \bar{\boldsymbol{x}} + \boldsymbol{d}^i\rangle \quad &\text{and} \quad D_{\mu^i}(\boldsymbol{d}^i) + D_{\mu^i}^*(-\bar{\boldsymbol{z}}^i) = -\langle \bar{\boldsymbol{z}}^i, \boldsymbol{d}^i\rangle \;\;.
\end{aligned}$$

These generalize most of the relationships that are needed to prove convergence of a PRB; for instance, one can prove the suggestive

$$\Delta^i = \underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}^i) - f(\bar{\boldsymbol{x}}) = (\underline{f}^i)^*(\bar{\boldsymbol{z}}^i) - f^*(\boldsymbol{z}^i) + \langle \boldsymbol{z}^i - \bar{\boldsymbol{z}}^i, \bar{\boldsymbol{x}} + \boldsymbol{d}^i\rangle$$

(with $\boldsymbol{z}^i \in \partial f(\boldsymbol{x}^i = \bar{\boldsymbol{x}} + \boldsymbol{d}^i)$), which gives a dual interpretation to the SS condition (7). Note, however, that not all the relevant relationships of the PBM generalize; most notably, $\boldsymbol{d}^i = -\bar{\boldsymbol{z}}^i/\mu^i$ is *not* true in general, which prevents using some important arguments (basically, a GBM is not necessarily a subgradient-type method in the same was as the PBM is). Yet, convergence can still be proven, provided of course that D_μ has the right properties; those proposed in [36] are nicely symmetric w.r.t. the conjugacy operation:

1. $\forall \mu > 0$, $D_\mu(\boldsymbol{0}) = 0$ and $\boldsymbol{0} \in \partial D_\mu(\boldsymbol{0}) \equiv D_\mu^*(\boldsymbol{0}) = 0$ and $\boldsymbol{0} \in \partial D_\mu^*(\boldsymbol{0})$;
2. $\forall \mu > 0$ and $\varepsilon > 0$, $\operatorname{lev}(D_\mu, \varepsilon)$ is *compact* and $0 \in \operatorname{int}\operatorname{lev}(D_\mu, \varepsilon) \equiv$ $\operatorname{lev}(D_\mu^*, \varepsilon)$ is *compact* and $0 \in \operatorname{int}\operatorname{lev}(D_\mu^*, \varepsilon)$;
3. $\forall \mu' \geq \mu > 0$, $D_\mu \leq D_{\mu'} \equiv D_\mu^* \geq D_{\mu'}^*$;
4. $\lim_{\mu\to 0} D_\mu(\boldsymbol{d}) = 0 \;\forall \boldsymbol{d} \equiv \forall \varepsilon > 0$, $\lim_{\mu\to 0} \inf\{\, D_\mu^*(\boldsymbol{z}) \;:\; \|z\| \geq \varepsilon \,\} = +\infty$.

That is, both D_μ and D_μ^* must be non-negative and have bounded level sets with nonempty interior. Of course, some properties are only symmetric insomuch as it is allowed by conjugacy: D_μ has to be *increasing* in μ and converge pointwise to the constant zero function as $\mu \to 0$, which means that D_μ^* has to be *decreasing* in μ and converge "uniformly" to the indicator function of $\{\,\boldsymbol{0}\,\}$ as $\mu \to 0$. That is, D_μ becomes less and less stabilizing as $\mu \to 0$: (50) becomes more and more like (3), hence (51) becomes more and more like (24) ($\bar{\boldsymbol{z}}^i$ is constrained to remain closer and closer to $\boldsymbol{0}$). It is easy to see that these properties are respected both by the proximal and by the trust-region stabilization; in particular $D_\mu(\boldsymbol{d}) = \imath_{\{\,\boldsymbol{d}\,:\,\|\boldsymbol{d}\|_1 \leq 1/\mu\,\}} \equiv D_\mu^*(\boldsymbol{z}) = \|\boldsymbol{z}\|_\infty/\mu$ and $D_\mu(\boldsymbol{d}) = \imath_{\{\,\boldsymbol{d}\,:\,\|\boldsymbol{d}\|_\infty \leq 1/\mu\,\}} \equiv D_\mu^*(\boldsymbol{z}) = \|\boldsymbol{z}\|_1/\mu$ (if a norm $\|\cdot\|$ is used in the primal, then its dual norm $\|\cdot\|_*$ appears in the dual). Thus, (50)/(51) cover both the TRBM and the PBM, as well as with stabilizing terms that behave *both* as a distance and as the indicator of a ball. Also, one can have a trust region in the dual, such as $D_\mu^*(\boldsymbol{z}) = \imath_{\{\,\boldsymbol{z}\,:\,\|\boldsymbol{z}\|_\infty \leq \mu\,\}} \equiv D_\mu(\boldsymbol{d}) = \mu\|\boldsymbol{z}\|_1$, a setting not really considered so far.

The above properties are the basic ones, but other assumptions are required to closely reproduce the convergence properties of the PBM. For instance, D_μ *strongly coercive* ($\lim_{\|\boldsymbol{d}\|\to\infty} D_\mu(\boldsymbol{d})/\|\boldsymbol{d}\| = +\infty$), which is equivalent D_μ^* finite everywhere, ensures that (50) is always bounded below/(51) is nonempty. The assumption can be avoided if boundedness is guaranteed in other ways, the simplest one being that a lower bound $\underline{f} \leq f_*$ is *known* and explicitly inserted in $\mathcal{B}^i$ via the pair $(\,\boldsymbol{0}\,,\, f(\bar{\boldsymbol{x}}^i) - \underline{f}\,)$; in the case of (39) this is equivalent to inserting in $\mathcal{B}^i$ a $\underline{\boldsymbol{u}} \in \mathit{conv}(U)$ such that $A\underline{\boldsymbol{u}} = \boldsymbol{b}$. Also, *smoothness in* $\boldsymbol{0}$ is important for the properties of the algorithm, although not symmetrically between D_μ and D_μ^*. In particular, $\nabla D_\mu(\boldsymbol{0}) = \boldsymbol{0}$ (which is equivalent to strict convexity of D_μ^* in $\boldsymbol{0}$) ensures that $\boldsymbol{d}^i = \boldsymbol{0}$ implies that $\bar{\boldsymbol{x}}$ is optimal for (1); if D_μ is not differentiable in $\boldsymbol{0}$ the algorithm is not guaranteed to converge to an optimum of the problem, and this has to be ensured by forcing $\mu^i \to 0$ along iterations. Instead, smoothness of D_μ^* in $\boldsymbol{0}$ (which is equivalent to strict convexity of D_μ in $\boldsymbol{0}$) is crucial for proving convergence under "extreme aggregation", directly generalizing (36); the results can actually be strengthened somewhat by requiring

that the dependency on μ is "simple", i.e., that $D_\mu = \mu D \equiv D^*_\mu = D^*/\mu$ for some fixed D/D^* with the above properties. With a nonsmooth D^*_μ, in principle information cannot be discarded from $\mathcal{B}$ like for the CPM. Practical approaches to discard some information exist—it is always possible to entirely reset $\mathcal{B}$ at each SS—but no finite bound on $|\mathcal{B}|$ can be established (which may not be too much of an issue in practice due to the possibly dire consequences of too aggressive removals, cf. §3.1).

All in all, (more or less strong) convergence results are available for many choices of D_μ/D^*_μ, potentially allowing to adapt stabilization to the application at hand. For instance, piecewise-linear stabilizing terms with "few" pieces can be used to try to obtain a stabilization effect close to that of the PBM without paying the price of a quadratic MP [12]. Often the computational results show that the PBM has better practical convergence behaviour, and therefore is preferable [46]; however, the cost of making the MP a QP can be so high that piecewise-linear functions result in better running times [40, 43]. Arguably, Fenchel's duality would not have been necessary to use piecewise-linear functions, as the corresponding MP are LP ones; however, other forms of nonlinear stabilization have been proposed. For instance, *Bregman functions* [18] with the form $D_{\bar{\boldsymbol{x}}}(\boldsymbol{d}) = \psi(\bar{\boldsymbol{x}} + \boldsymbol{d}) - \psi(\bar{\boldsymbol{x}}) - \langle \nabla\psi(\bar{\boldsymbol{x}}), \boldsymbol{d}\rangle$ with ψ fixed, strictly convex, differentiable and with compact level sets, can be used to implicitly express the set X via a barrier-like approach, thus possibly making the MP easier to solve [64]. Also, other stabilization terms have been proposed in the context of solving (42) that could be adapted for GBM, such as the smooth approximations of $\|\cdot\|_1$ [84] (below, left) and of $\|\cdot\|_\infty/\mu$ [52] (below, right, for $\boldsymbol{z} \geq 0$)

$$D^*_\mu(\boldsymbol{z}) = \sum_i \begin{cases} z_i^2/(2\mu) & \text{if } -\mu \leq z_i \leq \mu \\ |z_i| - \frac{\mu}{2} & \text{otherwise} \end{cases} \quad , \qquad D^*_\mu(\boldsymbol{z}) = \ln \textstyle\sum_i e^{z_i/\mu} \ .$$

Thus, quite a variety of stabilization terms can be employed, offering a vast trade-off between the theoretical/practical convergence properties of the BM and the cost of the MP. We also mention that a somehow more general approach is proposed in [80], where BM are interpreted, a-la (47), as the problem of computing $f^*(\mathbf{0})$. The information provided by the oracle is used to construct the epigraph of $f^*_{\mathcal{B}}$, an inner approximation of the epigraph of f^* (cf. (52)), and a MP is solved that finds the closest point of epi $f^*_{\mathcal{B}}$ to $(\,\mathbf{0}\,,\,0\,)$ under a general norm $|||\cdot|||$. The GBM can be interpreted as an instance of this process where the norm is separable between the subgradient component and the linearization error component, i.e., $|||\,(\,\boldsymbol{z}\,,\,\alpha\,)\,||| = D^*(\,\boldsymbol{z}\,) + |\,\alpha\,|$, whereas the approach of [80] does not require this assumption. On the other hand, several important practical aspects of the method are not extensively discussed, and there is no computational indication that using more complex norms can significantly improve performances.

We finish this section mentioning that a Generalized DSBM (cf. §2.3) should be possible with

$$\min \left\{\, \underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}) + D_{\mu^i}(\boldsymbol{d}) \ : \ \underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}) \leq l^i \,\right\} \ . \tag{53}$$

Somewhat surprisingly, to derive a meaningful dual it is simpler to start with Lagrangian duality (as opposed to Fenchel's)

$$\begin{gathered} \max_{\rho\geq 0} \left\{ \ -\rho l^i + \min \left\{ \left\{ (1+\rho)\underline{f}^i(\bar{\boldsymbol{x}} + \boldsymbol{d}) + D_{\mu^i}(\boldsymbol{d}) \right\} \right\} = \\ [-]\ \min_{\rho\geq 0} \left\{ \rho l^i + (1+\rho)(\underline{f}^i)^*(\boldsymbol{z}/(1+\rho)) + \langle \boldsymbol{z}, \bar{\boldsymbol{x}}\rangle + D^*_{\mu^i}(-\boldsymbol{z}) \right\} \ , \end{gathered}$$

although in the second step one does apply (50)/(51) (together with standard properties of the conjugate, among which $(\gamma f(\cdot))^*(\boldsymbol{z}) = \gamma f^*(\boldsymbol{z}/\gamma)$ for $\gamma > 0$). Then, using (2)/(52) for $\underline{f}_{\mathcal{B}} = \check{f}_{\mathcal{B}}$ one gets

$$\begin{gathered} \min \rho l^i + (1+\rho) \textstyle\sum_{b\in\mathcal{B}} \alpha^b\theta^b + \langle \boldsymbol{z}, \bar{\boldsymbol{x}}\rangle + D^*_{\mu^i}(-\boldsymbol{z}) \\ \textstyle\sum_{b\in\mathcal{B}} \boldsymbol{z}^b\theta^b = \boldsymbol{z}/(1+\rho) \ , \ \boldsymbol{\theta} \in \Theta \ , \ \lambda \geq 0 \ , \end{gathered}$$

that via the (nonlinear) rescaling $\boldsymbol{\theta} \leftarrow (1+\rho)\boldsymbol{\theta}$ finally becomes

$$\begin{gathered} \min \rho l^i + \textstyle\sum_{b\in\mathcal{B}} \alpha^b\theta^b + \langle \sum_{b\in\mathcal{B}} \boldsymbol{z}^b\theta^b, \bar{\boldsymbol{x}}\rangle + D^*_{\mu^i}\big(\ -\sum_{b\in\mathcal{B}} \boldsymbol{z}^b\theta^b\ \big) \\ \textstyle\sum_{b\in\mathcal{B}} \theta^b = 1 + \rho \ , \ \boldsymbol{\theta} \geq \mathbf{0} \ , \ \rho \geq 0 \ , \end{gathered}$$

readily generalizing (34). To the best of our knowledge, this derivation is new; convergence of the GDSBM has not yet been firmly established, although it should follow easily enough by combining [36] with [26].

One property of all the stabilizing approaches discussed so far is that they are completely independent on the specific choice of f, comprised the fact that is has, or not, the form (39). While this is some sense an advantage, it also means that the stabilizing terms are not, on the outset, capable of *exploiting* any available information about the form of f. However, besides the stabilizing term D_μ, (50)/(51) also

depend on the model $\underline{f}_{\mathcal{B}}$. So far we have mostly assumed the use of the cutting-plane model $\check{f}_{\mathcal{B}}$, but most of the convergence arguments only require very few specific properties from $\underline{f}_{\mathcal{B}}$ [36]. Indeed, the model can be chosen to exploit specific properties of f, as discussed in the next section.

4 Alternative models

This section is devoted to improvements of the BM that pertain to using "better models" of f. Since all these are, in essence, orthogonal to the details of the stabilization, we will only present them in the context of the standard PBM (which is where, actually, they have for the most part been discussed), with the understanding that they could be applied to the other forms with some (possibly not entirely trivial) adjustment of the convergence analysis.

4.1 Quadratic models

Following well-established approaches in nonlinear optimization, the first idea that would likely spring to mind is to use quadratic models of f, in order to capture its second-order behaviour. As already remarked, this is possible using sophisticated tools that are beyond the scope of this treatment (cf. e.g. (11)). Yet, some attempts have used simpler techniques that are based on the concept that $\underline{f}_{\mathcal{B}}$ should not "deviate too much" from $\check{f}_{\mathcal{B}}$.

One such model is the piecewise-quadratic [3]

$$\breve{f}_{\mathcal{B}} = \max \left\{ \; q^b(\boldsymbol{x}) = f(\boldsymbol{x}^b) + \langle \boldsymbol{z}^b, \boldsymbol{x} - \boldsymbol{x}^b \rangle + \epsilon^b \|\boldsymbol{x} - \boldsymbol{x}^b\|_2^2/2 \; : \; b \in \mathcal{B} \; \right\} \; ,$$

i.e., the pointwise maximum of the quadratic expansions q^b of f generated at each $\boldsymbol{x}^b$; note that, unlike for $\check{f}$, this clearly requires keeping the $\boldsymbol{x}^b$ in $\mathcal{B}$. This is in general not a valid lower model of f, unless all $\epsilon^b = 0$ in which case it falls back to $\check{f}_{\mathcal{B}}$; yet, it is easy to compute "small enough" ϵ^b such that $\breve{f}_{\mathcal{B}}(\boldsymbol{x}^b) \leq f(\boldsymbol{x}^b)$ for all $\boldsymbol{x}^b$, i.e., the model never *knowingly overestimates* f. Actually, it is sufficient to guarantee the property only for a subset of the previous iterates, possibly only the current stability center $\bar{\boldsymbol{x}}$. The model can be translated w.r.t. $\bar{\boldsymbol{x}}$, although this now requires

$$\breve{\alpha}^b = \alpha^b - \epsilon^b \|\bar{\boldsymbol{x}} - \boldsymbol{x}^b\|_2^2/2 \quad \text{and} \quad \breve{\boldsymbol{z}}^b = \boldsymbol{z}^v + \epsilon^b(\bar{\boldsymbol{x}} - \boldsymbol{x}^b) \; ,$$

which allows to define the "doubly-stabilized" MP

$$\min \left\{ \, v + \mu^i \|\boldsymbol{d}\|_2^2/2 \; : \; v \geq \; \epsilon^b \|\boldsymbol{d}\|^2/2 + \langle \breve{\boldsymbol{z}}^b, \boldsymbol{d} \rangle - \breve{\alpha}^b \quad b \in \mathcal{B}^i \; , \; \gamma^i \|\boldsymbol{d}\|_2^2 \leq 2 \, \right\}$$

having both a proximal term weighted with μ^i and a trust region one governed by γ^i. The rationale for the trust region in the L_2 norm is that the problem is a quadratically constrained QP anyway, so there is no significant penalty in an extra quadratic constraint. The MP is actually a Second-Order Cone Program (SOCP); this is more easily seen computing its dual

$$\min \; \left\{ \; \frac{\|\sum_{b \in \mathcal{B}^i} \breve{\boldsymbol{z}}^b \theta^b\|_2^2}{2(\mu^i + \rho + \sum_{b \in \mathcal{B}^i} \theta^b \epsilon^b)} + \sum_{b \in \mathcal{B}^i} \breve{\alpha}^b \theta^b + \frac{\rho}{\gamma^i} \; : \; \boldsymbol{\theta} \in \Theta \; , \; \rho \geq 0 \right\} \; ,$$

where the apparently nasty fractional term in the objective function can be transformed into a rotated SOCP constraint with a well-known reformulation trick. Hence, the MP can be solved with off-the-shelf IP methods, at a cost comparable with a convex QP of the same size. All this allows to define a convergent BM, whose two stability parameters can be quite freely managed: indeed, as soon as at least one ϵ^b is strictly positive, one can even take $\mu^i = \rho^i = 0$, as the quadratic model is "self stabilizing". The convergence arguments follow the standard pattern of BM; the only nontrivial step is aggregation, as together with $\bar{\boldsymbol{z}}^i$ and $\bar{\alpha}^i$ one must also compute a $\bar{\boldsymbol{x}}^i$ to match, which requires some appropriate but overall simple computation. While the results seemed to show that this model was in fact capable of improving practical performances w.r.t. a standard PBM, this happened only with functions f that had the same piecewise-quadratic nature as $\check{f}_{\mathcal{B}}$.

Another recent take on the approach [55] is different in two key aspects: i) it insists in having only *one* quadratic term by modifying the proximal term in (26) into $\boldsymbol{d}^T H^i \boldsymbol{d}$, a-la (11), and ii) it insists on not *underestimating* the cutting plane model too much. The basic formula can be written in a "poorman's"

setting (cf. §3.2), where one has the aggregated linearization $(\,\bar{\boldsymbol{z}}^i\,,\,\bar{\alpha}^i\,)$ and just one other linearization $(\,\boldsymbol{z}^i\,,\,\alpha^i\,)$; then,

$$\langle \bar{\boldsymbol{z}}^i, \boldsymbol{d}\rangle - \bar{\alpha}^i + \tfrac{1}{2}\boldsymbol{d}^T H \boldsymbol{d} \geq \langle \boldsymbol{z}^i, \boldsymbol{d}\rangle - \alpha^i - \varepsilon \;\; \forall \boldsymbol{d} \in \mathbb{R}^n \;\; \equiv \;\; H \succeq \tfrac{1}{2(\alpha^i - \bar{\alpha}^i + \varepsilon)}(\,\bar{\boldsymbol{z}}^i - \boldsymbol{z}^i\,)(\,\bar{\boldsymbol{z}}^i - \boldsymbol{z}^i\,)^T \;\; . \tag{54}$$

Note that $\bar{\alpha}^i \leq \alpha^i + \varepsilon$ must hold for (54) to have any chance to hold (set $\boldsymbol{d} = \boldsymbol{0}$), i.e., the scaling term must be positive; apart from that ε is "free" and can serve as a stabilization parameter. One can then build a SemiDefinite Program (SDP) with as many semidefinite constraints of the form (54) as there are elements in $\mathcal{B}$ to find the least-curvature H that ensures that $\langle \bar{\boldsymbol{z}}^i, \boldsymbol{d}\rangle - \bar{\alpha}^i + \frac{1}{2}\boldsymbol{d}^T H \boldsymbol{d} \geq \check{f}^i(\boldsymbol{d}) - \varepsilon$; this can be shown in simple cases (f convex quadratic) to be reasonably related with the Hessian. Because solving the SDP at each step would be too costly, an approximate solution can be obtained by computing the Singular Value Decomposition of an appropriate matrix (think that with $\bar{\boldsymbol{z}}^i - \boldsymbol{z}^b$ as columns) and taking "a few" of the columns corresponding to the largest singular values. This has been shown to be quite successful in improving practical convergence speed of the BM in one application.

Albeit interesting, the previous two models are "general-purpose": they do not make any assumption on f, and therefore arguably cannot exploit any of its specific properties. In the next sections we will instead describe models that exploit different forms of structure in f.

4.2 Disaggregate models

Perhaps the most frequent structure in f is the sum one, i.e., $f(\boldsymbol{x}) = \sum_{k\in\mathcal{K}} f_k(\boldsymbol{x})$ where $\mathcal{K}$ is a finite index set. A prolific source of this kind of problems is the Lagrangian one, in which U in (40)—or, for that matter, $conv(U)$ in (42)—is a Cartesian product: $U = \bigoplus_{k\in\mathcal{K}} U_k$, so that

$$\max\left\{\; \textstyle\sum_{k\in\mathcal{K}} \langle \boldsymbol{c}_k, \boldsymbol{u}_k\rangle \;:\; \sum_{k\in\mathcal{K}} A_k \boldsymbol{u}_k = \boldsymbol{b} \;\;,\;\; \boldsymbol{u}_k \in U_k \quad k \in \mathcal{K} \;\right\} \tag{55}$$

for $\boldsymbol{u} = [\,\boldsymbol{u}_k\,]_{k\in\mathcal{K}}$, and therefore

$$f(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{b}\rangle + \textstyle\sum_{k\in\mathcal{K}} \left[\, f_k(\boldsymbol{x}) = \max\left\{\; \langle \boldsymbol{c}_k - \boldsymbol{x}A_k, \boldsymbol{u}_k\rangle \;:\; \boldsymbol{u}_k \in U_k \;\right\}\right] \;\; . \tag{56}$$

For each $k \in \mathcal{K}$, any optimal solution $\boldsymbol{u}_k(\boldsymbol{x})$ of (56) provides the individual function value $f_k(\boldsymbol{x}) = \langle \boldsymbol{c}_k - \boldsymbol{x}A_k, \boldsymbol{u}_k(\boldsymbol{x})\rangle$ and the individual subgradient $\boldsymbol{z}_k = -A_k\boldsymbol{u}_k(\boldsymbol{x}) \in \partial f_k(\boldsymbol{x})$. We immediately remark that there is a small (and intended) inconsistency between (56) and the original definition, in that in the former there actually are $|\mathcal{K}| + 1$ components of the sum, comprised the linear one $\langle \boldsymbol{x}, \boldsymbol{b}\rangle$; clearly such a term can (and should) be dealt with in a specific way, as discussed in details in §4.3. Disregarding this point for the time being, one could obviously define the *aggregated* function value and subgradient out of the individual $f_k(\boldsymbol{x})$ and $\boldsymbol{z}_k$, and then fall back to the previously developed theory. However, there is clearly another alternative: defining *individual models* for each component, say the cutting-plane ones

$$\check{f}^i_k(\boldsymbol{x}) = \max\left\{\; \langle \boldsymbol{z}^b_k, \boldsymbol{x}\rangle - \alpha^b_k \;:\; b \in \mathcal{B}^i_k \;\right\} \leq f_k(\boldsymbol{x}) \tag{57}$$

depending on *individual bundles* $\mathcal{B}^i_k = \{\,(\boldsymbol{z}^b_k, \alpha^b_k = \langle \boldsymbol{z}^b_k, \boldsymbol{x}^b\rangle - f_k(\boldsymbol{x}^b) = f^*_k(\boldsymbol{z}^b_k)\,\}$. We can still refer to $\mathcal{B} = [\,\mathcal{B}_k\,]_{k\in\mathcal{K}}$ as "the bundle", and still avoid to distinguish between the un-translated α^b_k and the linearization errors $\alpha^b_k(\bar{\boldsymbol{x}}) = \alpha^b_k - \langle \boldsymbol{z}^b_k, \bar{\boldsymbol{x}}\rangle + f_k(\bar{\boldsymbol{x}})$ (cf. (21)) unless strictly necessary. It is then immediate to define the *disaggregate master problem*

$$\min\left\{\; \langle \boldsymbol{b}, \boldsymbol{d}\rangle + \textstyle\sum_{k\in\mathcal{K}} v_k + \frac{\mu^i}{2}\|\boldsymbol{d}\|^2_2 \;:\; v_k \geq \langle \boldsymbol{z}^b_k, \boldsymbol{d}\rangle - \alpha^b_k \quad b \in \mathcal{B}^i_k \;\; k \in \mathcal{K} \;\right\} \tag{58}$$

using the disaggregate model instead of the original (aggregated) one. It is quite obvious that, for the same set of information produced by evaluating f (all the f_k), (58) provides a better representation of f than (26). This is clearer when comparing the dual of (58)

$$\min\left\{\; \tfrac{1}{2\mu^i}\|\boldsymbol{b} + \textstyle\sum_{k\in\mathcal{K}}\sum_{b\in\mathcal{B}^i_k} \boldsymbol{z}^b_k\theta^b_k\|^2_2 + \sum_{k\in\mathcal{K}}\sum_{b\in\mathcal{B}^i_k} \alpha^b_k\theta^b_k \;:\; \boldsymbol{\theta}_k \in \Theta_k \;\; k \in \mathcal{K} \;\right\} \tag{59}$$

with (27): in fact, its is obvious that the latter is the restriction of the former obtained by imposing that all the multipliers θ^b_k corresponding to all the individual subgradients attained at the same iterate b have the same value. Intuitively, "gluing together" the individual $\boldsymbol{z}^b_k$ into one aggregated $\boldsymbol{z}^b$ just because they have happened to have been produced at the same iteration is rather arbitrary, as they are in fact independent information about independent functions. Analogously, individual aggregated pairs $(\,\bar{\boldsymbol{z}}^i_k\,,\,\bar{\alpha}^i_k\,)$ can be obtained out of the solutions $\boldsymbol{\theta}^i_k$ of (59), and independently inserted in each $\mathcal{B}^i_k$; despite all them having been obtained with multipliers corresponding to one specific MP solution, there is no

reason why two different pairs should be later on constrained to each other. Nowhere this is clearer than in the Lagrangian case (56): (59) is equivalent to

$$[\langle \bar{\boldsymbol{x}}, \boldsymbol{b} \rangle +] \ \max \left\{ \ \textstyle\sum_{k \in \mathcal{K}} \langle \boldsymbol{c}_k - \bar{\boldsymbol{x}} A_k, \boldsymbol{u}_k \rangle - \frac{1}{2\mu^i} \| \sum_{k \in \mathcal{K}} A_k \boldsymbol{u}_k - \boldsymbol{b} \|_2^2 \ : \ \boldsymbol{u}_k \in U_k^i \ \ k \in \mathcal{K} \right\} \ . \tag{60}$$

In other words, the feasible region of (60) is a Cartesian product of convex hulls, whereas that of the aggregated (43) is the convex hull of a Cartesian product: it is very easy to see that the former set (for, ideally, the same $\mathcal{B}_k^i$) is much larger than the latter one. All this justifies why *disaggregate BM* using (57) typically converge much faster than aggregated ones, all the rest being equal [6,14,43]; indeed, convergence happens when enough information has been accrued that allows to express the optimal solution, and disaggregate BM make much better use of the gathered information.

Of course, there is a negative aspect in using disaggregate models: the master problems are larger (roughly, "by a factor of $|\mathcal{K}|$"), and therefore potentially (much) more costly to solve. Therefore, the trade-off between aggregated and disaggregate BM strongly depends on the relative weight of the MP cost and of the f_k computation cost. Often, the increase in convergence speed obtained by using a disaggregate model is worth the extra MP cost. Indeed, by converging much faster the disaggregate BM can actually end up collecting *less* information that the aggregated one (while making much better use of it), so that the disaggregate MP simply does not have the time to become too large. However, if the subproblems (56) are easy but "many", the cost of the disaggregate MP can become by far the computational bottleneck of the algorithm.

In order to face this issue, an intuitively promising approach is *partial aggregation*. That is, one may partition $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2 \cup \ldots \cup \mathcal{K}_h$ into h disjoint subsets, and then define the corresponding partly aggregated functions, subgradients and linearization errors. This is clearly possible, with the size of the MP now increasing "only" of a factor of h, at the cost of some (but less than in the fully aggregated case) arbitrary information aggregation. It is still unclear how to choose h, and how to distribute the different components across the partition. Some experiments [82, Chapter 2] seemed to show a potential for the approach, in that a small h was sufficient to significantly increases convergence speed w.r.t. the fully aggregated case, becoming comparable to that of the fully disaggregate case as a fraction of the latter's MP cost. However, even within the same class of problems the trade-off was very dependent on the specific type of instance, and it seemed hard to devise dependable guidelines. In this line of approach, it might be useful if the partition could be *dynamic*; this is indeed possible, as advocated in [96]. By arbitrarily choosing any $\mathcal{Z} \subseteq \mathcal{K}$ one may insert in (58) *partly aggregated cuts*

$$\textstyle\sum_{k \in \mathcal{Z}} v_k \geq \langle \sum_{k \in \mathcal{Z}} \boldsymbol{z}_k^i, \boldsymbol{d} \rangle - (\sum_{k \in \mathcal{Z}} \alpha_k^i) \ . \tag{61}$$

Recent results [55] indicate that such an approach may be promising, in particular by using disaggregate cuts for a small set of "critical" components (whose subgradients seem to vary rapidly, thus exhibiting nondifferentiable behaviour), while all the remaining ones are aggregated into one component. A specific application where this technique makes especially sense is two-stage stochastic programs, since there a partly aggregated cut has a clear meaning in terms of sub-sampled estimate of the true subgradient (cf. §5.2). It is not surprising, then, that good results have been reported, e.g. with a level-type BM [100]. For problems with fixed recourse, aggregation rules can be defined that benefit from information about the function and exploit ideas already presented in the stochastic programming community, again with a significant practical effect [95]. Yet, the implementation details required to achieve good results seem to be rather dependent on the specific application; this therefore remains an interesting, but still wide open, research line.

4.3 Constraints and easy components

The sum-funcion structure paves the way for further exploiting the specific structure of some of the components. We have actually already seen this happening: the function (56) has the linear component $f_0(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{b} \rangle$, which in the MP (58)/(60) is *not* treated like the other f_k, but rather "directly inserted in the model". The principle is readily applicable each time one of the components has the appropriate structure. That is, assume for simplicity that $\mathcal{K} = \{ 0 , 1 \}$, where f_1 is produced by a standard oracle, whereas f_0 is "easy" in the sense that it can be *directly written into the MP*:

$$\min \left\{ \ f_0(\bar{\boldsymbol{x}} + \boldsymbol{d}) + v_1 + \tfrac{\mu^i}{2} \| \boldsymbol{d} \|_2^2 \ : \ v_1 \geq \langle \boldsymbol{z}_1^b, \boldsymbol{d} \rangle - \alpha_1^b \quad b \in \mathcal{B}_1^i \ \right\} \ . \tag{62}$$

This is how $f_0(\boldsymbol{x}) = \langle \boldsymbol{b}, \boldsymbol{x} \rangle$ was dealt with in (58), and it is also the standard treatment of constraints in BM: with $f^0 = \imath_X$, this amounts to adding the constraints "$\bar{\boldsymbol{x}} + \boldsymbol{d} \in X$" to (8). Obviously, the general assumption is that (62) is not much more costly to solve than (58), which happens e.g. when X is defined by a "small" set of "simple" (say, linear or conic) constraints, as in the original (3). Clearly, a BM using (62) necessarily has to work if a BM using (58) was: the MP has better (indeed, "perfect") knowledge of f_0. Extension to any number of "easy" and "standard" component is immediate.

It is now appropriate to remark that constraints can be dealt with dynamically, so that a polyhedron X represented by a very large (say, exponential) number of constraints can still be used under the standard assumption that an efficient separation algorithm exist. A revealing case is that of a Lagrangian component over a non-compact polyhedron, i.e.,

$$f_0(\boldsymbol{x}) = \max\{\, \langle \boldsymbol{c}_0 - \boldsymbol{x}A_0, \boldsymbol{u}_0 \rangle \,:\, \bar{U}_0 \boldsymbol{u}_0 \leq \bar{\boldsymbol{u}}_0 \,\} \tag{63}$$

with $\bar{U}_0/\bar{\boldsymbol{u}}_0$ matrix/vector of appropriate dimension, for which $f_0(\boldsymbol{x}^i) = +\infty$ can happen. This means that one *ray* ω^i of the polyhedron exists (and is identified by whatever LP solver is used to compute f_0) that is also an ascent direction, i.e., $\bar{U}_0 \omega^i \leq 0$ and $\langle \boldsymbol{c}_0 - \boldsymbol{x}^i A_0, \omega^i \rangle > 0$. Obviously, the very same ray will prove unboundedness for any other $\boldsymbol{x}$ such that $\langle \boldsymbol{c}_0 - \boldsymbol{x}A_0, \omega^i \rangle > 0$; in other words, ω^i defines the *constraint* $\langle \boldsymbol{c}_0, \omega^i \rangle \leq \langle \boldsymbol{x}, A_0 \omega^i \rangle$ that must be satisfied by all points in the domain of f_0. Thus, each time $f_0(\boldsymbol{x}^i) = +\infty$ a new constraint can be added to the MP that "cuts away" $\boldsymbol{x}^i$, thereby necessarily changing its solution. Constraints are in fact a slightly different form of linearization describing epi f_0, which we can call "vertical" since the coefficient of v_0 is 0, and can be added to $\mathcal{B}_0^i$ instead of the standard ones; this only means that the corresponding dual variables θ^b in (27) do not participate to (have 0 coefficient in) the simplex constraint. Assuming that *the oracle only reports a finite set of rays* (say, the extreme ones) and that *vertical linearizations are never removed from* $\mathcal{B}_0$, then $f_0(\boldsymbol{x}^i) = +\infty$ can only happen a finite number of times, and the BM is still provably convergent. Similarly, constraints describing any polyhedron X for which a separation algorithm is available can be dynamically added to the MP whenever $\boldsymbol{x}^i \notin X$.

However, vertical linearizations/constraints are in some sense "more delicate": removing or aggregating them is not as easy as with subgradients. Removal is possible with the usual rules—as soon as a SS is performed, $\mathcal{B}^i$ can be entirely reset of either type of linearization—but no finite bound on $|\mathcal{B}|$ can be established. Furthermore, all this only works under the assumption that the set of constraints is finite in the first place. It is easy to see where the catch is by thinking to (1) in which f is "simple" (say, linear) and X is given by a separation oracle only reporting vertical linearizations. If X is not a polyhedron, the CPM would still work—actually, this is the very setting in which it has been defined [60]—but it is completely possible that $f(\boldsymbol{x}^i) = \infty$ for *all* iterates $\boldsymbol{x}^i$. While this is not a problem for the CPM, it is typically so for a BM, which is based on tests like (7) to manage the stabilization center. Hence, either some mechanism is required that ensures that the BM obtains $f(\boldsymbol{x}^i) < \infty$ "frequently enough", or some alternative test has to be employed. Customarily, BM dealing with "complicated" constraints assume $X = \{\, \boldsymbol{x} \in \mathbb{R}^n \,:\, c(\boldsymbol{x}) \leq 0 \,\}$, where that *both* f and c are finite-valued; hence this does not exactly cover the previous example. Note that $c(\cdot)$ can w.l.o.g. be taken as a scalar function, since any finite set of convex constraints can be turned into one by taking their maximum, which is still convex (but nondifferentiable). Finiteness of $c(\cdot)$ is crucial to implement *infeasible* BM, which can make good use of unfeasible iterates $c(\boldsymbol{x}^i) > 0$; the required theoretical tool is the *improvement function* $h_{\bar{\boldsymbol{x}}}(\boldsymbol{x}) = \max\{\, f(\boldsymbol{x}) - f(\bar{\boldsymbol{x}}) \,,\, c(\boldsymbol{x}) \,\}$ such that $\bar{\boldsymbol{x}}$ solves the constrained (1) if and only if it is an unconstrained minimizer of $h_{\bar{\boldsymbol{x}}}$, the optimal value then being $h_{\bar{\boldsymbol{x}}}(\bar{\boldsymbol{x}}) = 0$ [87]. Basically, a standard unconstrained BM—allowing, in particular, aggregation—can then be used to minimize $h_{\bar{\boldsymbol{x}}}$; subgradients of both f and c computed at previous iterates are separately kept, analogously to (58), and transformed into subgradients of $h_{\bar{\boldsymbol{x}}}$ by simple formulæ. If an appropriate improvement in the value of $h_{\bar{\boldsymbol{x}}}$ is attained, the stability center is changed; this also changes the objective function, but again existing information—comprised aggregated one—can be used to compute valid approximate subgradients to the new $h_{\bar{\boldsymbol{x}}}$, allowing the method to continuously accrue information as in the standard case. The algorithm can then be shown to converge; furthermore, if a feasible iterate is ever produced, then all subsequent iterates remain feasible. Alternatively, filter techniques can be used [59]. Under stronger assumptions on X, *feasible* BM can be constructed: for instance, [68] requires *knowledge* (and hence, a fortiori, existence) of a *Slater point* $\boldsymbol{x}_{int}$ such that $c(\boldsymbol{x}_{int}) < 0$. Hence, whenever an unfeasible iterate $\boldsymbol{x}^i$ is obtained, an *interpolated point* can be defined—not unlike in the IOA (cf. §2.5)—in the segment $[\, \boldsymbol{x}_{int} \,,\, \boldsymbol{x}^i \,]$ that is feasible, and therefore whose objective function value can be used in the descent test. A similar approach has been used in [94] in the context of chance-constrained

optimization; the specific advantage is that the computation of the chance constraint requires a costly numerical procedure that can be ill-conditioned for "extreme" $\boldsymbol{x}^i$ with very high or low probability, whereas it is more reliable and efficient for points "in the middle". Yet, for some applications experiments have shown that infeasible starts can actually be beneficial [91, 99].

Returning to the original subject of this paragraph, it is clear that inserting f_0 in the MP is not limited to linear or indicator functions, but can be done whenever the MP remains "reasonably easy". This is often the case of Lagrangian functions, where the underlying Lagrangian subproblems can have special structures that make their dual function manageable without resorting to linearizations. An interesting example are *nonlinear multicommodity network design problems with congestion costs*, if only because they have been tackled twice, once with an ACCPM [5] and once with a PBM [71]. In the problem, the objective function is nonlinear because of many single-variable terms of the form $k(t) = t/(c-t)$, where t is the total flow on an arc of the underlying network and c is its capacity; this is the widely used *Kleinrock's delay function*. Once the linking constraints are relaxed, one is typically left with many single-variable optimization problems of the form $f(x) = \min\{\, t/(c-t) - xt \,:\, 0 \leq t < c \,\}$ (the original problem being a minimization one), each one depending on one single Lagrangian multiplier x; this immediately reveals itself as *the opposite of the conjugate of Kleinrock's delay function*, $-k^*(x)$. Due to its simple form this can be computed with a closed formula: $k^*(x) = 1 + cx - 2\sqrt{cx}$ whenever $x \geq 1/c$. Hence, f_0 in (62) is a sum of those terms. Directly inserting these in the MP results in a problem that is no longer a QP; to address this issue, in [71] f_0 is replaced with its second-order approximation around the current stability center $\bar{\boldsymbol{x}}^i$, resulting in a hybrid BM/Newton's method. Only relatively minor changes are required in the convergence analysis, all using well-understood techniques from smooth optimization (basically, an appropriate line search); furthermore, the Newton's term directly stabilizes the approach, removing the need for the "artificial" proximal stabilization $\|\boldsymbol{d}\|_2^2$. This is even less of an issue for ACCPM, whose MP (16) is already not a QP: adding the terms corresponding to f_0 in the KKT system of the IP method (itself again basically Newton's method) is easy. In both cases, inserting "exact" information about one (many) component(s) of f in the MP is shown to significantly improve performances in practice.

The approach does not even require the conjugate being easy to compute: as advocated in [43], any Lagrangian function whose form is "not more complex than that of the MP" lends itself to the treatment. That is, consider

$$\max \left\{\, \langle \boldsymbol{c}_0, \boldsymbol{u}_0 \rangle + \langle \boldsymbol{c}_1, \boldsymbol{u}_1 \rangle \;:\; \bar{U}_0 \boldsymbol{u}_0 \leq \bar{\boldsymbol{u}}_0 \;,\; \boldsymbol{u}_1 \in U_1 \;,\; A_0 \boldsymbol{u}_0 + A_1 \boldsymbol{u}_1 = \boldsymbol{b} \,\right\}$$

where f_0 is again (63). The key is, again, duality: while the dual of (62)

$$\min \left\{\, \tfrac{1}{2\mu^i} \|\boldsymbol{b} - \boldsymbol{z}_0 - \textstyle\sum_{b \in \mathcal{B}_1^i} \boldsymbol{z}_1^b \theta_1^b\|_2^2 + \sum_{b \in \mathcal{B}_1^i} \alpha_1^b \theta_1^b - \bar{\boldsymbol{x}} \boldsymbol{z}_0 + (f_0)^*(\boldsymbol{z}_0) \;:\; \theta_1 \in \Theta_1 \,\right\}$$

may at first look intimidating, f_0^* is, basically, nothing else than the original Lagrangian subproblem: in other words, the above can be rewritten as

$$\begin{array}{ll} \max & \langle \boldsymbol{c}_0, \boldsymbol{u}_0 \rangle + \sum_{b \in \mathcal{B}_1^i} \alpha_1^b \theta_1^b + \langle \bar{\boldsymbol{x}}, \boldsymbol{z} \rangle - \frac{1}{2\mu^i} \|\boldsymbol{z}\|_2^2 \\ & \boldsymbol{z} = \boldsymbol{b} - \sum_{b \in \mathcal{B}_1^i} \boldsymbol{z}_1^b \theta_1^b - A_0 \boldsymbol{u}_0 \quad , \quad \bar{U}_0 \boldsymbol{u}_0 \leq \bar{\boldsymbol{u}}_0 \quad , \quad \theta_1 \in \Theta_1 \end{array} \;. \tag{64}$$

The idea is therefore straightforward when seen in the dual MP: for the "standard" component the usual linearization is employed, whereas the "easy" one is basically *inserted unchanged in the (dual) MP*. This can be done beyond LPs; for instance, if the objective function $c_0(\boldsymbol{u}_0)$ were convex quadratic, then (64) would still be a convex QP, hence roughly as hard to solve as the original MP. Again, any BM working with an approximated model $\underline{f}_0$ will *a fortiori* work if the "true" f_0 is used, once a few minor details are taken care of. Of course, trade-offs reveal themselves in practice: (64) may be more costly to solve, especially at the beginning, but the part concerning f_0 will never grow in size, as opposed to the part concerning f_1. Furthermore, by having an "exact" model for one component one can expect faster convergence, often quite significantly so, as repeatedly reported in applications as diverse as multicommodity network design problems [43], stochastic unit commitment problems [89], chance-constrained optimization [92] and SDP relaxations for hard combinatorial problems [46].

4.4 Specialized dynamic models

Most (but not all) specialized models of §4.3 are "static", in that all the information corresponding to f_0 is known at the beginning of the solution process. This is clearly not necessary, as the present section

will show.

A specialized model is the basis of the *Spectral* BM (SBM) [56] for solving SDP. This starts with the fact that the dual of the standard SDP

$$\max \left\{ \langle C, U \rangle \ : \ AU = b \ , \ U \succeq 0 \right\}$$

under mild assumptions can be recast as the eigenvalue optimization problem

$$\min \left\{ f(\boldsymbol{x}) = \langle \boldsymbol{b}, \boldsymbol{x} \rangle + \lambda_{\max}(C - \boldsymbol{x}A) \right\} \ ,$$

with $\lambda_{\max}(\cdot)$ indicating the maximum eigenvalue of a matrix, a convex nondifferentiable function. Each time $f(\boldsymbol{x})$ is computed, by standard linear algebra techniques, any eigenvector $\boldsymbol{w}$ associated with the maximal eigenvalue produces a subgradient $\boldsymbol{z} = \boldsymbol{b} - A(\boldsymbol{w}\boldsymbol{w}^T)$; that is, $\partial f(\boldsymbol{x})$ is spanned by all possible such eigenvectors, and therefore f is differentiable only if the maximum eigenvalue has multiplicity one. Rather than the standard $\check{f}$, the SBM uses

$$\underline{f}_{\mathcal{B}}(\boldsymbol{x}) = \max \left\{ \langle \boldsymbol{b}, \boldsymbol{x} \rangle + \langle C - \boldsymbol{x}A, W \rangle \ : \ W \in W_{\mathcal{B}} \right\} \ ,$$

with $W_{\mathcal{B}} = \{ W = \theta \bar{W}_{\mathcal{B}} + P_{\mathcal{B}} V P_{\mathcal{B}}^T \ : \ \theta + \mathrm{tr}(V) = 1 \ , \ V \succeq 0 \}$. At first read, one can take the columns of the matrix $P_{\mathcal{B}}$ as being the (orthogonalized) eigenvectors $\boldsymbol{w}^b$ computed at previous iterations, and $\bar{W}_{\mathcal{B}}$ as corresponding to the aggregated subgradient $\bar{\boldsymbol{z}}^i$, although updating $P_{\mathcal{B}}$ and $\bar{W}_{\mathcal{B}}$ at each iteration requires some care. All in all, minimizing $\underline{f}_{\mathcal{B}}$ is a SDP, and a small-scale one if the size of $P_{\mathcal{B}}$ is kept in check; hence, it can be efficiently solved by IP methods. Clearly, adding a quadratic stabilizing term (or a trust region in the L_2 norm, for that matter) does not significantly change the computational cost of the MP. However, note that the efficiency of the MP solution is strictly related to the fact that the main matrix variable V has small size; this, for instance, may change if constraints $\boldsymbol{x} \in X$ (even simple bounds) are present, requiring the use of nontrivial techniques [54]. Not surprisingly, using the specialized model is much more efficient than using $\check{f}_{\mathcal{B}}$, and it can be competitive with IP methods in particular for solving sparse large-scale SDP.

In the somewhat different context of Lagrangian functions of structured problems, a quite general class of models has been proposed. The idea is that the standard Dantzig-Wolfe reformulation of $\mathrm{conv}(U)$, which gives rise to the standard cutting-plane model (cf. §3.3), is not the only possible formulation that lends itself to dynamic generation. Motivated by results on 0-1 reformulations of multicommodity network design problems [39], general requirements have been defined for any other "large" formulation of $\mathrm{conv}(U) = \{ \ \boldsymbol{u} = C\boldsymbol{\theta} \ : \ \Gamma\boldsymbol{\theta} \leq \boldsymbol{\gamma} \ \}$ that can be "constructed piecemeal" [40]. In this setting, the bundle is $\mathcal{B} = (\mathcal{B}^c, \mathcal{B}^r)$, where $\mathcal{B}^c$ is a subset of the variables $\boldsymbol{\theta}$ (columns of Γ and C), and $\mathcal{B}^r$ is a subset of the constraints (rows in Γ) which impact *at least one* variable in $\mathcal{B}^c$; this immediately defines the restrictions $\boldsymbol{\theta}_{\mathcal{B}}$, $\Gamma_{\mathcal{B}}$, $\boldsymbol{\gamma}_{\mathcal{B}}$ and $C_{\mathcal{B}}$ of the formulation. The first requirement is that any partial solution can always be completed with zeroes, i.e., $\Gamma_{\mathcal{B}}\bar{\boldsymbol{\theta}}_{\mathcal{B}} \leq \boldsymbol{\gamma}_{\mathcal{B}}$ and $\boldsymbol{\theta} = [\, \bar{\boldsymbol{\theta}}_{\mathcal{B}} \, , \mathbf{0} \,] \Longrightarrow \Gamma\boldsymbol{\theta} \leq \boldsymbol{\gamma}$; this immediately implies that

$$U_{\mathcal{B}} = \left\{ \nu = C_{\mathcal{B}}\boldsymbol{\theta}_{\mathcal{B}} \ : \ \Gamma_{\mathcal{B}}\boldsymbol{\theta}_{\mathcal{B}} \leq \boldsymbol{\gamma}_{\mathcal{B}} \right\} \ \subseteq \ conv(U) \ ,$$

and therefore that

$$\underline{f}_{\mathcal{B}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{b} \rangle + \max \left\{ \langle \boldsymbol{c} - \boldsymbol{x}A, C_{\mathcal{B}}\boldsymbol{\theta}_{\mathcal{B}} \rangle \ : \ \Gamma_{\mathcal{B}}\boldsymbol{\theta}_{\mathcal{B}} \leq \boldsymbol{\gamma}_{\mathcal{B}} \right\}$$

is an alternative lower model of f. Hence, the MP can be defined that uses this model; again, this is more easily seen in the dual, which is just (with the obvious notation)

$$\max \left\{ \langle \boldsymbol{c}, \boldsymbol{u} \rangle + \langle \bar{\boldsymbol{x}}, \boldsymbol{b} - A\boldsymbol{u} \rangle - \tfrac{1}{2\mu^i} || \, A\boldsymbol{u} - \boldsymbol{b} \, ||_2^2 \ : \ \boldsymbol{u} = C^i\boldsymbol{\theta}_{\mathcal{B}^i} \ : \ \Gamma^i\boldsymbol{\theta}_{\mathcal{B}^i} \leq \boldsymbol{\gamma}^i \right\}$$

(cf. (43)). The other necessary assumption is that, once the oracle is called in $\boldsymbol{x}^i$ and a new $\boldsymbol{u}^i$ is obtained, it can be efficiently used to update $\mathcal{B}^i$. This can be stated (intentionally vaguely) as follows: if $\bar{\boldsymbol{u}} \in \mathrm{conv}(U) \setminus U_{\mathcal{B}}$, it must be easy to update $\mathcal{B}$ to a set $\mathcal{B}' \supset \mathcal{B}$ such that there exists $\mathcal{B}'' \supseteq \mathcal{B}'$ with $\bar{\boldsymbol{u}} \in U_{\mathcal{B}''}$. In plain words, one has to be able to see what are the missing variables and constraints in the formulation, and add at least some of them; this is basically a "dynamic version of the easy components approach". It is then possible to develop a BM using this approach, which generalizes the Dantzig-Wolfe decomposition/Column Generation; this has been proven efficient in several applications [39, 86, 89].

4.5 Diminishing the MP cost

Many of the ideas discussed in the last sections lead to "large" MP; solving them can therefore easily become the computational bottleneck. Although the increased convergence speed may still make these large MP worthwhile, it is clear that techniques for lessening this computational cost could be crucial; some notable developments are discussed here.

The first have to do with the fact that $\boldsymbol{x}$ can be a "very long" vector. For instance, in the Lagrangian case (39), the constraints $A\boldsymbol{u} = \boldsymbol{b}$ can be exponentially many (of course, with an attached efficient separation routine), or anyway a very large number. Especially if the constraints are inequalities, though, one can expect only a small fraction of them actually being active at optimality, which means that only a small fraction of the components of $\boldsymbol{x}$ will be different from 0 at any optimal solution. One can then define a *Dynamic* BM (DBM)—be it a PBM [10, 38, 44], a SBM [53], a ACCPM [5] or any other, even with specialized models [43], and comprised subgradient-type methods [41]—that has, basically, a simple *active-set* strategy on $\boldsymbol{x}$. At the beginning, only a small subset of the variables (constraints in the dual) is actually defined in the MP, which is therefore smaller and cheaper. An arbitrary number of iterations can be performed with $\boldsymbol{x}$ restricted to the active set; then, occasionally—but surely if convergence in the current subspace is detected—one has to check the entire subgradient to see if some components need to be added to the active set. In the Lagrangian case, this simply amounts at verifying which of the constraints (say) $A\boldsymbol{u} \leq \boldsymbol{b}$ are violated by the *aggregated primal solution* $\bar{\boldsymbol{u}}^i$ (cf. (44)). If no components are ever removed from the active set the DBM is trivially convergent: after a finite number of updates the active set is the full space, and "true convergence" begins. Careful removal is also possible with mild assumptions on the separation process [10], although in practice the technique works well even without them. This has been shown to considerably improve performances, especially when the cost of the f computation is low [38, 41, 43, 44].

The approach of [67] rather deals with approximately solving MPs for sum-structured f (cf. §4.2). The idea is (again and again) quite simple when seen from the dual viewpoint: each of the (for simplicity) two components has distinct dual variables, say $\boldsymbol{\theta}_0$ and $\boldsymbol{\theta}_1$. It is then easy to implement a *block descent approach*, where $\boldsymbol{\theta}_0$ is kept fixed to some (feasible) value and $\boldsymbol{\theta}_1$ is optimized, and then the roles are reversed. This may also work when the $\boldsymbol{\theta}_k$ are not convex multipliers, say as in (64), and potentially when there are more than two, although to the best of our knowledge this has never been studied. From the primal viewpoint, this means that one of the two models $\underline{f}_0$ and $\underline{f}_1$, in turn, is substituted with its aggregated linearization $(\, \bar{\boldsymbol{z}}_h^i \,,\, \bar{\alpha}_h^i \,)$. This may allow to use specialized solvers that exploit the individual structure of the two separate subproblems. For instance, $\underline{f}_0$ may be the indicator function of a "simple" set X, say a box, whereas $\underline{f}_1$ may be the standard cutting-plane model; then, the first subproblem is the optimization of a linear function on a box, whereas for the second specialized algorithms exist that are more efficient in the unconstrained case [34] (although the latter algorithm specifically deals with box constraints with a technique that is not entirely uncorrelated with the one we are discussing). A very small number of iterations, down to only one, may be sufficient to construct a direction $\boldsymbol{d}^i$ that allows to continue the BM, thus potentially reducing the MP cost. The approach may applied many different structures, see e.g. again [54].

Finally, it is clearly always possible to specialise well-known approaches to the specific structure of the MP. This is the case of [34] for active-set methods and of [74] for structure-exploiting IP methods applied to the parallel solution of the disaggregate MP (58)/(59).

5 Inexact and incremental approaches

Overall, the computational cost of the BM depends on both the number of iterations and their cost, in turn the sum of the MP cost and of the oracle cost. Clearly, reducing the number of iterations (improving convergence speed) is of paramount importance to reduce the total cost, and it has been therefore the focus of basically all the discussion so far. Indeed, often paying a larger MP cost to reduce the number of iteration is worth, although of course the cost of the MP must also be kept in check (cf. §3.2, 4.2, 4.4, 4.5). What has not been discussed so far are methods to decrease the oracle cost. These would hardly seem to be subject of a general treatment in BM, since they clearly depend on the specific application giving rise to (1); however, general concepts of *approximate oracle* can be defined, whereby one loosens the requirement that $f(\boldsymbol{x})$ be computed exactly, and that $\boldsymbol{z} \in \partial f(\boldsymbol{x})$. This can be clearly beneficial, if

only in the Lagrangian case (39) where the oracle is an optimization problem (but then again, any (1) can be considered a dual problem, cf. (47)); allowing to solve it approximately should reasonably decrease its cost. This is the subject of the present section.

5.1 Inexact approaches

The Lagrangian case is indeed a good one to inform the discussion: an approximate oracle for (39) (with $\boldsymbol{x} = \boldsymbol{x}^i$) might just compute any feasible solution $\underline{\boldsymbol{u}}^i \in U$, hopefully a "good" one. Using $\underline{\boldsymbol{u}}^i$ instead of the optimal solution $\boldsymbol{u}^i$ yields a lower bound $\underline{l}^i = \langle \boldsymbol{c} - \boldsymbol{x}^i A, \underline{\boldsymbol{u}}^i \rangle + \langle \boldsymbol{x}^i, \boldsymbol{b} \rangle \le f(\boldsymbol{x}^i)$, together with $\underline{\boldsymbol{z}}^i = \boldsymbol{b} - A\underline{\boldsymbol{u}}^i$ such that $\underline{\boldsymbol{z}}^i \in \partial_{\varepsilon^i} f(\boldsymbol{x}^i)$ for the error $\varepsilon^i = f(\boldsymbol{x}^i) - \underline{l}^i \ge 0$. That is, such an approximated oracle delivers ε-subgradients rather than subgradients, and lower approximations to the function value. Crucially, the (say) cutting-plane model constructed with this information is still is a valid lower model, which makes it surprisingly easy to define an *Inexact BM* (IxBM). In fact, assuming that $\varepsilon^i \to 0$ "naturally" along the iterations, there is basically nothing to do [63]. In other words, as in the case of subgradient-type methods [22], what really counts for BM is the *asymptotic maximum error* $\varepsilon^\infty = \limsup_{i\to\infty} \varepsilon^i$: any "large" error $\varepsilon^i \gg \varepsilon^\infty$ occurring in the early iterations can be automatically corrected as the algorithm proceeds towards the optimum. This is an attractive feature in that, intuitively, it should not be required that the function be computed with high accuracy at the beginning of the algorithm, while the error reasonably need be reduced when approaching the optimal solution. However, such an *asymptotically exact* oracle is not necessary: a BM can converge under the quite minimal condition that $\varepsilon^i \le \bar{\varepsilon} < \infty$, with $\bar{\varepsilon}$ fixed but *not necessarily known*. Of course, in this case one can expect nothing better than a $\bar{\varepsilon}$-optimal solution [22, Observation 2.7].

In order to ensure convergence, though, some modifications are necessary. This stems from the fact that defining the linearization errors as $\underline{\alpha}^b(\bar{\boldsymbol{x}}) = \underline{l}^i - [\, \underline{l}^b + \langle \boldsymbol{z}^b, \bar{\boldsymbol{x}} - \boldsymbol{x}^b \rangle \,]$, i.e., using the lower estimates in place of the function values $f(\bar{\boldsymbol{x}}^i)$ and $f(\boldsymbol{x}^b)$, may lead to $\underline{\alpha}^b < 0$. Indeed, $\boldsymbol{z}^b$ is a $(\underline{\alpha}^b + \varepsilon^b)$-subgradient of f at $\bar{\boldsymbol{x}}$, with $\underline{\alpha}^b + \bar{\varepsilon} \ge \underline{\alpha}^b + \varepsilon^b \ge 0$; yet, $\bar{\varepsilon}$ and ε^b are unknown. In turn, when put e.g. in (28) this may lead to $v^i > 0$, i.e., $\boldsymbol{d}^i$ not being a descent direction. The point is that $\bar{\boldsymbol{x}}^i$ has been chosen as the stability center on the basis of $\underline{l}^i$, implicitly assuming it to be a reasonable approximation of the function value; yet, later on other information inserted in in $\mathcal{B}^i$ reveals that in fact $\underline{l}^i \ll f(\bar{\boldsymbol{x}}^i)$. A possible solution, originally due to [66], is to exploit the fact that any BM has one (or more) proximal parameter(s), that can be almost freely adjusted. The idea is that whenever $v^i > 0$ the proximal parameter is adjusted so that the MP becomes *less* stabilized—say, μ^i is reduced in the PBM—so that $v^{i+1} < v^i$, hopefully becoming negative (enough). This is called a *Noise Reduction* (or Noise Attenuation) step (NR), because the "noise" in the function computation is higher than the "signal" corresponding to v^i; by increasing v^i (in absolute value), the signal-to-noise ratio also increases (being the error bounded). With minimal care, a finite number of NR leads to two possible outcomes. The first is that the solution $\boldsymbol{x}^i$ of the MP becomes a solution of (3), i.e., a global minimum of the model $\underline{f}^i$; in this case, $\bar{\boldsymbol{x}}$ is $\bar{\varepsilon}$-optimal and the BM can stop (it actually has to, as there is no other recourse). Otherwise, v^i will eventually become "sufficiently negative", and the normal course of the BM can resume. This approach has been shown to work for the PBM under even looser assumptions on the oracle, i.e., $\underline{l}^i$ may not even be a guaranteed lower bound on $f(\boldsymbol{x}^i)$ and $\boldsymbol{z}^i$ may not even be a guaranteed ε-subgradient at $\boldsymbol{x}^i$, provided that the errors are suitably bounded [25, 99]. The PLBM has some different technicalities [24], in particular in the constrained case [93]; interestingly, the DSBM does not require NR at all, since the level constraint always ensures that $v^i < 0$ [26, §4].

The previous analysis assumed no control on the oracle error, but this is not the only possible case. There have been different definitions of *controllable* inexact oracles [24, 25, 75, 93], but perhaps the most complete is that of the inexact *informative, cooperative* oracle of [96]. This takes in input, besides $\boldsymbol{x}$, *three* parameters $-\infty \le \underline{\tau} \le \overline{\tau} \le \infty$ (the *lower and upper targets*, with $\overline{\tau} > -\infty$ and $\underline{\tau} < \infty$), and $0 \le \varepsilon \le \infty$ (the *accuracy*), and provides

$$\left[\begin{array}{l} \textit{function value information}\text{: } \textit{two} \text{ values } \underline{f} \text{ and } \bar{f} \text{ s.t.} \\ \qquad -\infty \le \underline{f} \le f(\boldsymbol{x}) \le \bar{f} \le \infty \quad , \quad \bar{f} - \underline{f} \le \varepsilon; \\ \text{and } \textit{at least one} \text{ between } \bar{f} \le \overline{\tau} \text{ and } \underline{f} \ge \underline{\tau} \text{ holds} \\ \textit{first-order information}\text{: if } \underline{f} > -\infty, \text{ a } \boldsymbol{z} \text{ s.t. } f(\cdot) \ge \underline{f} + \langle \boldsymbol{z}, \cdot - \boldsymbol{x} \rangle \end{array}\right. \tag{65}$$

It is always possible to attain (65), possibly at the cost of computing $f(\boldsymbol{x})$ with high accuracy, but the many parameters "allow to stop computation earlier". In particular, if $\bar{f} \le \overline{\tau}$ then it is possible to return

$\underline{f} = -\infty$, and hence *no linearization* $\boldsymbol{z}$ *at all*. This is motivated by the Lagrangian case in which (39) is *hard*, say a Mixed-Integer Linear Problem (MILP), whose solution process actually amounts at *three* different parts:

1. finding a feasible solution $\underline{\boldsymbol{u}} \in U$ (hence $\underline{f}$ and $\boldsymbol{z}$) by appropriate *heuristics*;
2. producing an upper bound $\bar{f}$ by the exact solution of some *relaxation* of (39), or a feasible solution of an appropriate dual problem;
3. if $\underline{f}$ and $\bar{f}$ are not "close enough", performing an arbitrary amount of branching and/or cutting and running 1. and 2. again.

The three parameters have different roles in stopping the process, and are not redundant. If $\varepsilon = \infty$, the thresholds $\overline{\tau}/\underline{\tau}$ may allow to stop after that step 2./1. above (respectively) have been ran, *possibly without even running the other one* (and therefore, in the case of $\overline{\tau}$, not even producing $\boldsymbol{z}$). If, instead, a finite ε is given, stopping requires both bounds, but is independent from which of the two thresholds is satisfied. It is possible to set "minimal" values for the parameters ($\overline{\tau}$ and ε as large as possible, $\underline{\tau}$ as small as possible) that ensure convergence of a IxBM, thereby hopefully reducing the computational cost of (39) as much as possible. It has to be remarked, though, that doing so may potentially impact convergence speed, a trade-off that has not been well enough investigated in practice yet.

Oracle (65) is *collaborative* in that it must in principle be able to compute the function with arbitrary accuracy, although the BM can strive to keep the requirements at a minimum. Not in all cases it is possible, or reasonable, to do so: some oracles (problems) may only be solvable up to some specific accuracy $\bar{\varepsilon}$. Actually, there are *three* different ways in which this can happen. The first is that $\bar{\varepsilon}$ is explicitly known beforehand. Otherwise, the oracle may stop with $\varepsilon < \bar{f} - \underline{f} \leq \bar{\varepsilon}$, but still produce correct upper and lower estimate. Finally, the oracle can "cheat" by (say) reporting $\bar{f} = \underline{f}$, thus formally respecting $\bar{f} - \underline{f} \leq \varepsilon$, but doing so at the cost of returning incorrect information. It turns out [96, §4] that each of the three cases corresponds to an entirely different NR, where μ^i is decreased in response to a different condition; in all these cases, convergence of the IxBM to a $\bar{\varepsilon}$-optimal solution can be proven.

5.2 Incremental approaches

Another (albeit strictly related) way in which the oracle cost can be reduced is specific to sum-functions (cf. §4.2). There, "the oracle" is actually a set of separate oracles, one for each $k \in \mathcal{K}$: in alternative/addition to allowing approximate computation in each of them separately, a rather drastic way of saving computation time is to *completely avoid to call some of them*. Hence, at each iteration one has (possibly, approximate) f-values and subgradients only for some subset $\mathcal{Z} \subseteq \mathcal{K}$ of the components, out of which the estimates $f_{\mathcal{Z}}(\boldsymbol{x}^i) = \sum_{k\in\mathcal{Z}} f_k(\boldsymbol{x}^i)$ and $\boldsymbol{z}_{\mathcal{Z}} = \sum_{k\in\mathcal{Z}} \boldsymbol{z}_k$ are obtained. A BM doing so is called *Incremental* (IcBM) by analogy with incremental subgradient-type methods [13, 41, 65]. The latter are in turn closely related with *stochastic subgradient* methods for stochastic optimization and *mini-batch* approaches in Machine Learning; there, each f_k is a specific *realization* of a stochastic process or *sample* of a process to learn, again ideally drawn at random from an infinite set. Thus, for a random $\mathcal{Z}$, there can be hope that $\boldsymbol{z}_{\mathcal{Z}}$ be a reasonable estimate of the true (stochastic) subgradient, and hence a rationale for using it to define the step. In fact, convergence for these methods is perhaps more naturally proven in a probabilistic sense; deterministic results require to compute the "full" f ($\mathcal{Z} = \mathcal{K}$)—a *batch iteration* in ML parlance—often enough. This kind of analysis is not well-suited for BM.

However, at least with the disaggregate model (cf. §4.2) it is easy enough to construct a IcBM by, basically, considering it a IxBM. Indeed, for each $k \notin \mathcal{Z}$ one can pretend that the model information at $\boldsymbol{x}^i$, say $\underline{f}_k^i(\boldsymbol{x}^i)$ and $\bar{\boldsymbol{z}}^i$, is the output of an approximate oracle; thus, it is possible to analyse IcBM using, say, the very general results of [25], as done in [31]. However, the IcBM—at least, with exact individual sub-oracles—corresponds to a *controllable* oracle, in that by evaluating more and more components it is possible to arbitrarily reduce the error; hence, one would expect to be able to do without NR. What is actually easy is declaring a NS by only evaluating a subset of all the components. In fact, since $\underline{f}_k^i(\boldsymbol{x}^i) \leq f_k(\boldsymbol{x}^i)$, clearly $\Delta f^i = \sum_{k\in\mathcal{K}}(\Delta f_k^i = f_k(\boldsymbol{x}^i) - \underline{f}_k^i(\boldsymbol{x}^i)) \geq \Delta f_{\mathcal{Z}}^i = \sum_{k\in\mathcal{Z}} \Delta f_k^i$. Hence, if $\Delta f_{\mathcal{Z}}^i > -mv^i$ implies that a fortiori (37) holds, and therefore (38) does. Declaring a SS is instead trickier, as any un-evaluated component may counterbalance the descent of all the evaluated ones with a very

steep ascent. One possible strategy is to only perform incremental NS, while requiring a "full" iteration ($\mathcal{Z} = \mathcal{K}$) to declare a SS, analogously to what incremental subgradients do. A different approach has been proposed in [96], under the assumption that all the f_k are Lipschitz continuous with *known* constant L_k. This allows to perform incremental SS as well by using the *upper model*

$$\hat{f}^k_{\mathcal{P}}(\boldsymbol{x}) = \min \left\{ \sum_{p \in \mathcal{P}_k} f^p_k \theta^p_k + L_k \|\boldsymbol{s}_k\|_2 \ : \ \sum_{p \in \mathcal{P}_k} \boldsymbol{x}^p \theta^p_k + \boldsymbol{s}_k = \boldsymbol{x} \ , \ \boldsymbol{\theta}_k \in \Theta_k \right\} \ ,$$

where $\mathcal{P}_k$ is the *upper bundle* formed of pairs $(\boldsymbol{x}^p \, , \, f^p_k = f_k(\boldsymbol{x}^p))$. The upper bundle can be compressed similarly to the ordinary (lower) one $\mathcal{B}_k$, with its poorman's version containing only $\bar{\boldsymbol{x}}$, thus making the computation of $\hat{f}^k_{\mathcal{P}}$ potentially inexpensive. With this expedient, an IcBM that need not necessarily compute all the components neither at NS nor at SS can be defined, and its convergence analysed with quite standard results, basically those of [19]. It is also easy to combine the two techniques by only computing a subset of the components *and* do that only approximately, e.g. with oracle (65).

An even stronger version of IcBM requires that the MP is not solved, as usual, for all components together, but component-wise, somehow more in the spirit of incremental subgradient methods; this entails some complications [47]. Here one could, however, employ the approach of [67] discussed in §4.5, whereby only the dual variables of the currently "active" component are allowed to vary whereas all the others are kept fixed, so that all components but one are represented by one fixed linearization.

All in all, IcBM have already been shown to improve performances in practice [31,48], but more work is required to characterize the many trade-offs they entail.

We finish this section with an apparently different, but in fact strongly related, way to decrease the function computation time: exploiting the fact that the oracles f_k are independent, and therefore can be computed *in parallel*. This can be done in the obvious master-slave fashion, which has obvious drawbacks. First, the MP is a sequential bottleneck, which by Amdahl's law limits the maximum achievable speedup [16], requiring specific efforts to decrease the MP cost (with the corresponding nontrivial trade-offs). Furthermore, subdividing the components between different processors so that the computation takes roughly the same time can be reasonably easy if all components are alike, but in many applications some of them require considerably more effort than others. Thus, a truly *asynchronous* BM would be required. A proposal in this sense is [33], which however is tailored to the case where $|\mathcal{K}|$ is large, but each component actually depends on only a few of the variables $\boldsymbol{x}$. A general-purpose asynchronous BM should be possible, in particular using the results of [96], but several theoretical and practical issues still have to be ironed out.

6 Conclusion

Bundle-type methods have now a quite long history, spanning over 40 years from [69,76,101], and almost 60 from the seminal [60]. This work shows that this time has not been wasted: motivated by the ever increasing requirements of applications, many variants have been proposed and analysed that can provide significant performance benefits. As a very quick summary, investigation has focussed on i) different forms of stabilization, with different trade-offs between the cost of the corresponding MP and the theoretical and practical convergence speed; ii) different forms of (lower, and recently also upper) models that better exploit the properties of the function at hand; iii) solution methods for the MP that provide trade-offs between the accuracy of the solution and the computational cost; iv) a detailed characterization of the accuracy with which (the different components of) f has (have) to be computed in order to be able to proceed with optimization.

Yet, several theoretical and practical issues still remain open. The understanding of efficiency of standard BM is still rather partial, with the only available results depicting the almost hopelessly slow method corresponding to full aggregation—basically, a subgradient-type one—and therefore completely failing to capture facets of the practical convergence like the "fast tail". Furthermore, almost all efficiency estimate treat NS and SS as almost entirely unrelated processes, whereas intuitively the practical efficiency of BM precisely hinges on the fact that they are not. In general, dealing with the stabilization parameter(s) remains more of an art than a science, thereby making BM rather susceptible to breaking down due to mismanagement of the algorithmic parameters; this limits their application potential, due to the difficulty of providing a "black-box" implementation that can be used by an inexperienced user without knowledge of its inner working and a significant parameter tuning phase. Besides the stabiliza-

tion and related algorithmic parameters, this also applies to the fact that there are many variants of BM regarding to the stabilization technique, the model and the computation of the function; finding the right one for one's application, and capturing the proper trade-offs between all these aspects, is a rather complex process currently requiring specific knowledge and skills. It is therefore perhaps not surprising that there are not many available BM software packages, and that their practical use in applications is rather limited in comparison to more "stable" algorithmic techniques like simplex and IP methods for linear/quadratic/conic programming. Admittedly, this also has to do with the inherent complexity of choosing, say, the right Lagrangian relaxation of one's problem, as opposed to just writing the model and using standard tools, a more general issue having more to do with the currently available modelling tools and solvers than with the specific characteristics of BM in particular.

Also, this work only deals with "standard" BM for convex problems. Significant research, often motivated by specific application like Machine Learning, has been poured into BM for nonconvex problems, or "nonstandard" ones trying to make better use of whatever available second-order information may be (if any). Thus, our treatment does not cover many other important facets of research in BM. Yet, we have hopefully shown that "standard" BM for convex optimization are a vast, diverse, and interesting class of algorithms with many relevant applications, and therefore a worthy research subject.

Acknowledgement

I'm very grateful to Wim van Ackooij, Christoph Helmberg, Welington de Oliveira, Claudia Sagastizábal, and Jerome Malick for their useful suggestions that helped me improve the contents and presentation of this work. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 773897 "plan4res".

References

[1] K.M. Anstreicher and L.A. Wolsey. Two "well-known" properties of subgradient optimization. *Mathematical Programming*, 120(1):213–220, 2009.

[2] A. Astorino, A. Frangioni, A. Fuduli, and E. Gorgone. A nonmonotone proximal bundle method with (potentially) continuous step decisions. *SIAM Journal on Optimization*, 23(3):1784–1809, 2013.

[3] A. Astorino, A. Frangioni, M. Gaudioso, and E. Gorgone. Piecewise quadratic approximations in convex numerical optimization. *SIAM Journal on Optimization*, 21(4):1418–1438, 2011.

[4] F. Babonneau, C. Beltran, A. Haurie, C. Tadonki, and J.-Ph. Vial. *Proximal-ACCPM: a versatile oracle based optimization method*. Advances in Computational Management Science. Springer, 2007.

[5] F. Babonneau and J.-Ph. Vial. ACCPM with a nonlinear constraint and an active set strategy to solve nonlinear multicommodity flow problems. *Mathematical Programming*, 120:179–210, 2009.

[6] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal. Bundle methods in stochastic optimal power management: a disaggregate approach using preconditionners. *Computation Optimization and Applications*, 20(3):227–244, 2001.

[7] L. Bahiense, N. Maculan, and C. Sagastizábal. The volume algorithm revisited: Relation with bundle methods. *Mathematical Programming*, 94(1):41–69, 2002.

[8] F. Barahona and R. Anbil. The volume algorithm: Producing primal solutions with a subgradient method. *Mathematical Programming*, 87(3):385–399, 2000.

[9] A. Belloni, A.L. Diniz, M.E. Maceira, and C. Sagastizábal. Bundle relaxation and primal recovery in unit-commitment problems. the brazilian case. *Annals of Operations Research*, 120(1-4):21–44, 2003.

[10] A. Belloni and C. Sagastizábal. Dynamic bundle methods. *Mathematical Programming*, 120(2):289–311, 2009.

[11] W. Ben-Ameur and J. Neto. Acceleration of cutting-plane and column generation algorithms: Applications to network design. *Networks*, 49(1):3–17, 2007.

[12] H. Ben Amor, J. Desrosiers, and A. Frangioni. On the choice of explicit stabilizing terms in column generation. *Discrete Applied Mathematics*, 157(6):1167–1184, 2009.

[13] D.P. Bertsekas and A. Nedić. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12(1):109–138, 2001.

[14] A. Borghetti, A. Frangioni, F. Lacalandra, and C.A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18:313–323, 2003.

[15] O. Briant, C. Lemaréchal, Ph. Meurdesoif, S. Michel, N. Perrot, and F. Vanderbeck. Comparison of bundle and classical column generation. *Mathematical Programming*, 113(2):299–344, 2008.

[16] P. Cappanera and A. Frangioni. Symmetric and asymmetric parallelization of a cost-decomposition algorithm for multi-commodity flow problems. *INFORMS Journal on Computing*, 15(4):369–384, 2003.

[17] J. Castro and J. Cuesta. Quadratic regularizations in an interior-point method for primal block-angular problems. *Mathematical Programming*, 2(130):415–445, 2011.

[18] G. Chen and M. Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, 1993.

[19] R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(2):261–275, 1993.

[20] T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design problems. *Discrete Applied Mathematics*, 112:73–99, 2001.

[21] A. Daniildis and C. Lemaréchal. On a primal-proximal heuristic in discrete optimization. *Mathematical Programming*, 104:105–128, 2005.

[22] G. d'Antonio and A. Frangioni. Convergence analysis of deflected conditional approximate subgradient methods. *SIAM Journal on Optimization*, 20(1):357–386, 2009.

[23] W. de Oliveira. Target radius methods for nonsmooth convex optimization. *Operations Research Letters*, 45:659–664, 2017.

[24] W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.

[25] W. de Oliveira, C. Sagastizábal, and C. Lemaréchal. Convex proximal bundle methods in depth: a unified analysis for inexact oracles. *Mathematical Programming*, 148:241–277, 2014.

[26] W. de Oliveira and M. Solodov. A doubly stabilized bundle method for nonsmooth convex optimization. *Mathematical Programming*, 156(1):125–159, 2016.

[27] G. Desaulniers, J. Desrosiers, and M.M. Solomon, editors. *Column Generation*. Springer, 2005.

[28] Y. Du and A. Ruszczyński. Rate of convergence of the bundle method. *Journal of Optimization Theory and Applications*, 173(3):908–922, 2017.

[29] O. du Merle, J.-L. Goffin, and J.-P. Vial. On improvements to the analytic center cutting plane method. *Computational Optimization and Applications*, 11:37–52, 1998.

[30] L. Dubost, R. Gonzalez, and C. Lemaréchal. A primal-proximal heuristic applied to french unit-commitment problem. *Mathematical Programming*, 104(1):129–151, 2005.

[31] G. Emiel and C. Sagastizábal. Incremental like bundle methods with applications to energy planning. *Computational Optimization and Applications*, 46(2):305–332, 2009.

[32] S. Feltenmark and K.C. Kiwiel. Dual applications of proximal bundle methods, including lagrangian relaxation of nonconvex problems. *SIAM Journal on Optimization*, 10(3):697–721, 2000.

[33] F. Fischer and C. Helmberg. A parallel bundle framework for asynchronous subspace optimisation of nonsmooth convex functions. *SIAM Journal on Optimization*, 24(2):795–822, 2014.

[34] A. Frangioni. Solving semidefinite quadratic problems within nonsmooth optimization algorithms. *Computers & Operations Research*, 21:1099–1118, 1996.

[35] A. Frangioni. *Dual-Ascent Methods and Multicommodity Flow Problems*. PhD thesis, TD 5/97, Dipartimento di Informatica, Università di Pisa, Pisa, Italy, 1997.

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

[37] A. Frangioni. About lagrangian methods in integer optimization. *Annals of Operations Research*, 139(1):163–193, 2005.

[38] 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.

[39] A. Frangioni and B. Gendron. 0-1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157(6):1229–1241, 2009.

[40] A. Frangioni and B. Gendron. A stabilized structured dantzig-wolfe decomposition method. *Mathematical Programming*, 104(1):45–76, 2013.

[41] A. Frangioni, B. Gendron, and E. Gorgone. On the computational efficiency of subgradient methods: a case study with lagrangian bounds. *Mathematical Programming Computation*, 9(4):573–604, 2017.

[42] A. Frangioni, C. Gentile, and F. Lacalandra. Solving unit commitment problems with general ramp contraints. *International Journal of Electrical Power and Energy Systems*, 30:316–326, 2008.

[43] A. Frangioni and E. Gorgone. Generalized bundle methods for sum-functions with "easy" components: Applications to multicommodity network design. *Mathematical Programming*, 145(1):133–161, 2014.

[44] A. Frangioni, A. Lodi, and G. Rinaldi. New approaches for optimizing over the semimetric polytope. *Mathematical Programming*, 104(2-3):375–388, 2005.

[45] A. Fuduli and M. Gaudioso. Tuning strategy for the proximity parameter in convex minimization. *Journal of Optimization Theory and Applications*, 130(1):95–112, 2006.

[46] E. Gaar. *Efficient Implementation of SDP Relaxations for the Stable Set Problem*. PhD thesis, Alpen-Adria-Universität Klagenfurt, Fakultät für Technische Wissenschaften, Klagenfurt, Austria, 2018.

[47] M. Gaudioso, G. Giallombardo, and G. Miglionico. An incremental method for solving convex finite min-max problems. *Mathematics of Operations Research*, 31:173–187, 2006.

[48] M. Gaudioso, G. Giallombardo, and G. Miglionico. On solving the lagrangian dual of integer programs via an incremental approach. *Computational Optimization and Applications*, 44:117–138, 2007.

[49] J.-L. Goffin, A. Haurie, and J.-P. Vial. Decomposition and nondifferentiable optimization with the projective algorithm. *Management Science*, 38:284–302, 1992.

[50] J. Gondzio and P. González-Brevis. A new warmstarting strategy for the primal-dual column generation method. *Mathematical Programming*, 152:113–146, 2015.

[51] J. Gondzio, P. González-Brevis, and P. Munari. New developments in the primal-dual column generation technique. *European Journal of Operational Research*, 224:41–51, 2013.

[52] M.D. Grigoriadis and L.G. Khachiyan. An exponential function reduction method for block angular convex programs. *Networks*, 26(2):59–68, 1995.

[53] C. Helmberg. A cutting plane algorithm for large scale semidefinite relaxations. In M. Grötschel, editor, *The Sharpest Cut*, MPS-SIAM Series on Optimization, pages 233–256. SIAM/MPS, 2004.

[54] C. Helmberg and K.C. Kiwiel. A spectral bundle method with bounds. *Mathematical Programming*, 93:173–194, 2002.

[55] C. Helmberg and A. Pichler. Dynamic scaling and submodel selection in bundle methods for convex optimization. Optimization Online 6180, 2017.

[56] C. Helmberg and F. Rendl. A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696, 2000.

[57] J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Number 306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag Berlin Heidelberg, 1996.

[58] T.G. Moore J. Elzinga. A central cutting plane algorithm for the convex programming problem. *Mathematical Programming*, 8:134–145, 1975.

[59] E. Karas, A. Ribeiro, C. Sagastizábal, and M. Solodov. A bundle-filter method for nonsmooth convex constrained optimization. *Mathematical Programming*, 116(1):297–320, 2009.

[60] J.E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960.

[61] K. Kiwiel. Efficiency of proximal bundle methods. *Journal of Optimization Theory and Applications*, 104(3):589–603, 2000.

[62] K.C. Kiwiel. Proximity control in bundle methods for convex nondiferentiable minimization. *Mathematical Programming*, 46(1-3):105–122, 1990.

[63] K.C. Kiwiel. Approximations in proximal bundle methods and decomposition of convex programs. *Journal of Optimization Theory and Applications*, 84:529–548, 1995.

[64] K.C. Kiwiel. A bundle bregman proximal method for convex nondifferentiable optimization. *Mathematical Programming*, 85(2):241–258, 1999.

[65] K.C. Kiwiel. Convergence of approximate and incremental subgradient methods for convex optimization. *SIAM Journal on Optimization*, 14(3):807–840, 2003.

[66] K.C. Kiwiel. A proximal bundle method with approximate subgradient linearizations. *SIAM Journal on Optimization*, 16(4):1007–1023, 2006.

[67] K.C. Kiwiel. An alternating linearization bundle method for convex optimization and nonlinear multicommodity flow problems. *Mathematical Programming*, 130(1):59–84, 2011.

[68] K.C. Kiwiel and C. Lemaréchal. An inexact bundle variant suited to column generation. *Mathematical Programming*, 118(1):177–206, 2009.

[69] C. Lemaréchal. An extension of Davidon methods to nondifferentiable problems. *Mathematical Programming Study*, 3:95–109, 1975.

[70] C. Lemaréchal, A. Nemirovskii, and Y. Nesterov. New variants of bundle methods. *Mathematical Programming*, 69(1):111–147, 1995.

[71] C. Lemaréchal, A. Ouorou, and G. Petrou. A bundle-type algorithm for routing in telecommunication data networks. *Computational Optimization and Applications*, 44:385–409, 2009.

[72] C. Lemaréchal and A. Renaud. A geometric study of duality gaps, with applications. *Mathematical Programming*, 90:399–427, 2001.

[73] C. Lemaréchal and C. Sagastizábal. Variable metric bundle methods: from conceptual to implementable forms. *Mathematical Programming*, 76(3):393–410, 1997.

[74] M. Lubin, K. Martin, and B. Sandıkçı C.G. Petra. On parallelizing dual decomposition in stochastic integer programming. *Operations Research Letters*, 41:252–258, 2013.

[75] J. Malick, W. de Oliveira, and S. Zaourar. Uncontrolled inexact information within bundle methods. *EURO Journal of Computational Optimization*, 5(1-2):5–29, 2017.

[76] R.E. Marsten, W.W. Hogan, and J.W. Blankenship. The BOXSTEP method for large-scale optimization. *Operations Research*, 23(3):389–405, 1975.

[77] R. Mifflin and C. Sagastizábal. On VU-theory for functions with primal-dual gradient structure. *SIAM Journal on Optimization*, 11(2):547–571, 2000.

[78] R. Mifflin and C. Sagastizábal. A VU-algorithm for convex minimization. *Mathematical Programming*, 104(2-3):583–608, 2005.

[79] R. Mifflin, D. Sun, and L. Qi. Quasi-newton bundle-type methods for nondifferentiable convex optimization. *SIAM Journal on Optimization*, 8(2):583–603, 1998.

[80] E.A. Nurminski. Separating planes algorithms for convex optimization. *Mathematical Programming*, 76:373–391, 1997.

[81] A. Ouorou. A proximal cutting plane method using Chebychev center for nonsmooth convex optimization. *Mathematical Programming*, 119(2):239–271, 2009.

[82] T. Parriani. *Decomposition Methods and Network Design Problems*. PhD thesis, Dottorato di Ricerca in Automatica e Ricerca Operativa, Alma Mater Studiorum - Università di Bologna, 2014.

[83] A. Pessoa, R. Sadykov, E. Uchoa, and F. Vanderbeck. Automation and combination of linear-programming based stabilization techniques in column generation. *INFORMS Journal on Computing*, to appear, 2018.

[84] M.C. Pinar and S.A. Zenios. Parallel decomposition of multicommodity network flows using a linear-quadratic penalty algorithm. *ORSA Journal on Computing*, 4(3):235–249, 1992.

[85] P.A. Rey and C. Sagastizábal. Dynamical adjustment of the prox-parameter in variable metric bundle methods. *Optimization*, 51(2):423–447, 2002.

[86] R. Sadykov and F. Vanderbeck. Column generation for extended formulations. *EURO Journal on Computational Optimization*, 1(1-2):81–115, 2013.

[87] C. Sagastizábal and M. Solodov. An infeasible bundle method for nonsmooth convex constrained optimization without a penalty function or a filter. *SIAM Journal on Optimization*, 16(1):146–169, 2005.

[88] 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*, 2:121–152, 1992.

[89] M.R. Scuzziato, E.C. Finardi, and A. Frangioni. Comparing spatial and scenario decomposition for stochastic hydrothermal unit commitment problems. *IEEE Transactions on Sustainable Energy*, to appear, 2018.

[90] S. Subramanian and H.D. Sherali. An effective deflected subgradient optimization scheme for implementing column generation for large-scale airline crew scheduling problems. *INFORMS Journal on Computing*, 20(4):565–578, 2008.

[91] W. van Ackooij. Decomposition approaches for block-structured chance-constrained programs with application to hydro-thermal unit commitment. *Mathematical Methods of Operations Research*, 80(3):227–253, 2014.

[92] W. van Ackooij, V. Berge, W. de Oliveira, and C. Sagastizábal. Probabilistic optimization via approximate p-efficient points and bundle methods. *Computers & Operations Research*, 77:177–193, 2017.

[93] W. van Ackooij and W. de Oliveira. Level bundle methods for constrained convex optimization with various oracles. *Computation Optimization and Applications*, 57(3):555–597, 2014.

[94] W. van Ackooij and W. de Oliveira. Convexity and optimization with copulæ structured probability constraints. *Optimization*, 65(7):1349–1376, 2016.

[95] W. van Ackooij, W. de Oliveira, and Y. Song. An adaptive partition-based level decomposition for solving two-stage stochastic programs with fixed recourse. *Informs Journal on Computing*, 30(1):57–70, 2018.

[96] W. van Ackooij and A. Frangioni. Incremental bundle methods using upper models. *SIAM Journal on Optimization*, 28(1):379–410, 2018.

[97] W. van Ackooij, A. Frangioni, and W. de Oliveira. Inexact stabilized Benders' decomposition approaches: with application to chance-constrained problems with finite support. *Computational Optimization And Applications*, 65(3):637–669, 2016.

[98] W. van Ackooij and J. Malick. Decomposition algorithm for large-scale two-stage unit-commitment. *Annals of Operations Research*, 238(1):587–613, 2016.

[99] W. van Ackooij and C. Sagastizábal. Constrained bundle methods for upper inexact oracles with application to joint chance constrained energy problems. *SIAM Journal on Optimization*, 24(2):733–765, 2014.

[100] C. Wolf, C.I. Fábián, A. Koberstein, and L. Suhl. Applying oracles of on-demand accuracy in two-stage stochastic programming – a computational study. *European Journal of Operational Research*, 239:437–448, 2014.

[101] P. Wolfe. A method of conjugate subgradients for minimizing nondifferentiable functions. *Mathematical programming study*, 3:143–173, 1975.