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

Revisiting Augmented Lagrangian Duals

M. Cordova · W. de Oliveira · C. Sagastizábal

the date of receipt and acceptance should be inserted later

Abstract For nonconvex optimization problems, possibly having mixed-integer variables, a convergent primal-dual solution algorithm is proposed. The approach applies a proximal bundle method to certain augmented Lagrangian dual that arises in the context of the so-called generalized augmented Lagrangians. We recast these Lagrangians into the framework of a classical Lagrangian, by means of a special reformulation of the original problem. Thanks to this insight, the methodology yields zero duality gap. Lagrangian subproblems can be solved inexactly without hindering the primal-dual convergence properties of the algorithm. Primal convergence is ensured even when the dual solution set is empty. The interest of the new method is assessed on several problems, including of unit-commitment, that arise in energy optimization. These problems are solved to optimality by solving separable Lagrangian subproblems.

Keywords Bundle methods · Augmented Lagrangian Duals · Unit Commitment

Mathematics Subject Classification (2010) 65K10 · MSC 90C06 · 49J52
MINES ParisTech

1 Introduction

We are interested in optimization problems having some equality constraints that make their solution difficult, represented by the function h below

$$\texttt{PV} := \min_{x\in X} \varphi(x) \;\text{ s.t. }\; h(x) = 0\,. \tag{1}$$

We assume that (1) has an optimal solution and, therefore, its optimal value PV is finite. Both $\varphi : \mathbb{R}^n \to \mathbb{R}$ and $h : \mathbb{R}^n \to \mathbb{R}^m$ are continuous mappings, and $\emptyset \neq X \subset \mathbb{R}^n$ is a compact set.

The formulation (1) encompasses many types of problems, including those having hard inequality constraints. Indeed, since h need not be differentiable, $g(x) \leq 0$ can be rewritten as $h(x) = \max\{g(x), 0\}$. Important class of problems covered by our setting are mixed-integer linear and quadratic programs, respectively considered in [FAS17] and [GAD20].

A powerful tool for handling problems with complicating constraints is the Lagrangian function

$$L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R} \quad \text{given by} \quad L(x,y) := \varphi(x) + \langle y, h(x)\rangle\,.$$

M. Cordova
Labplan, UFSC, Brazil E-mail: cordova.m@posgrad.ufsc.br

W. de Oliveira
MINES ParisTech, PSL – Research University, CMA – Centre de Mathématiques Appliquées, F-06902 Sophia Antipolis, France.
E-mail: welington.oliveira@mines-paristech.fr

C. Sagastizábal
IMECC, Unicamp, Brazil E-mail: sagastiz@unicamp.br

The primal problem (1) is equivalent to its min-max Lagrangian formulation, while interchanging the optimization order gives a dual problem, whose optimal value DV provides a lower bound:

$$(1) \text{ is equivalent to } \mathtt{PV} = \min_{x\in X} \sup_{y\in\mathbb{R}^m} L(x,y) \geq \sup_{y\in\mathbb{R}^m} \min_{x\in X} L(x,y) =: \mathtt{DV}\,.$$

If h is hard to deal with, the main appeal in solving the dual problem instead of (1) is decomposability, [Sag12]. When (1) has a favorable structure, for each given value y the Lagrangian subproblems $\min_{x\in X} L(x,y)$ are separable and, hence, can be solved easily by means of smaller and simpler optimization problems.

Passing through the dual problem gives a solution to (1) when the primal problem is sufficiently convex, for example, when φ is a convex function, h is a linear mapping, and the set X is convex. Since strong duality holds, maxi-minimizing the Lagrangian provides a dual optimum from which a primal optimal solution can be obtained through straightforward computations. In the absence of convexity, for example when there are 0-1 decision variables, the dual problem obtained when maxi-minimizing the Lagrangian gives the best lower bound for PV but there is a *duality gap*. In this case, heuristics are needed to recover a point satisfying the relaxed difficult constraint.

Clearly, finding formulations that eliminate the duality gap for mixed 0-1 nonconvex problems is highly desirable. To this aim, we explore under a new light the generalized Lagrangians in [RW98, Chap. 11K]. Rather than dealing with penalty terms resembling an augmented Lagrangian, our approach is purely based on classical Lagrangian relaxation. This is achieved by considering the following reformulation of (1):

$$\min_{x\in X} \varphi(x) \;\text{ s.t. }\; h(x) = 0\,,\; \text{and }\; \sigma(h(x)) \leq 0\,, \tag{1^σ}$$

where the additional scalar inequality constraint uses an *augmenting function*

$$\sigma(\cdot) : \mathbb{R}^m \to \mathbb{R} \text{is continuous, and satisfies } \min_u \sigma(u) = 0 \;\text{ and }\; \{0\} = \arg\min_u \sigma(u)\,. \tag{2}$$

The corresponding Lagrangian,

$$L : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R} \quad \text{given by} \quad L(x,y,r) := \varphi(x) + \langle y, h(x)\rangle + r\sigma(h(x))\,, \tag{3}$$

has the remarkable property of closing the duality gap. As a mathematical object, the Lagrangian (3) is not new. It is the generalized Augmented Lagrangian in [RW98, Chap. 11K] and generates dual problems called augmented Lagrangian duals in [BE15], [FAS17] and [GAD20]. The Proximal and Sharp Lagrangians in [Gas02], [NOR07], [Bur+06], [BOK19] are particular instances. Nevertheless, our interpretation of the concept as a standard Lagrangian of the reformulation $(1)^\sigma$ gives a new insight, that we exploit both theoretically and numerically. Regarding theory, we can show that dual variables for $(1)^\sigma$ exist whenever the original problem (1) admits dual variables. Regarding practice, we propose a primal-dual bundle algorithm with the well-known properties of reliability and robustness of these methods, [Bon+06].

In some sense, this work provides an answer to the concluding section in [GAD20], that mentions as future direction of research the *development of separable exact algorithms utilizing the strong duality results*. Namely, in the setting of (3), the price to pay for the gained strong duality is the loss of decomposability: the Lagrangian subproblems are no longer separable, somehow defeating the whole purpose of relaxing the hard constraint. For this reason, allowing for *inexact* solution of the Lagrangian subproblems is of fundamental practical importance. Thanks to the understanding gained with the reformulation $(1)^\sigma$, we put in place an inexact form of the proximal bundle scheme [OSL] that provides reliable approximate solutions to (1) and to the dual problem derived from (3). The method is shown to be convergent, on both its primal and dual components, noting that for the asymptotic analysis of the primal iterates we develop a new theory that goes beyond the usual bundle mechanism.

The remainder of this work is organized as follows. In addition to showing existence of dual variables for $(1)^\sigma$, Section 2 characterizes approximate primal and dual solutions and the relation between (1) and (7) when the minima of Lagrangian subproblems are inexact. We show that approximate strong duality depends on certain "σ-simplex", defined in (9). We then turn to the solution method, laying out the primal-dual bundle approach in Sections 3 and 4, respectively considering exact and inexact calculations in the Lagrangian subproblems. Each of these sections includes numerical illustrations. Section 4 also presents the convergence analysis. The successful solution of unit-commitment problems, using separable Lagrangian subproblems, is the topic of Section 5. Section 6 closes this paper with some concluding comments. A final appendix has tables for the experiments in Sections 3, 4, and 5.

2 Primal and dual properties of the reformulation

Our first concern refers to existence of dual variables when the problem is augmented with one constraint. We show that if the original primal problem (1) has optimal multipliers, so does the reformulation $(1)^\sigma$. We then characterize strong duality both in an exact and approximate manners. Allowing for approximations in the theory is fundamental to tackle the inexact solution of Lagrangian subproblems in practice, typically inducing separability in (3) via approximations. The unit-commitment problems in Section 5 illustrate well the interest of such an approach.

2.1 Existence of dual variables

To simplify the developments in the proof of Proposition 1 below, we assume that $h : \mathbb{R}^n \to \mathbb{R}^m$ is a smooth mapping and $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function. The latter ensures that the Clarke subdifferential of φ is well defined for all $x \in \mathbb{R}^n$:

$$\partial^{\mathtt{C}}\varphi(x) := \mathtt{Conv}\Big\{ \lim_{\ell\to\infty} \nabla\varphi(x^\ell), x^\ell \to x,\ \varphi \text{ differentiable at } x^\ell \Big\}.$$

Proposition 1 (Multiplier Existence) *Consider problem* (1) *and suppose that $h : \mathbb{R}^n \to \mathbb{R}^m$ is a smooth mapping, X is a convex closed set, and $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a locally Lipschitz function. Let $\bar{y} \in \mathbb{R}^m$ be a vector of Lagrange multipliers associated to the constraint $h(x) = 0$, so that the primal-dual pair $(\bar{x}, \bar{y})$ satisfies the following (generalized) KKT conditions of problem* (1)*:*

$$\begin{cases} 0 \in \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top \bar{y} + N_X(\bar{x}) \\ 0 = h(\bar{x}),\ \bar{x} \in X, \end{cases} \tag{5}$$

where $Jh(\bar{x}) \in \mathbb{R}^{n\times m}$ is the Jacobian matrix of h at $\bar{x}$, and $N_X(\bar{x})$ is the cone normal to X at $\bar{x}$. Then, for all $\bar{r} \geq 0$, the triple $(\bar{x}, \bar{y}, \bar{r})$ is a KKT point of the augmented problem (1^σ)*:*

$$\begin{cases} 0 \in \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top \bar{y} + \bar{r}\, Jh(\bar{x})^\top \partial\sigma(h(\bar{x})) + N_X(\bar{x}) \\ 0 = h(\bar{x}),\ \bar{x} \in X \\ 0 \geq \sigma(h(\bar{x})) \\ 0 = \bar{r}\,\sigma(h(\bar{x})),\ \bar{r} \geq 0. \end{cases} \tag{6}$$

Proof Let $\bar{r} \geq 0$ be arbitrary. Except for the first inclusion in (6), all the other conditions are automatically satisfied due to (5) and assumptions in (2). Note that the latter implies $0 \in \partial\sigma(0)$. Then, starting from the first inclusion in (5), the following developments hold:

$$\begin{aligned} 0 &\in \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top \bar{y} + N_X(\bar{x}) && \text{(due to (5))} \\ &= \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top [\bar{y} + \bar{r}0] + N_X(\bar{x}) \\ &\subset \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top [\bar{y} + \bar{r}\partial\sigma(h(\bar{x}))] + N_X(\bar{x}) && \text{(because } 0 \in \partial\sigma(0) \text{ and } h(\bar{x}) = 0) \\ &= \partial^{\mathtt{C}}\varphi(\bar{x}) + Jh(\bar{x})^\top \bar{y} + \bar{r}\, Jh(\bar{x})^\top \partial\sigma(h(\bar{x})) + N_X(\bar{x}). \square \end{aligned}$$

The arguments in Proposition 1 make it clear that smoothness of h is not essential, but rather convenient to write down simple optimality conditions. The important message is that, as long as some constraint qualification for (1) ensures the existence of KKT points, a similar property can be derived for $(1)^\sigma$, thanks to the properties required for the augmenting function in (2).

2.2 Characterization of strong duality

We now turn our attention to the relation between the reformulation and its dual problem, obtained when maxi-minimizing the Lagrangian (3):

$$\mathtt{DV}^\sigma := \sup_{y\in\mathbb{R}^m, r\in\mathbb{R}_+} \min_{x\in X} L(x,y,r) = \sup_{(y,r)\in\mathbb{R}^{m+1}} \min_{x\in X} L(x,y,r)\,. \tag{7}$$

The rightmost equality, defining the dual as an unconstrained problem, follows from nonnegativity of $\sigma(\cdot)$, that makes the Lagrangian function increasing on variable r.

By weak duality, with any Lagrangian the inequality $\texttt{PV} \geq \texttt{DV}^\sigma$ holds. Here arises a very special feature of the Lagrangian (3): regardless nonconvexity, there is no optimality gap, thanks to the requirement that $\sigma(0) = 0$ in (2). Such result is given as a corollary of the following version of [Eve63, Theorem 3], extended to our setting. As in the original result by Everett, only mild assumptions are required, in particular, neither differentiability nor convexity is assumed for (1^σ).

Theorem 1 (Inexact Everett's) *For the augmented problem* (1^σ)*, given* $(y,r) \in \mathbb{R}^m \times \mathbb{R}_+$ *and an accuracy* $\varepsilon \geq 0$*, consider the set of approximate minimizers of the Lagrangian function* (3)*:*

$$X_\varepsilon(y,r) := \{x_\varepsilon \in X \quad \textit{satisfying} \quad L(x_\varepsilon,y,r) \leq L(x,y,r) + \varepsilon \quad \textit{for all } x \in X\}\,. \tag{8}$$

The following holds for any $(\tilde{y},\tilde{r}) \in \mathbb{R}^m \times \mathbb{R}_+$.

(i) The point $\tilde{x}_\varepsilon \in X_\varepsilon(\tilde{y},\tilde{r})$ *minimizes within* ε *the primal function* $\varphi(x)$ *over the perturbed feasible set*

$$\left\{ x \in X \;\middle|\; \begin{array}{ll} h_j(x) \leq h_j(\tilde{x}_\varepsilon) & \textit{if } \tilde{y}_j > 0 \\ h_j(x) \geq h_j(\tilde{x}_\varepsilon) & \textit{if } \tilde{y}_j \leq 0 \\ \sigma(h(x)) \leq \sigma(h(\tilde{x}_\varepsilon)) & \end{array} \right\}.$$

(ii) The point $\tilde{x}_\varepsilon \in X_\varepsilon(\tilde{y},\tilde{r})$ *is an* ε*-solution to the primal problem* (1) *if and only if it is primal feasible:*

$$\left.\begin{array}{c} \tilde{x}_\varepsilon \in X_\varepsilon(\tilde{y},\tilde{r}) \textit{ satisfies } \varphi(\tilde{x}_\varepsilon) \leq \varphi(x) + \varepsilon \\ \textit{for all } x \in X \textit{ such that } h(x) = 0 \end{array}\right\} \textit{ if and only if } \left\{\begin{array}{l} \tilde{x}_\varepsilon \in X_\varepsilon(\tilde{y},\tilde{r}) \\ \textit{and } h(\tilde{x}_\varepsilon) = 0. \end{array}\right.$$

Proof For all $x \in X$ the following inequality follows from the definition of the set $X_\varepsilon(\tilde{y},\tilde{r}) \ni \tilde{x}_\varepsilon$:

$$\varphi(\tilde{x}_\varepsilon) \leq \varphi(x) + \langle \tilde{y}, h(x) - h(\tilde{x}_\varepsilon)\rangle + \tilde{r}[\sigma(h(x)) - \sigma(h(\tilde{x}_\varepsilon))] + \varepsilon\,.$$

In particular, for $\tilde{x}$ in the perturbed feasible set (which is contained in X), by the assumptions for σ in (2), we see that

$$\varphi(\tilde{x}_\varepsilon) \leq \varphi(\tilde{x}) + \langle \tilde{y}, h(\tilde{x}) - h(\tilde{x}_\varepsilon)\rangle + \tilde{r}[\sigma(h(\tilde{x})) - \sigma(h(\tilde{x}_\varepsilon))] + \varepsilon \leq \varphi(\tilde{x}) + \varepsilon\,,$$

showing item (i). Next, notice that if $h(\tilde{x}_\varepsilon) = 0$, the perturbed feasible set in item (i) reduces to $\{x \in X : h(x) = 0\}$, again by (2), so $\tilde{x}_\varepsilon$ approximately solves (1), as stated. The converse relation in item (ii) is straightforward, and this concludes the proof. □

The characterization in item (ii) of Everett's theorem directly related to the requirements in (2) and to the fact that, unlike the well-known Hestenes and Powell Augmented Lagrangian, the scalar r is not considered a fixed penalty but a dual variable. When calculations are exact, in particular, Everett's theorem ensures there is no optimality gap.

Corollary 1 (Zero Duality Gap) *Suppose in Theorem 1 calculations are exact: in* (8) *the error* $\varepsilon = 0$*. Then*

$$\tilde{x}_0 \in X_0(\tilde{y},\tilde{r}) \textit{ satisfies } h(\tilde{x}_0) = 0 \iff \textit{there is no duality gap: } \texttt{PV} = \texttt{DV}^\sigma\,,$$

in which case $\tilde{x}_0$ *solves* (1) *and the pair* $(\tilde{y},\tilde{r})$ *solves* (7)*.*

Proof By Theorem 1(ii), the point $\tilde{x}_0$ solves (1), and by weak duality, $\varphi(\tilde{x}_0) = \texttt{PV} \geq \texttt{DV}^\sigma$. Combining primal feasibility of $\tilde{x}_0$ with the definition of the set $X_0(\tilde{y},\tilde{r})$ in (8) yields $\varphi(\tilde{x}_0) = L(x_\varepsilon,\tilde{y},\tilde{r}) \leq L(x,\tilde{y},\tilde{r})$, for all $x \in X$. Since the right-hand side above is smaller than $\texttt{DV}^\sigma$, the result follows. □

Zero duality gap is shown in [RW98, Theorem 11.59] using generalized augmented Lagrangians for the original problem (1); see also [Bur11, Lemma 2.1(c)]. Theorem 1 and Corollary 1 establish those key results in a much simpler manner, working directly with the classical Lagrangian function of the augmented problem (1^σ).

2.3 Characterization of approximate strong duality

We let $\Delta^{m+2} = \left\{(\alpha_1,\ldots,\alpha_{m+2}) \geq 0 : \sum_{i=1}^{m+2} \alpha_i = 1\right\}$, the unit simplex in $\mathbb{R}^{m+2}$, and consider the following σ-*simplex*, depending on given (ε, y, r),

$$\Delta_\varepsilon^\sigma(y,r) := \left\{ \begin{array}{ll} -\sum_{i=1}^{m+2} \alpha_i \sigma(h(x^i)) \in \mathbb{R} & \text{for some } x^1,\ldots,x^{m+2} \text{ and } \alpha \in \Delta^{m+2} \text{ such that} \\ & \sum_{i=1}^{m+2} \alpha_i L(x^i,\tilde{y},\tilde{r}) \leq \min_{x\in X} L(x,\tilde{y},\tilde{r}) + \varepsilon \end{array} \right\}. \tag{9}$$

By continuity of the functions in (3), this is an outer semicontinuous multifunction of (ε, y, r):

$$\left\{ \begin{array}{l} \text{given any sequence } \{\varepsilon_k, y^k, r^k\} \to (\bar{\varepsilon}, \bar{y}, \bar{r}) \\ \text{for each selection } \left\{ s^k = -\sum_{i=1}^{m+2} \alpha_i^k \sigma(h(x^{ik})) \ \in \Delta_{\varepsilon_k}(y^k, r^k) \right\} \to \bar{s} = -\sum_{i=1}^{m+2} \bar{\alpha}_i \sigma(h(\bar{x}^i)) \text{ as } k \to \infty \\ \text{it holds that } \bar{s} \in \Delta_{\bar{\varepsilon}}(\bar{y},\bar{r}). \end{array} \right.$$

We shall use this property in our convergence analysis in Section 4. Now, we study the role played by the σ-simplex in identifying approximate primal and dual optimality.

Theorem 2 (Characterization of ε-strong duality) *For any accuracy $\varepsilon \geq 0$ and $(\tilde{y},\tilde{r}) \in \mathbb{R}^m \times \mathbb{R}_+$, the following statements are equivalent.*

(i) The pair $(\tilde{y},\tilde{r})$ solves (7) *up to ε.*

(ii) There exist $x^1,\ldots,x^{m+2} \in X$ such that $\texttt{PV} \leq \sum_{i=1}^{m+2} \alpha_i \varphi(x^i) \leq \texttt{DV}^\sigma + \varepsilon$.

Among such points, $x^{i_{\min+}}$ solves (1) *up to ε, where we defined the index*

$$i_{\min+} := \arg \min_{i=1,\ldots,m+2} \{\varphi(x^i) : \alpha_i > 0\}.$$

(iii) The σ-simplex from (9) *contains the null vector: the inclusion $0 \in \Delta_\varepsilon^\sigma(\tilde{y},\tilde{r})$ holds.*

Proof Consider the convex function that results from taking the negative of the objective function in the dual problem (7),

$$f(y,r) := \max_{x\in X} -L(x,y,r),$$

and notice that $-\texttt{DV}^\sigma \leq f(y,r)$ for all (y,r). Showing approximate optimality of $(\tilde{y},\tilde{r})$ in (7) is equivalent to having $0 \in \partial_\varepsilon f(\tilde{y},\tilde{r})$, the ε-subdifferential of Convex Analysis, [HL96, Chapter XI]. The expression for this subdifferential is obtained by applying [SNH83, Theorem 4.2] to $(1)^\sigma$, making the identifications

[SNH83]	t	T	x	$\bar{x}$	f_t
in this proof	x	X	(y,r)	$(\tilde{y},\tilde{r})$	$f_x(y,r) = -\varphi(x) - \langle y, h(x)\rangle - r\sigma(h(x))$

The theorem states that, for compact X,

$$\begin{array}{c} (y^*,r^*) \in \partial_\varepsilon f(\tilde{y},\tilde{r}) \quad \text{if and only if} \\ \exists x^1,\ldots,x^{m+2} \in X, \quad \alpha \in \Delta^{m+2}, \quad \text{and} \quad \varepsilon_1,\ldots,\varepsilon_{m+1} \geq 0 \text{ such that} \\ (y^*,r^*) \in \sum_{i=1}^{m+2} \partial_{\varepsilon_i}\Big(\alpha_i f_{x^i}\Big)(\tilde{y},\tilde{r}) \text{ for } 0 \leq f(\tilde{y},\tilde{r}) - \sum_{i=1}^{m+2} \alpha_i f_{x^i}(\tilde{y},\tilde{r}) \leq \varepsilon - \sum_{i=1}^{m+2} \varepsilon_i. \end{array} \tag{10}$$

Each function $\alpha_i f_{x^i}$ is affine on (y,r), so the approximate subdifferential is a singleton

$$\forall \eta \geq 0 \qquad \partial_\eta\Big(\alpha_i f_{x^i}\Big)(\tilde{y},\tilde{r}) = -\alpha_i\Big(h(x^i), \sigma(h(x^i))\Big).$$

The role of ε_i in the formula from [SNH83] becomes vacuous, and we can set those values to 0:

$$\partial_\varepsilon f(\tilde{y},\tilde{r}) = \left\{ -\sum_{i=1}^{m+2} \alpha_i \Big(h(x^i), \sigma(h(x^i)) \Big) \mid (x^i,\alpha_i) \text{ from (10) satisfy } f(\tilde{y},\tilde{r}) \le \sum_{i=1}^{m+2} \alpha_i f_{x^i}(\tilde{y},\tilde{r}) + \varepsilon \right\}.$$

Rewriting the right most relation above using the Lagrangian yields the inequality in (9):

$$-\min_{x\in X} L(x,\tilde{y},\tilde{r}) = f(\tilde{y},\tilde{r}) \le -\sum_{i=1}^{m+2} \alpha_i f_{x^i}(\tilde{y},\tilde{r}) + \varepsilon = -\sum_{i=1}^{m+2} \alpha_i L(x^i,\tilde{y},\tilde{r}) + \varepsilon\,.$$

Therefore, $(y^*,r^*) \in \partial_\varepsilon f(\tilde{y},\tilde{r}) \Longrightarrow r^* \in \Delta_\varepsilon^\sigma(\tilde{y},\tilde{r})$. Additionally, recalling from (2) that σ is positive except at the origin, where it is null, if $r^* = 0$ then $y^* = 0$. It follows that

$$0 \in \partial_\varepsilon f(\tilde{y},\tilde{r}) \iff 0 \in \Delta_\varepsilon^\sigma(\tilde{y},\tilde{r})\,.$$

The arguments for showing the chain of statements go as follows. Approximate dual optimality holds if and only if $0 \in \partial_\varepsilon f(\tilde{y},\tilde{r})$, so (i) and (iii) are equivalent. Item (iii) holds if and only if

$$r^* = 0 = -\sum_{i=1}^{m+2} \alpha_i \sigma(h(x^i)) \quad \text{with} \quad \sum_{i=1}^{m+2} \alpha_i L(x^i,\tilde{y},\tilde{r}) \le \min_{x\in X} L(x,\tilde{y},\tilde{r}) + \varepsilon\,.$$

The left identity is equivalent to each term positive coefficient yielding a primal feasible point:

$$h(x^i) = 0 \quad \text{for all } i \text{ such that } \alpha_i > 0\,.$$

For such points $L(x^i,\tilde{y},\tilde{r}) = \varphi(x^i) \ge \texttt{PV}$ and the first assertion in item (ii) follows from weak duality. For the special index $i_{\min+}$, using that $\alpha \in \Delta^{m+2}$ below, we see that

$$\varphi(x^{i_{\min+}}) = L(x^{i_{\min+}},\tilde{y},\tilde{r}) \le \sum_{i=1}^{m+2} \alpha_i L(x^i,\tilde{y},\tilde{r}) \le \min_{x\in X} L(x,\tilde{y},\tilde{r}) + \varepsilon\,,$$

we conclude that $x^{i_{\min+}} \in X_\varepsilon(\tilde{y},\tilde{r})$ and Theorem 1(ii) gives approximate primal optimality. □

Some results in the literature are special cases of Theorem 2, written with $\varepsilon = 0$. When $\texttt{DV}^\sigma = \texttt{PV}$ and $0 \in \Delta_0^\sigma(\tilde{y},\tilde{r})$ for some $\tilde{r}$, the vector $\tilde{y}$ *supports an exact penalty representation* for problem (1), [RW98, Definition 11.60]. Specializing Theorem 2, to $\varepsilon = 0$ gives a characterization of the property like the one in [Bur11, Theorem 2.2].

The results above justify the necessity of putting in place a nonsmooth optimization method to solve the dual problem (7). This is the topic addressed in the next sections. Our primal-dual bundle algorithm generates a subsequence of special iterates $(\hat{y}^k,\hat{r}^k)$ satisfying asymptotically condition (iii) in Theorem 2. In the bundle terminology, we shall see that those special points are related the serious-step iterations.

3 Exact bundle method for maxi-minimization

By Corollary 1, zeroing the duality gap between the augmented dual and (1) is achieved when the Lagrangian (3) is *globally* minimized over X (i.e. $\tilde{x}_0 \in X_\varepsilon(\tilde{y},\tilde{r})$ with $\varepsilon = 0$), and its global solution is primal feasible (i.e. $h(\tilde{x}_0) = 0$). While the first task is difficult in general, the second one can be efficiently addressed with by a suitable application of the proximal bundle method to solve (7). In the algorithm, it is possible to have inexact calculations (allowing $\varepsilon > 0$ in (8)), thus somehow weakening the necessity of global solvers. To introduce gradually the difficulties, we present first a conceptual scheme, with $\varepsilon = 0$, that eases the understanding of the main ingredients of the approach. The implementable version is given in Section 4.

3.1 Minimizing exactly the Lagrangian subproblems

In the conceptual scheme, for every given pair of dual vectors (y,r) its primal counterpart x_0 is defined as in (8), written with $\varepsilon = 0$. Recall that this means that some *global solver* for the Lagrangian subproblems yields the *exact* value of the dual function.

Assumption 3 (Global exact solution of Lagrangian subproblems: $\varepsilon = 0$ in (8)) *An exact procedure is available for* (1) *such that, when applied to the Lagrangian subproblem, delivers x_0 as its global minimizer. Such a property is represented on the primal-dual pairs by the relation*

$$x_0 \in X_0(y,r) \quad \textit{satisfies} \quad L(\tilde{x}_\varepsilon, y, r) \leq L(x,y,r) \textit{ for all } x \in X \,. \square \tag{11}$$

The following instance of the polyhedral Difference-of-Convex (DC) programs in [Tuy16] is a very special case, where an exact global solver is available.

3.1.1 Example: a particular class of DC optimization problems

Consider the following nonsmooth nonconvex optimization problem fitting (1)

$$\min_{x\in X} \frac{1}{2}\langle Qx,x\rangle + \langle q,x\rangle - \max_{i\in\{1,\ldots,N\}} \{\langle \alpha_i,x\rangle + \beta_i\} \quad \text{s.t.} \quad Ax = b\,, \tag{12}$$

where $Q \in \mathbb{R}^{n\times n}$ is a positive semidefinite matrix and $(\alpha_i,\beta_i) \in \mathbb{R}^{n+1}$ are given coefficients of the affine functions defining the maximum.

Since in (12) the polyhedral structure of the second component of the DC function is *known*, a global solution $\bar{x}$ for (12) can be obtained by solving N subproblems. In this case, it suffices to proceed by enumeration, computing

$$\bar{x}^i \in \arg\min_{x\in X} \frac{1}{2}\langle Qx,x\rangle + \langle q-\alpha_i,x\rangle - \beta_i \quad \text{s.t.} \quad Ax = b\,, \tag{13}$$

for all $i = 1,\ldots,N$, and setting $\bar{x} := \bar{x}^{i^*}$, with $i^* := \arg\min_{i\in\{1,\ldots,N\}}\{\frac{1}{2}\langle Q\bar{x}^i,\bar{x}^i\rangle + \langle q-\alpha_i,\bar{x}^i\rangle - \beta_i\}$.

However, depending on the problem dimension n and on the number N of affine functions, solving N quadratic programming (QP) problems as above can become prohibitive in terms of computational burden, rendering the enumerative strategy inapplicable.

For the proximal Lagrangian ($\sigma(\cdot) = \frac{1}{2}\|\cdot\|^2$ in (3)), so that

$$L(x,y,r) = \frac{1}{2}\langle Qx,x\rangle + \langle q,x\rangle - \max_{i\in\{1,\ldots,N\}} \{\langle \alpha_i,x\rangle + \beta_i\} + \langle y,Ax-b\rangle + \frac{r}{2}\|Ax-b\|^2\,.$$

Thanks to the special structure of the objective function, globally solving the Lagrangian subproblem for given dual variables (y,r) boils down to solving N separate optimization problems:

$$\min_{x\in X} L(x,y,r) = \min_{i\in\{1,\ldots,N\}} \left\{ \min_{x\in X} \frac{1}{2}\langle Qx,x\rangle + \langle q-\alpha_i,x\rangle - \beta_i + \langle y,Ax-b\rangle + \frac{r}{2}\|Ax-b\|^2 \right\}.$$

These are convex QP problems, but unlike (13), there is no constraint $Ax = b$, only $x \in X$. We discuss two exact procedures that can be devised, if X is unconstrained or an orthant.

Case 1: the set X is the full space. When $X = \mathbb{R}^n$, solving the linear system

$$(Q + rA^\top A)x = \alpha_i - q + A^\top(rb - y) \tag{14}$$

gives a global solution. Furthermore, since only the right-hand side vector α_i varies with i, just one LU decomposition of the (symmetric positive-definite) matrix $(Q + rA^\top A)$ is necessary for solving all the N linear systems and computing the exact global minimum of $\min_{x\in\mathbb{R}^n} L(x,y,r)$. In Matlab, we designed such a procedure by using the Matlab's function `linsolve`.

Case 2: the set X is the positive orthant. When $x \geq 0$, the optimality conditions

$$\begin{cases} w = (Q + rA^\top A)x - [\alpha_i - q + A^\top(rb - y)] \\ 0 \leq w \perp x \geq 0\,, \end{cases}$$

define a *Linear Complementary Problem* - LCP. Since $(Q + rA^\top A) \in \mathbb{R}^{n\times n}$ is a symmetric positive definite matrix, solving the LCP by specialized methods can be more efficient than directly applying QP solvers. Once again, only the vector α_i changes for the different subproblems, allowing for warm-starts is possible, giving rise to another procedure satisfying Assumption 3. □

3.2 The conceptual bundle scheme

Although we have given two examples of exact procedures satisfying Assumption 3, it is important to keep in mind that supposing that a global solver is available is a very strong requirement for most problems. This is why Algorithm 1 below is referred to as a *conceptual* scheme.

A practical algorithm requiring (8) rather than (11), i.e., allowing the inexact minimization of the Lagrangian subproblems, is considered in Section 4. The interest of Algorithm 1 is mostly in presenting the main ingredients of the primal-dual bundle methodology. Additionally, notice that although highly unlikely to hold exactly, the stopping test of the conceptual Algorithm 1 is a valid optimality certificate: by (2), this means that $h(x^{k+1}) = 0$ and Corollary 1 ensures that (y^{k+1}, r^{k+1}) solves (7) and x^{k+1} solves (1). In practice, the algorithm stops when $\sigma(h(x^{k+1})) \leq \texttt{Tol}$, for some tolerance $\texttt{Tol} > 0$.

Algorithm 1 A CONCEPTUAL PRIMAL-DUAL SCHEME FOR GLOBALLY SOLVING (1) UNDER ASSUMPTION 3

Input and starting point. Choose $(y^1, r^1) \in \mathbb{R}^m \times \mathbb{R}_+$ and let $x^1 \in X_0(y^1, r^1)$.
Parameters. $m \in (0,1)$ and $t_{\mathbf{y}}{}^1 \geq t_{\mathbf{y}\min} > 0$, $t_{\mathbf{r}}{}^1 \geq t_{\mathbf{r}\min} > 0$.
Step 0: Initialization. Set $(\hat{x}^1, \hat{y}^1, \hat{r}^1) := (x^1, y^1, r^1)$, $\mathscr{B}_1 := \{1\}$ and $k = 1$.
Step 1: Dual Iterate. Define (y^{k+1}, r^{k+1}) as the unique solution of

$$\max_{(y,r)\in\mathbb{R}^{m+1}} \left\{ \min_{j\in\mathscr{B}^k} L(x^j, y, r) - \frac{1}{2t_{\mathbf{y}}{}^k}\|y - \hat{y}^k\|_2^2 - \frac{1}{2t_{\mathbf{r}}{}^k}(r - \hat{r}^k)^2 \right\}. \tag{15}$$

Compute the predicted increase as

$$\delta^k := \min_{j\in\mathscr{B}^k} L(x^j, y^{k+1}, r^{k+1}) - L(\hat{x}^k, \hat{y}^k, \hat{r}^k). \tag{16}$$

Step 2: Exact Primal Iterate. Find $x^{k+1} \in X_0(y^{k+1}, r^{k+1})$.
Step 3: Stopping Test. If $\sigma(h(x^{k+1})) = 0$ stop: x^{k+1} globally solves (1) and (y^{k+1}, r^{k+1}) solves (7).
Step 4: Monotonicity Test. Declare the iteration *serious* if the following inequality holds

$$L(x^{k+1}, y^{k+1}, r^{k+1}) \geq L(\hat{x}^k, \hat{y}^k, \hat{r}^k) + m\,\delta^k\,. \tag{17}$$

Set $(\hat{x}^{k+1}, \hat{y}^{k+1}, \hat{r}^{k+1}) := (x^{k+1}, y^{k+1}, r^{k+1})$ and choose $t_{\mathbf{y}}{}^{k+1} \geq t_{\mathbf{y}}{}^k$ and $t_{\mathbf{r}}{}^{k+1} \geq t_{\mathbf{r}}{}^k$.
Otherwise, the iteration is declared *null*:
Set $(\hat{x}^{k+1}, \hat{y}^{k+1}, \hat{r}^{k+1}) := (\hat{x}^k, \hat{y}^k, \hat{r}^k)$ and choose $t_{\mathbf{y}}{}^{k+1} \in (0, t_{\mathbf{y}}{}^k]$ and $t_{\mathbf{r}}{}^{k+1} \in (0, t_{\mathbf{r}}{}^k]$.
Step 5: Loop. Set $\mathscr{B}^{k+1} = \{i \in \mathscr{B}^k : L(x^i, y^{k+1}, r^{k+1}) = \min_{j\in\mathscr{B}^k} L(x^j, y^{k+1}, r^{k+1})\} \cup \{k+1\}$, $k := k+1$ and go back to Step 1.

The *bundle* of information $\mathscr{B}^{k+1}$ is updated in Step 5 so that, in addition to the new information, only active indices (j achieving the minimum $L(x^j, y^{k+1}, r^{k+1})$ are kept. The ability of dropping inactive indices (or "cuts" in a cutting-plane jargon) without impairing convergence is a strength of the bundle methodology. The feature is important for keeping manageable the size of the QP problems (15), as k increases. This point becomes clear when working with the dual formulation (19) given below, a QP problem over the unit simplex of dimension $|\mathscr{B}_k|$.

Proposition 2 (Alternative computation of dual iterates and aggregate elements) *For a given iteration k, the dual iterate in Step 1 of Algorithm 1 satisfies the relation*

$$\begin{aligned} y^{k+1} &= \hat{y}^k + t_{\mathrm{y}}{}^k G_{\mathrm{y}}{}^k \quad \text{for} \quad G_{\mathrm{y}}{}^k := \sum_{j\in\mathscr{B}^k} \alpha_j^k h(x^j) \\ r^{k+1} &= \hat{r}^k + t_{\mathrm{r}}{}^k G_{\mathrm{r}}{}^k \quad \text{for} \quad G_{\mathrm{r}}{}^k := \sum_{j\in\mathscr{B}^k} \alpha_j^k \sigma(h(x^j)) \end{aligned} \tag{18}$$

where the coefficients α_j^k can be computed in any of the two ways below.

(i) Either by considering the following QP problem, equivalent to (15)

$$\begin{cases} \max\limits_{y,r,w} \; w - \frac{1}{2t_{\mathrm{y}}{}^k}\|y-\hat{y}^k\|_2^2 - \frac{1}{2t_{\mathrm{y}}{}^k}(r-\hat{r}^k)^2 \\ s.t. \quad w \le L(x^j,y,r), \quad \text{for all } j\in\mathscr{B}^k \\ \qquad y\in\mathbb{R}^m,\; r\in\mathbb{R},\; w\in\mathbb{R}, \end{cases}$$

in which case $\alpha_j^k \ge 0$ is the Lagrange multiplier associated with the constraint $w \le L(x^j,y,r)$ at the solution. Or as a solution α^k of the dual QP problem

$$\min_{\alpha\in\Delta^{|\mathscr{B}^k|}} \quad \frac{t_{\mathrm{y}}{}^k}{2}\alpha^\top \Gamma\alpha + \frac{t_{\mathrm{r}}{}^k}{2}\alpha^\top\Gamma^\sigma\alpha + \gamma^\top\alpha, \tag{19}$$

where we defined the Gram matrices Γ and $\Gamma^\sigma \in \mathbb{R}^{|\mathscr{B}^k|\times|\mathscr{B}^k|}$ with entries

$$\Gamma_{i,j} := \langle h(x^i), h(x^j)\rangle \text{ and } \Gamma^\sigma_{i,j} := \sigma(h(x^i))\sigma(h(x^j)) \text{ for all } i,j\in\mathscr{B}^k,$$

and the vector $\gamma\in\mathbb{R}^{|\mathscr{B}^k|}$ with components $\gamma_j := L(x^j,\hat{y}^k,\hat{r}^k)$ for all $j\in\mathscr{B}^k$.

(ii) The aggregate error and the aggregate gap

$$e^k := \sum_{j\in\mathscr{B}^k}\alpha_j^k L(x^j,\hat{y}^k,\hat{r}^k) - L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \quad \text{and} \quad \phi^k := \sum_{j\in\mathscr{B}^k}\alpha_j^k\varphi(x^j) - L(\hat{x}^k,\hat{y}^k,\hat{r}^k)$$

satisfy the relations

$$\phi^k = e^k - \langle G_{\mathrm{y}}{}^k,\hat{y}^k\rangle - \langle G_{\mathrm{r}}{}^k,\hat{r}^k\rangle \quad \text{and} \quad \delta^k = t_{\mathrm{y}}{}^k\|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k\|G_{\mathrm{r}}{}^k\|^2 + e^k.$$

(iii) The partial aggregate gradient $G_{\mathrm{r}}{}^k$ defined in (18) *belongs to the σ-simplex from* (9)*:*

$$-G_{\mathrm{r}}{}^k \in \Delta^\sigma_{\mathrm{E}^k}(\hat{y}^k,\hat{r}^k) \text{ for } \mathrm{E}^k := \sum_{j\in\mathscr{B}^k}\alpha_j^k L(x^j,\hat{y}^k,\hat{r}^k) - \min_{x\in X} L(x^j,\hat{y}^k,\hat{r}^k) \ge 0.$$

Proof The last item is just a combination of the definitions of $G_{\mathrm{r}}{}^k$ and the σ-simplex in (9), with E^k replacing ε. Similarly for the expression for ϕ^k in item (iii), combining the definitions of the aggregate error and of the Lagrangian (3). For the other statements we shorten y^{k+1} to y^+ and similarly for r^{k+1}. Denote by $M(y,r) := \min_{j\in\mathscr{B}^k} L(x^j,y,r)$ the concave model function defining the first term in the maximand of (15). By standard duality arguments, (15) and (19) are equivalent, and their optimal values coincide, so

$$M(y^+,r^+) - \frac{1}{2t_{\mathrm{y}}{}^k}\|y^+-\hat{y}^k\|_2^2 - \frac{1}{2t_{\mathrm{y}}{}^k}(r^+-\hat{r}^k)^2 = \frac{1}{2}t_{\mathrm{y}}{}^k\|G_{\mathrm{y}}{}^k\|^2 + \frac{1}{2}t_{\mathrm{r}}{}^k\|G_{\mathrm{r}}{}^k\|^2 + \gamma^\top\alpha^k,$$

by the definition of the Gram matrices. Since in (16) we have that $\delta^k = M(y^+,r^+) - L(\hat{x}^k,\hat{y}^k,\hat{r}^k)$,

$$\delta^k = t_{\mathrm{y}}{}^k\|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k\|G_{\mathrm{r}}{}^k\|^2 + (\gamma^\top\alpha^k - L(\hat{x}^k,\hat{y}^k,\hat{r}^k)),$$

and the expression for δ^k in (iii) follows from the definitions of γ and e^k. □

The dual form for (15) given in Proposition 2 reveals the importance of keeping the size of the bundle small, to reduce the computational burden per iteration.

Remark 1 When solving (15) in its dual form, active-set QP solvers provide at most $m+1$ Carathéodory-like positive simplicial multipliers α_j^k, [Kiw] and [Fra02]. Since in Step 5 the updated bundle only keeps active elements, it will always be formed by most $m+2$ indices, *independently* of the cardinality of $\mathscr{B}^k$. □

Note in passing that the update (18) confirms that having $r\ge 0$ is superfluous in (7): the dual r-component remains nonnegative, as long as the algorithm starts with $r^0\ge 0$.

3.3 Subgradient and bundle methods

Recall that in (7) we are dealing with the maximization of the nonsmooth dual function defined by $\min_{x\in X} L(x,y,r)$. Combining various definitions, it is easy to see that

$$\begin{pmatrix} h(x^k) \\ \sigma(h(x^k)) \end{pmatrix} \quad \text{with} \quad x^k \in X_0(y^k, r^k)$$

is a supergradient of the concave dual function. Therefore, a subgradient method to solve (7) sets

$$\begin{pmatrix} y^{k+1} \\ r^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ r^k \end{pmatrix} + \begin{pmatrix} t_{\mathbf{y}}{}^k & h(x^k) \\ t_{\mathbf{r}}{}^k & \sigma(h(x^k)) \end{pmatrix}.$$

This dual update can be compared with a very specific instance of Algorithm 1, working a singleton bundle index set, that is with $\mathscr{B}^k = \{k\}$ for all k. In this case, (15) has the explicit solution

$$\begin{pmatrix} y^{k+1} \\ r^{k+1} \end{pmatrix} = \begin{pmatrix} \hat{y}^k \\ \hat{r}^k \end{pmatrix} + \begin{pmatrix} t_{\mathbf{y}}{}^k & h(x^k) \\ t_{\mathbf{r}}{}^k & \sigma(h(x^k)) \end{pmatrix}.$$

This is an *extra* subgradient step (it would be a genuine subgradient step if $(\hat{y}^k, \hat{r}^k) = (y^k, r^k)$).

Both the subgradient and cutting-plane methods generally fail in delivering accurate numerical solutions as they tend to oscillate and exhibit bad numerical behavior when iterations progress. In nonsmooth optimization, monotonicity is a useful device to counter oscillations. The *Modified Subgradient Method* – MSM – studied in [Gas02], [Bur+06], [BIM13] perturbs slightly the subgradient step in a manner that makes the sequence $\{L(x^k, y^k, r^k)\}$ monotonically increasing. This is achieved by carefully choosing a correcting stepsize $s^k > 0$ and setting (with $t_{\mathbf{y}}{}^k = t_{\mathbf{r}}{}^k$)

$$\begin{pmatrix} y^{k+1} \\ r^{k+1} \end{pmatrix} = \begin{pmatrix} y^k \\ r^k \end{pmatrix} + \begin{pmatrix} t_{\mathbf{y}}{}^k & h(x^k) \\ t_{\mathbf{r}}{}^k & \sigma(h(x^k)) \end{pmatrix} + \begin{pmatrix} 0 \\ s^k \sigma(h(x^k)) \end{pmatrix}.$$

In agreement with the bundle method philosophy, we now show that the serious-step subsequence, with iterates satisfying (17) in Step 4 of Algorithm 1, is monotone.

Lemma 1 (Monotonicity of serious subsequence) *Consider the dual iterate (y^{k+1}, r^{k+1}) in Step 1 of Algorithm 1. The following holds for the predicted increase given in* (16)*:*

$$\hat{x}^k \in X_0(\hat{y}^k, \hat{r}^k) \Longrightarrow \delta^k = t_{\mathbf{y}}{}^k \|G_{\mathbf{y}}{}^k\|^2 + t_{\mathbf{r}}{}^k (G_{\mathbf{r}}{}^k)^2 + e^k \geq t_{\mathbf{y}}{}^k \|G_{\mathbf{y}}{}^k\|^2 + t_{\mathbf{r}}{}^k (G_{\mathbf{r}}{}^k)^2 . \tag{20}$$

As a result, the subsequence $\{L(\hat{x}^k, \hat{y}^k, \hat{r}^k)\}$ is monotonically increasing.

Proof The identity $\delta^k = t_{\mathbf{y}}{}^k \|G_{\mathbf{y}}{}^k\|^2 + t_{\mathbf{r}}{}^k (G_{\mathbf{r}}{}^k)^2 + e^k$, with $e^k = \min_{j\in\mathscr{B}^k} L(x^j, \hat{y}^k, \hat{r}^k) - L(\hat{x}^k, \hat{y}^k, \hat{r}^k)$, is given in Proposition 2(iii). The result then follows because $\hat{x}^k \in X_0(\hat{y}^k, \hat{r}^k)$ implies $e^k \geq 0$. □

The convergence analysis for Algorithm 1 is subsumed by the inexact version presented in Section 4. For now we mention that if the conceptual Algorithm 1 stops after finitely many steps with $h(x^{k+1}) = 0$, Corollary 1 gives primal and dual optimality.

3.4 Numerical assessment

We consider 28 instances of problem (12), randomly (and sparsely) generated by using three different seeds (to initialize the pseudo-random number generators `randn` in Matlab) to generate vector q and matrix A. The matrix Q was generated as $Q = R^\top R + I$, where $I \in \mathbb{R}^{n\times n}$ is the identity matrix and $R^{10\times n}$ was randomly generated by the Matlab's command `randn`. The right-hand-side vector of (12) was obtained by setting $b = Ax_0$, where x_0 is the vector having the first half of elements equal to 1 and the second half equal to 0: $x_0 = (1,\ldots,1,0,\ldots,0)$. The number of linear constraints was set to $\min\{n/2, 800\}$.

For both cases in Subsection 3.1.1, we consider `ENUM`, a direct solution strategy solving all the N QP's (13), using `Gurobi`, and two solvers. Namely, `PDBM` - Algorithm 1 with the LU-decomposition described

in Subsection 3.1.1; and MSM - in the notation of [BIM13, § 3], the stepsize is $s^k = 0.5\,(\eta_k + \beta_k)$, where $\eta_k := \min\{0.5,\, \|h(x)\|_2\}$ and $\beta_k := \max\{2,\, \sigma(h(x))\}$. Moreover, we set $\alpha_k := \max\{1/k,\, 10^{-5}\}$.

All the results can be found in the appendix, reported in Table 4. The columns in the table contain indicators of optimality gap and feasibility ($\Delta\varphi(\bar{x})$ and $h(\bar{x})$), the number of iterations (how many times the Lagrangian subproblems were solved, $\#X_0$) and the total CPU time. The gaps were computed considering as the optimal value of the primal problem (1) the one found by ENUM. Both PDBM and MSM start with $y^0 = 0 \in \mathbb{R}^m$ and $r^0 = 1$, and stop either when Tol $= 10^{-7}$, or after reaching 500 iterations.

The figures in Table 4 confirm that both MSM and PDBM were successful in getting close to the value computed by ENUM. However, PDBM did so in much shorter CPU times and with higher accuracy, having performed less iterations. In two instances MSM reached the maximum of iterations. Those are difficult instances for PDBM too, that needed about 150 iterations to converge, when typically only 15 iterations sufficed for the other 26 instances. For these two problems, the final value of r computed by PDBM (not reported in the table) was also much larger than for the rest (in all of our experiments, adjusting r seemed more delicate). When compared to the exact ENUM, our method proves significantly faster for the larger instances. The enumerative approach took approximately 32 minutes to solve the last instance with $n = 2000$ and $N = 500$ whereas PDBM took only 29 seconds and solver MSM 133 seconds.

For Case 2, the same test problems have the additional constraint $x \geq 0$ in (12). The global minimization of Lagrangian subproblems involves solving a sequence of LCP's by the method *Block Principal Pivoting* of [JP94]. For this harder exact procedure MSM was not competitive. Table 5 in the appendix compares PDBM10 and PDBM100, respectively referring to PDBM initialized with $(y^0, r^0) = (0, \ldots, 0, 10)$ and $(y^0, r^0) = (0, \ldots, 0, 100)$. Once more, our primal-dual approach is more efficient than the enumeration for the larger problems. The initialization (y^0, r^0) certainly impacts the iterative process, however, the limited results are not conclusive on that matter: sometimes PDBM10 performs fewer iterations than PDBM10, and vice-versa.

A summary of the results for both cases can be found in Table 1.

Table 1: Mean and standard deviation for Tables 4 (left) and 5 (right)

	CASE 1						CASE 2					
	PDBM		MSM		ENUM		PDBM10		PDBM100		ENUM	
	avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev
$\Delta\varphi$	-1E-03	3E-03	-7E-02	3E-01	0	0	1E-03	4E-03	1E-03	46E-03	0	0
$h(\bar{x})$	4E-08	3E-08	1E-03	6E-03			4E-08	3E-08	3E-08	39E-08		
$\#X_0$	25	42	105	115			58	81	28	56		
CPU (s)	5	8	21	36	209	515	199	366	196	368	744	1904

4 An implementable primal-dual bundle method

For most applications, there is no exact procedure to globally minimize the Lagrangian function, as required by Assumption 3. In order to render implementable the conceptual Algorithm 1, the exact primal iterates in Step 2 must be replaced by approximations. This procedure is formalized by Assumption 4 below. The bundle machinery is set up as before, but now there is an additional check, to identify situations in which the inaccuracy in the approximate primal iterates starts tampering with convergence. This is the mechanism of *noise attenuation*, introduced by [Kiw06], already hinted in [Hin01] and explained in Subsection 4.2.

Assumption 4 (Inexact solution of Lagrangian subproblems: $\varepsilon \geq 0$ in (8)) *An inexact procedure is available for* (1) *such that, when applied to the Lagrangian subproblem* (11)*, delivers an approximate minimizer as in* (8)*. Such property is represented on the primal-dual iterates (x^k, y^k, r^k) of Algorithm 2 below by the inclusion below*

$$\forall k\,, x^k \in X_{\varepsilon_k}(y^k, r^k) \iff L(x^k, y^k, r^k) \leq L(x, y^k, r^k) + \varepsilon_k \text{ for all } x \in X\,.\square$$

4.1 Example: minimization over the ℓ_0-norm

Consider the following variant of problem (1)

$$\min_{x\in\mathbb{R}^n_+} \varphi(x) \quad \text{s.t.} \quad Ax = b \ \text{ and } \ \|x\|_0 \le K\,, \tag{21}$$

where $\varphi : \mathbb{R}^n \to \mathbb{R}$ is a convex and continuously differentiable function, $0 < K < n$ is a given natural number and $\|x\|_0 := \#\{i \in \{1,\dots,n\} : x^i > 0\}$ is the ℓ_0-norm of x (the number of nonzero elements of x). The ℓ_0-norm is employed to induce sparsity in compressed sensing, logistic regression, and applications in machine learning and telecommunications.

Proximal Lagrangian. Taking $h(x) = Ax - b$ and letting $\sigma(\cdot) = \frac{1}{2}\|\cdot\|^2$ yields

$$L(x,y,r) = \varphi(x) + \langle y, Ax-b\rangle + \frac{r}{2}\|Ax-b\|^2\,,$$

whose minimization entails solving the nonconvex smooth Lagrangian subproblem:

$$\min_{x\ge 0} L(x,y,r) \quad \text{s.t.} \quad \|x\|_0 \le K\,.$$

Despite nonconvexity, the projection $\bar{x} := P_X[z]$ of a given $z \in \mathbb{R}^n$ can be easily computed, proceeding in two steps. First compute the projection onto $\mathbb{R}^n_+$, $z^+ := P_{\mathbb{R}^n_+}[z]$ and sort the components in the descending order: $z^+_{o(1)} \ge z^+_{o(2)} \ge \ldots \ge z^+_{o(n)}$. Second, take $\bar{x}^{o(i)} := z^+_{o(i)}$ for all $i = 1,\dots,K$, and $\bar{x}^{o(i)} := 0$ for all $i = K+1,\dots,n$.

As the projection onto X is easily performed, we can employ the *Nonconvex Projected Gradient Method* of [Pan+17]. Since the ordering of elements $z^+_{o(1)} \ge z^+_{o(2)} \ge \ldots \ge z^+_{o(n)}$ is not necessarily unique, the projection onto X may not be unique either, thus precluding the global convergence of projected gradient methods. But computing an inexact minimum as in Assumption 4 is possible.

Sharp Lagrangian. We first rewrite the constraints $\|x\|_0 \le K$ as an equality,

$$\|x\|_0 \le K \quad \equiv \quad \|x\|_1 - \|x\|_{(K)} = 0\,,$$

where $\|x\|_1 = \sum_{i=1}^n |x^i|$ is the ℓ_1-norm and $\|x\|_{(K)} = \sum_{i=1}^K |x^{o(i)}|$ is the sum of the K largest components (in absolute value) of x.

Letting $\sigma(\cdot) = |\cdot|$, taking $X := \{x \in \mathbb{R}^n_+ : Ax = b\}$ and $h(x) = \|x\|_1 - \|x\|_{(K)} = 0$, yields in (3)

$$\begin{aligned} L(x,y,r) &= \varphi(x) + y(\|x\|_1 - \|x\|_{(K)}) + r\,|(\|x\|_1 - \|x\|_{(K)})| \\ &= \varphi(x) + (y+r)(\|x\|_1 - \|x\|_{(K)}) \end{aligned}$$

Since $x \mapsto \|x\|_{(K)}$ is a convex function, the Sharp Lagrangian subproblems

$$\min_{x\ge 0} L(x,y,r) \quad \text{s.t.} \quad Ax = b$$

are DC programs, solvable by specialized DC algorithms, such as [OT19]. Solving a DC program globally is in general NP hard, but applying a locally convergent DC algorithm yields an inexact procedure as in Assumption 4.

4.2 The need of attenuating noise

In an implementable algorithm employing an inexact procedure, errors need to be tackled with when generating dual iterates and also in the stopping test. To properly manage the "noise" introduced by inexactness in the primal iterates, monotonicity of the subsequence of Lagrangian values at serious steps must be preserved. The inequality (20), which depends on having $\hat{x}^k \in X_0(\hat{y}^k, \hat{r}^k)$, is crucial in this sense.

In the inexact setting of Assumption 4, the primal iterate no longer belongs to $X_0(\hat{y}^k, \hat{r}^k)$ but to $X_{\varepsilon_{\hat{x}}^k}(\hat{y}^k, \hat{r}^k)$ for some inaccuracy $\varepsilon_{\hat{x}}^k > 0$. There is no reason for (20) to hold, but that crucial inequality is used to detect when inexactness became cumbersome, to the extent of preventing the desired monotonicity property to hold. The rationale is that with exact calculations (20) holds, so, if the inequality does not hold, i.e., when

$$\delta^k < t_{\mathrm{y}}{}^k \|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k (G_{\mathrm{r}}{}^k)^2 ,$$

this flags that the primal value $\hat{x}^k \in X_{\varepsilon_{\hat{x}}^k}(\hat{y}^k, \hat{r}^k)$ was too inaccurate.

The implementable algorithm handles inexactness along those lines, to identify when the error from the inexact procedure is excessive. Given a parameter $\beta \in [0.5, 1)$, if

$$\delta^k < (1-\beta)[t_{\mathrm{y}}{}^k \|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k (G_{\mathrm{r}}{}^k)^2] \quad = \quad (1-\beta)\Big(\frac{1}{t_{\mathrm{y}}{}^k}\|y^{k+1} - \hat{y}^k\|^2 + \frac{1}{t_{\mathrm{y}}{}^k}(r^{k+1} - \hat{r}^k)^2\Big) \tag{22}$$

then the error $\varepsilon_{\hat{x}}^k$ is excessively large. If such is the case, the corrective action consists in discarding (y^{k+1}, r^{k+1}), sharply increasing t_{y} and t_{r}, and finding a new dual pair in Step 1. Neither the bundle nor $(\hat{y}^k, \hat{r}^k)$ are changed. Increasing the stepsizes drives the dual iterate closer to the reference values $\hat{y}^k$ and $\hat{r}^k$. The process is repeated until the inequality above holds; see [OSL] and the discussion in [OS20, § 4.2].

4.3 Implementable algorithm

We are now in a position to state a primal-dual bundle method that allows Lagrangian subproblems to be solved inexactly, and therefore is suitable to induce separability in (3) via approximations.

This is Algorithm 2 below, where we highlight the main differences with respect to the conceptual scheme in Algorithm 1. Essentially, allowing for inexact calculations in Step 2 introduces an intermediate Step $1\frac{1}{2}$ detecting if the inaccuracy is too large, to activate a noise attenuation flag, if need be ($\mathtt{na} = 1$). The flag is used in Step 4 to prevent the stepsizes t_{y} and t_{r} to change in case of null steps (a convergence requirement). The other steps are as in the conceptual scheme.

Algorithm 2 AN IMPLEMENTABLE PRIMAL-DUAL BUNDLE METHOD FOR AUGMENTED LAGRANGIAN DUALS UNDER ASSUMPTION4

Input and starting point. Choose $(y^1, r^1) \in \mathbb{R}^m \times \mathbb{R}_+$ and let $x^1 \in X_{\varepsilon_1}(y^1, r^1)$.
Parameters. $m \in (0,1)$, $t_{\mathrm{y}}{}^1 \geq t_{\mathrm{y}\min} > 0$, $t_{\mathrm{r}}{}^1 \geq t_{\mathrm{r}\min} > 0$, $\beta \in [0.5, 1)$, and $\mathtt{Tol} \geq 0$.
Step 0: Initialization. As Step 0 of Algorithm 1. Initialize the noise attenuation flag $\mathtt{na} := 0$.
Step 1: Dual Iterate. Solve the QP (19) to obtain α^k, compute (y^{k+1}, r^{k+1}) by rule (18) and the predicted increase (16).
Step $1\frac{1}{2}$: Noise Attenuation. If (22) holds, set $\mathtt{na} = 1$, $t_{\mathrm{y}}{}^{k+1} = 10 t_{\mathrm{y}}{}^k$, $t_{\mathrm{r}}{}^{k+1} = 10 t_{\mathrm{r}}{}^k$, $\mathscr{B}^{k+1} = \mathscr{B}^k$, $(\hat{x}^{k+1}, \hat{y}^{k+1}, \hat{r}^{k+1}) := (\hat{x}^k, \hat{y}^k, \hat{r}^k)$. Set $k = k+1$, and go back to Step 1.
Step 2: Inexact Primal Iterate. Find $x^{k+1} \in X_{\varepsilon_{k+1}}(y^{k+1}, r^{k+1})$
Step 3: Stopping Test. If $\sigma(h(x^{k+1})) \leq \mathtt{Tol}$ stop: x^{k+1} solves approximately (1) and (y^{k+1}, r^{k+1}) solves approximately (7).
Step 4: Monotonicity Test. As Step 4 of Algorithm 1. If the iteration is declared serious, set $\mathtt{na} := 0$.
If the iteration is null, choose $t_{\mathrm{y}}{}^{k+1} \in [(1-\mathtt{na})t_{\mathrm{y}\min} + \mathtt{na}\, t_{\mathrm{y}}{}^k, t_{\mathrm{y}}{}^k]$ and $t_{\mathrm{r}}{}^{k+1} \in [(1-\mathtt{na})t_{\mathrm{r}\min} + \mathtt{na}\, t_{\mathrm{r}}{}^k, t_{\mathrm{r}}{}^k]$.
Step 5: Loop. As Step 5 of Algorithm 1.

We suppose the inexact procedure delivers errors that are uniformly bounded:

$$\text{in Assumption 4 there exists some } \eta \geq 0 \text{ such that } \varepsilon_k \in [0, \eta] \text{ for all } k. \tag{23}$$

The error bound in (23) can be unknown, but must be the same for all primal-dual points (x^k, y^k, r^k). Additionally, note that when $\eta = 0$ we are back to the setting of Assumption 3, i.e., requiring an exact procedure delivering global solutions.

We first examine the loop of noise attenuation. The properties of the augmenting function (2) prevent Algorithm 2 from doing infinitely many consecutive noise attenuation steps.

Lemma 2 (Noise attenuation terminates) *The following holds for Algorithm 2.*

(i) Let $\hat{\varepsilon}_{\hat{x}}^k$ denote the error incurred when computing the primal point at a serious step, so that $\hat{x}^k \in X_{\hat{\varepsilon}_{\hat{x}}^k}(\hat{y}^k,\hat{r}^k)$. Then $-G_{\mathrm{r}}{}^k \in \Delta_{\mathrm{E}^k}^{\sigma}(\hat{y}^k,\hat{r}^k)$ for $0 \le \mathrm{E}^k \le e^k + \hat{\varepsilon}_{\hat{x}}^k$.

(ii) If (23) holds and $\mathtt{Tol} = 0$, the noise attenuation loop between Steps 1 and $1\frac{1}{2}$ is finite.

Proof The inclusion in item (i) is just Proposition 2(iii). The expression for the error E^k is

$$\mathrm{E}^k = \sum_{j\in\mathscr{B}^k} \alpha_j^k L(x^j,\hat{y}^k,\hat{r}^k) - \min_{x\in X} L(x,\hat{y}^k,\hat{r}^k) = e^k + L(\hat{x}^k,\hat{y}^k,\hat{r}^k) - \min_{x\in X} L(x,\hat{y}^k,\hat{r}^k)\,,$$

by the definition of e^k in Proposition 2(ii). The bound for E^k follows, because $\hat{x}^k \in X_{\hat{\varepsilon}_{\hat{x}}^k}(\hat{y}^k,\hat{r}^k)$ is equivalent to having $L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \le \min_{x\in X} L(x,\hat{y}^k,\hat{r}^k) + \hat{\varepsilon}_{\hat{x}}^k$. Regarding item (ii), first notice using (23) in $0 \le \mathrm{E}^k \le e^k + \hat{\varepsilon}_{\hat{x}}^k$ provides the uniform lower bound $e^k \ge -\eta$. Next, recall the identity $\delta^k = t_{\mathrm{y}}{}^k\|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k(G_{\mathrm{r}}{}^k)^2 + e^k$ from Proposition 2(ii) and note that there is noise attenuation if and only if $e^k < -\beta[t_{\mathrm{y}}{}^k\|G_{\mathrm{y}}{}^k\|^2 + t_{\mathrm{r}}{}^k(G_{\mathrm{r}}{}^k)^2]$. Combining the bounds for e^k yields

$$\frac{\eta}{\min\{t_{\mathrm{y}}{}^k, t_{\mathrm{r}}{}^k\}} > \beta[\|G_{\mathrm{y}}{}^k\|^2 + (G_{\mathrm{r}}{}^k)^2] \quad \text{whenever there is noise attenuation.}$$

Suppose that at certain iteration $\bar{k}$ the algorithm loops forever between Steps 1 and $1\frac{1}{2}$. In this case, $\mathscr{B}^k = \bar{\mathscr{B}} := \mathscr{B}^{\bar{k}}$ for all $k \ge \bar{k}$, Step $1\frac{1}{2}$ drives $\min\{t_{\mathrm{y}}{}^k, t_{\mathrm{r}}{}^k\}$ to infinity and, therefore, $\lim_{k\to\infty}\|G_{\mathrm{r}}{}^k\| = 0$ from the above inequality. This limit implies $\lim_{k\to\infty}\sum_{j\in\bar{\mathscr{B}}} \alpha_j^k \sigma(h(x^j)) = 0$ due to the definition of $G_{\mathrm{r}}{}^k$ in (18) and (2). As $\alpha^k \in \Delta^k$ and the primal points x^j, with $j \in \bar{\mathscr{B}}$, are always the same regardless the iteration $k \ge \bar{k}$, we conclude that there exists at least a point x^j such that $\sigma(h(x^j)) = 0$. But in this case, as $\mathtt{Tol} = 0$, the algorithm would have stopped at iteration $j < \bar{k}$, yielding the desired contradiction. □

We shall see that the partial aggregate subgradient in Lemma 2(i) eventually vanishes, triggering the approximate primal and dual optimality results stated in Theorem 2.

4.4 Dual convergence

To ease the understanding of the proofs that follow, the elements of our primal-dual bundle scheme are identified with those in [OSL].

Remark 2 (Relations with [OSL]) The Algorithmic Pattern 4.2 in [OSL] minimizes a convex function f on variable u, which for Algorithm 2 amounts to taking

$$u := (y,r) \in \mathbb{R}^{m+1} \quad \text{and} \quad f(u) = -\min_{x\in X} L(x,y,r)\,.$$

The aggregate subgradient $\hat{g}_k$ in Lemma 4.2 in [OSL] corresponds to $-(G_{\mathrm{y}}{}^k, G_{\mathrm{r}}{}^k)$ in (18). The level ℓ_k and model decrease δ_k^M in [OSL, (3.8)] are, respectively, $-L(\hat{x}^k,\hat{y}^k,\hat{r}^k)$ and δ^k, and the descent test (3.7) in [OSL] is the monotonicity test (17). The nominal decrease is equal to δ^k, so $\alpha_k = 0$ in [OSL, (4.3)]. The effective decrease in the descent test is $-L(\hat{x}^k,\hat{y}^k,\hat{r}^k) + L(x^{k+1},y^{k+1},r^{k+1})$ and $\hat{e}_k$, the aggregate linearization gap [OSL, (4.4)] is our e^k.

A procedure in Assumption 4 for which (23) holds can be seen as a inexact oracle of the lower type, as in [OS14, (2.3)]. In the notation of [OS14],

$$f_k^M(u) := -\min_{j\in\mathscr{B}_k} L(x^j,y,r) \le -\min_{x\in X} L(x,y,r) = f(u) \text{ for all } u \in \mathbb{R}^{m+1},$$

yielding satisfaction of [OS14, (4.11)], written with $\eta^M = 0$.

Regarding the noise detection mechanism, the parameters in [OS14, Lemma 5.1] are $\alpha_k = 0$, $b = 1$ and $\beta_k = \beta$. The test in Step $1\frac{1}{2}$ corresponds to the right-hand side one in [OS14, (5.2)].

Finally, the optimality certificate [OSL, (4.8)-(4.9)] is given by the pair

$$\begin{pmatrix} G_{\mathbf{y}}{}^k \\ G_{\mathbf{r}}{}^k \end{pmatrix} \quad \text{and} \quad \phi^k \quad \text{from Proposition 2,}$$

endequation respectively controlling feasibility and the duality gap.

Altogether, conditions (6.15) and (6.16) in [OS14] hold, and Theorems 6.11 and 4.5 in [OS14] are applicable to Algorithm 2. □

For the convergence analysis, the stopping tolerance is set to zero and either the algorithm terminates having triggered its stopping test, or it loops forever. The first situation is addressed by the following statement, a direct application of Theorem 1.

Lemma 3 (Primal-dual optimality in case of finite termination) *Suppose that Algorithm 2 with* $\texttt{Tol} = 0$ *stops at iteration* $k \geq 1$. *Then* (y^{k+1}, r^{k+1}) *is an* ε_{k+1} *solution of* (7) *and* x^{k+1} *is an* ε_{k+1}*-solution to the primal problem* (1). □

When the algorithm does not stop, by Lemma 2, noise attenuation is finite so either there is a last serious iterate followed by infinitely many null steps, or an infinite sequence of serious iterates is generated. Both possibilities are covered by Theorems 6.11 and 4.5 in [OS14].

Theorem 5 (Dual convergence) *Suppose the inexact procedure in Assumption 4 satisfies* (23). *When* $\texttt{Tol} = 0$ *and Algorithm 2 does not stop, only one of the situations below can occur.*

(i) At iteration $\hat{k}$ *there is a last serious step* $\hat{x} \in X_{\hat{\varepsilon}}(\hat{y}, \hat{r})$*, followed by infinitely many null steps. Then the primal iterates satisfy*

$$\lim_{k \geq \hat{k}} \sum_{j \in \mathscr{B}_k} \alpha_j^k \sigma(h(x^j)) = 0 \quad \textit{and} \quad \limsup_{k \geq \hat{k}} \sum_{j \in \mathscr{B}_k} \alpha_j^k \varphi(x^j) \leq L(\hat{x}, \hat{y}, \hat{r}) \,. \tag{24}$$

The bound below for the optimality gap holds

$$\texttt{DV}^\sigma \leq L(\hat{x}, \hat{y}, \hat{r}) \leq \texttt{DV}^\sigma + \hat{\varepsilon} \leq \texttt{PV}^\sigma + \hat{\varepsilon} \,,$$

and $(\hat{y}, \hat{r})$ *solves* (7) *up to* $\hat{\varepsilon}$.

(ii) When the subsequence of serious steps is infinite, we let K_∞ *denote the set of corresponding iteration indices and* $\hat{\varepsilon}_\infty := \limsup_{k \in K_\infty} \hat{\varepsilon}_{\hat{x}}^k$ *. The primal iterates satisfy*

$$\lim_{k \in K_\infty} \sum_{j \in \mathscr{B}_k} \alpha_j^k \sigma(h(x^j)) = 0 \quad \textit{and} \quad \limsup_{k \in K_\infty} \sum_{j \in \mathscr{B}_k} \alpha_j^k \varphi(x^j) \leq \limsup_{k \in K_\infty} L(\hat{x}^k, \hat{y}^k, \hat{r}^k) \,. \tag{25}$$

The bound below for the optimality gap holds

$$\texttt{DV}^\sigma \leq \limsup_{k \in K_\infty} L(\hat{x}^k, \hat{y}^k, \hat{r}^k) \leq \texttt{DV}^\sigma + \hat{\varepsilon}_\infty \leq \texttt{PV}^\sigma + \hat{\varepsilon}_\infty \,.$$

(iii) In the setting of item (ii), if the dual part of the serious subsequence clusters at (y^∞, r^∞)*, then this limit point solves* (7) *up to* $\hat{\varepsilon}^\infty$.

Proof Using the correspondences in Remark 2, define the following infinite index-set K_∞:

$$K_\infty := \left\{k \geq \hat{k}\right\} \quad \text{in case (i)} \quad \text{and} \quad K_\infty := \{k \text{ declaring a serious step}\} \qquad \text{in case (ii).}$$

We use the following two results from [OS14]. First, by [OS14, Theorem 4.5 and (4.9)], the measures of feasibility and optimality gap satisfy

$$\begin{pmatrix} G_{\mathbf{y}}{}^k \\ G_{\mathbf{r}}{}^k \end{pmatrix} \to 0 \text{ and } \phi^k \to \phi \leq 0 \text{ for } k \in K_\infty \,.$$

Writing down the expressions for the partial subgradient and aggregate gap in Proposition 2,

$$\lim_{k\in K_\infty} \sum_{j\in|\mathscr{B}_k|} \alpha_j^k \sigma(h(x^j)) = 0 \quad \text{and} \quad \lim_{k\in K_\infty} \Big(\sum_{j\in\mathscr{B}_k} \alpha_j^k \varphi(x^j) - L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \Big) \le 0\,,$$

showing both (24) and (25). Second, the inequality [OS14, (4.12)] ensures that

$$\limsup_{k\in K_\infty} -L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \le -\sup_{(y,r)\in\mathbb{R}^{m+1}} \min_{x\in X} L(x,y,r) = -\mathtt{DV}^\sigma\,.$$

Since $\hat{x}^k \in X_{\hat{\varepsilon}_{\hat{x}}^k}(\hat{y}^k,\hat{r}^k)$ implies that $L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \le \min_{x\in X} L(x,\hat{y}^k,\hat{r}^k) + \hat{\varepsilon}_{\hat{x}}^k \le \mathtt{DV}^\sigma + \hat{\varepsilon}_{\hat{x}}^k$, we see that

$$\mathtt{DV}^\sigma \le \limsup_{k\in K_\infty} L(\hat{x}^k,\hat{y}^k,\hat{r}^k) \le \mathtt{DV}^\sigma + \limsup_{k\in K_\infty} \hat{\varepsilon}_{\hat{x}}^k\,,$$

showing the assertion on optimality gaps for both items (i) and (ii) ($\mathtt{DV}^\sigma \le \mathtt{PV}$ by weak duality).

It remains to show dual optimality in (i) and (iii). By Lemma 2(i),

$$-G\mathbf{r}^k \in \Delta_{\mathtt{E}^k}^\sigma(\hat{y}^k,\hat{r}^k) \text{ for } 0 \le \mathtt{E}^k \le e^k + \hat{\varepsilon}_{\hat{x}}^k\,.$$

If the dual part of the serious-subsequence $\{(\hat{y}^k,\hat{r}^k)\}$ has a limit, say (y^∞,r^∞), then

$$0 \in \Delta_{\mathtt{E}^\infty}^\sigma(\hat{y}^\infty,\hat{r}^\infty) \text{ for } 0 \le \mathtt{E}^\infty \le \limsup_{k\in K_\infty} (e^k + \hat{\varepsilon}_{\hat{x}}^k) \tag{26}$$

(recall the σ-simplex is an outer semicontinuous multifunction of (ε,y,r)). For item (i), having a fixed $\hat{k}$ means that $(y^\infty,r^\infty) = (\hat{y},\hat{r})$ and the relation for ϕ^k in Proposition 2(ii) becomes

$$\phi^k = e^k - \langle G\mathbf{y}^k,\hat{y}\rangle - \langle G\mathbf{r}^k,\hat{r}\rangle$$

so $\limsup_{k\ge\hat{k}} e^k = \phi \le 0$. Therefore the error in (26) satisfies $\mathtt{E}^\infty \le \hat{\varepsilon}$, so $0 \in \Delta_{\hat{\varepsilon}}^\sigma(\hat{y},\hat{r})$, and Theorem 2 gives the result. For item (iii), we now write

$$e^k = \phi^k + \langle G\mathbf{y}^k,\hat{y}^k\rangle + \langle G\mathbf{r}^k,\hat{r}^k\rangle\,.$$

Over the index set K'_∞ the dual iterates $(\hat{y}^k,\hat{r}^k)$ form a convergent sequence and the two rightmost terms above vanish in the limit together with the aggregate gradients. Hence, $\limsup_{k\in K'_\infty} e^k \le 0$, the error in (26) satisfies $\mathtt{E}^\infty \le \hat{\varepsilon}^\infty$, and approximate dual optimality follows from Theorem 2. □

The bound for the optimality gap in Theorem 5(ii) holds *regardless* of the dual problem (7) having a nonempty solution set: whether $\mathtt{DV}^\sigma$ in (7) is finite or not, in the limit the monotone sequence $\{L(\hat{x}^k,\hat{y}^k,\hat{r}^k)\}$ will always approach the optimal value up to the accuracy.

4.5 Primal convergence

Primal convergence needs some more analysis, as the characterization in Theorem 2 now involves simplicial sums over the bundle sets $\mathscr{B}_k$, that change along iterations, in both (24) and (25). The following convoluted notation is introduced to adjust the simplicial coefficients and rewrite those sums over a fixed number of $m+2$ terms, independently of the iteration index k.

Notation 6 (Relabeling simplicial coefficients) *We denote by $i(k) \in \mathscr{B}^k$ the index of i^{th}-largest component of vector α^k solving* (19)*. The rigorous definition is given below*

$$i(k) := j_i \text{ for all } i = 1,\dots,|\mathscr{B}^k|, \text{ with } j_i \in \mathscr{B}^k \text{ s.t. } \alpha_{j_1}^k \ge \alpha_{j_2}^k \ge \dots \ge \alpha_{j_{|\mathscr{B}^k|}}^k\,.$$

Since by Remark 1 the number of nonzero components of the vector α^k is at most $m+2$,

$$\alpha_{i(k)}^k = 0 \quad \text{for all } i \text{ s.t. } m+2 < i \le |\mathscr{B}^k|$$

(if $|\mathscr{B}^k| < m+2$, the vector α^k can be complemented with zeros at the end). Then we denote

$$\mathtt{a}_i^k := \alpha^k_{i(k)} \quad \textit{for all } i = 1,\dots,m+2\,,$$

which, from the definitions of $i(k)$ and α^k, satisfies $\mathtt{a}^k \in \Delta^{m+1}$ with $\mathtt{a}_1^k \geq \mathtt{a}_2^k \geq \ldots \geq \mathtt{a}_{m+2}^k \geq 0$. Finally, for every index i we define the following sequences

$$\{\mathtt{a}_i^k\}_k \quad \textit{and} \quad \{x^{i(k)}\}_k, \quad \textit{for all } i = 1,\dots,m+2\,.\square$$

Thanks to the new notation, in (24) and (25) we can write, for instance,

$$\text{the sum} \quad \sum_{j\in\mathscr{B}^k} \alpha_j^k \varphi(x^j) \quad \text{as} \quad \sum_{i=1}^{m+2} \mathtt{a}_i^k \varphi(x^{i(k)})\,.$$

We now show that our bundle mechanism provides an implementation of Theorem 2(ii).

Theorem 7 *[Primal convergence] In the setting of items (i) and (ii) in Theorem 5, and using Notation 6, the following holds.*

(i) Algorithm 2 generates an infinite index-iteration set K'_∞ such that for all $i = 1,\dots,m+2$,

$$\lim_{k\in K'_\infty} \mathtt{a}_i^k = \bar{\mathtt{a}}_i \textit{ with } \bar{\mathtt{a}} \in \Delta^{m+2}, \quad \textit{and} \quad \lim_{k\in K'_\infty} x^{i(k)} = \bar{x}^i$$

with $h(\bar{x}^i) = 0$ for all i such that $\bar{\mathtt{a}}_i > 0$. These limit points satisfy

$$\mathtt{DV}^\sigma \leq \sum_{i=1}^{m+2} \bar{\mathtt{a}}_i \varphi(\bar{x}^i) \leq \mathtt{DV}^\sigma + \hat{\varepsilon}'_\infty \leq \mathtt{PV}^\sigma + \hat{\varepsilon}'_\infty\,.$$

(ii) Among the limit primal points, $\bar{x}^{\mathtt{i}_{\min+}}$ solves (1) up to ε'_∞, where we defined

$$\textit{the index } \mathtt{i}_{\min+} := \arg\min_{i=1,\dots,m+2}\{\varphi(\bar{x}^i) : \bar{\mathtt{a}}_i > 0\}\,, \textit{ and the accuracy } \hat{\varepsilon}'_\infty := \limsup_{k\in K'_\infty} \hat{\varepsilon}_{\hat{x}}^k\,.$$

In other words, $\bar{x}^{\mathtt{i}_{\min+}} \in X$, $h(\bar{x}^{\mathtt{i}_{\min+}}) = 0$ and $\varphi(\bar{x}^{\mathtt{i}_{\min+}}) \leq \mathtt{PV} + \hat{\varepsilon}'_\infty$.

(iii) If the dual part of the serious subsequence clusters at (y^∞, r^∞), as in Theorem 5(iii), then

$$\mathtt{DV}^\sigma \leq \sum_{i=1}^{m+2} \bar{\mathtt{a}}_i \varphi(\bar{x}^i) \leq L(x^\infty, y^\infty, r^\infty) \leq \mathtt{DV}^\sigma + \hat{\varepsilon}'_\infty \leq \mathtt{PV}^\sigma + \hat{\varepsilon}'_\infty\,,$$

for the limit primal point $\hat{x}^\infty := \lim_{k\in K'_\infty} \hat{x}^k$.

Proof Existence of the index set $K'_\infty \subset K_\infty$ is guaranteed because the sequences $\{\mathtt{a}^k\}$ and $\{x^k\}$ are bounded. Item (i) follows from the statements in Theorem 5 in the new notation, letting $K_\infty = \{k \geq \hat{k}\}$ and $\hat{\varepsilon}'_\infty = \hat{\varepsilon}$ for the null-step tail. To see (ii), consider the limits in (24) and (25), written for K'_∞. On the left-hand limit, by the continuity of functions h and σ,

$$0 = \lim_{k\in K'_\infty} \sum_{j\in\mathscr{B}^k} \alpha_j^k \sigma(h(x^j)) = \lim_{k\in K'_\infty} \sum_{i=1}^{m+2} \mathtt{a}_i^k \sigma(h(x^{i(k)})) = \sum_{i=1}^{m+2} \bar{\mathtt{a}}_i \sigma(h(\bar{x}^i))\,.$$

By the properties of σ in (2), this means that every cluster point $\bar{x}^i$ with $\bar{\mathtt{a}}_i > 0$ is primal feasible. Proceeding similarly with the right-hand side limits in (24) and (25), by continuity of φ,

$$\sum_{i=1}^{m+2} \bar{\mathtt{a}}_i \varphi(\bar{x}^i) = \lim_{k\in K'_\infty} \sum_{i=1}^{m+2} \mathtt{a}_i^k \varphi(x^{i(k)}) = \lim_{k\in K'_\infty} \sum_{j\in\mathscr{B}_k} \alpha_j^k \varphi(x^j) \leq \limsup_{k\in K'_\infty} L(\hat{x}^k, \hat{y}^k, \hat{r}^k)\,,$$

By definition of $\bar{x}^{\mathtt{i}_{\min+}}$ the leftmost term is bounded below by $\varphi(\bar{x}^{\mathtt{i}_{\min+}})$. This shows the second item, because the rightmost limit is bounded above by $\mathtt{PV} + \hat{\varepsilon}'_\infty$ in Theorem 5(i) and (ii). Finally, by continuity of φ, h, and σ, $\limsup_{k\in K'_\infty} L(\hat{x}^k, \hat{y}^k, \hat{r}^k) = L(x^\infty, y^\infty, r^\infty)$, and item (iii) follows. $\square$

We emphasize that primal convergence is guaranteed up to the inaccuracy, even if there is no solution to the dual problem (the first two items in Theorem 7 do not depend Theorem 5(iii)).

Recall the bundle mechanism allows discarding information in Step 5. Nothing then guarantees that the limit point at serious iterates, $\hat{x}^\infty$ will be one of the points $\bar{x}^i$ in the theorem. It may very well happen that $\hat{x}^\infty$ is not feasible for (1). In some particular settings of Assumption 4, however, primal feasibility for $\hat{x}^\infty$ is ensured. We show below that such is the case if the inexact procedure can compute increasingly accurate values at serious steps. Indeed, sometimes finding a global solution to the Lagrangian subproblems (11) is possible, but expensive. The setting called *on-demand accuracy* in [OS14] can then be put in place, so that the inexact procedure delivers primal information that is more and more exact at serious steps.

Corollary 2 (Asymptotically exact calculations) *In addition to the assumptions of Theorem 7, suppose the inexact procedure in Assumption 4 has the ability to provide information with any level of accuracy, including exact one, so that* $\hat{\varepsilon}_\infty = 0$. *The following holds.*

(i) All the cluster points $\bar{x}^i$ *having* $\bar{a}_i > 0$ *solve* (1) *exactly.*
(ii) If the dual part of the serious subsequence clusters at (y^∞, r^∞)*, the corresponding limiting primal point,* $\hat{x}^\infty$*, solves* (1) *exactly.*

Proof By Theorem 7(i), if the error vanishes then $0 \geq \sum_{i=1}^{m+2} \bar{a}_i \varphi(\bar{x}^i) - \mathtt{DV}^\sigma \geq 0$. The first item thus follows from inequalities $\varphi(\bar{x}^i) \geq \mathtt{DV}^\sigma$ because $\bar{x}^i$ is primal feasible for all $i = 1, \ldots, m+2$ such that $\bar{a}_i > 0$. The second item is similar, applying Theorem 7(iii) with $\hat{\varepsilon}_\infty = 0$. □

4.6 Assessment on sparse optimization

The performance of `PDBM` and `MSM` is tested on 24 instances of problem (21) with $\varphi(x) = \frac{1}{2}\langle x, Qx\rangle + \langle q, x\rangle$. The matrix Q was generated by the Matlab's command `gallery('toeppd',n)`, the vector q components were randomly and uniformly sampled in the interval $[-10, 0]$. The number of linear constraints was set to 10, with A sparsely generated with nonzero components following the standard normal distribution (in Matlab `A = spradn(10,n)`). We set $b = A\tilde{x}$ for a random $\tilde{x} \in \mathbb{R}^n$ with exactly K non-zero components.

We denote by `PDBM-P` and `PDBM-S` the solver `PDBM` applied with the Proximal and Sharp Lagrangians, respectively. The respective initial points were $(y^0, r^0) = (0, 1) \in \mathbb{R}^2$ and $(y^0, r^0) = (0, \ldots, 0, 1 \in \mathbb{R}^{11})$. The benchmark also runs the `MSM` with the Proximal Lagrangian and two variants of the *mixed-binary quadratic programming* (MBQP) solver of Gurobi, applied to

$$\min_{x \in \mathbb{R}^n,\, z \in \{0,1\}^n} \frac{1}{2}\langle Qx, x\rangle + \langle q, x\rangle \quad \text{s.t.} \quad Ax = b,\ 0 \leq x \leq \mathtt{BigM}\, z \ \text{ and } \ \sum_{i=1}^{n} z_i \leq K,$$

with $\mathtt{BigM} = 1000$. There are two variants, `MSM` with a a CPU time limit of 3 minutes for every problem, and `MBQP-T`, with the same time budget to the one required by `PDBM-P`.

Table 2: Mean and standard deviation for Table 6

	PDBM-P		PDBM-S		MSM		MBQP		MBQP-T	
	avg	stdev	avg	stdev	avg	stdev	avg	stdev	avg	stdev
$\Delta\varphi$	0.3	0.5	2.5	2.1	1.4	1.4	1.1	2.1	4.2	8.8
$h(\bar{x})$	2E-05	5E-05	2E-06	2E-06	7E-04	9E-04				
$\#X_0$	111	84	19	5	352	279				
CPU (s)	71	64	20	13	181	1	172	38		

Table 2 summarizes the results in Table 6 in the appendix. `MBQP` always terminated having exhausted the CPU time budget, of 3 minutes. The result with the proximal variant `PDBM-P` are better than those obtained with `PDBM-S`. However, the gain is obtained at the expense of more CPU time in general. Additionally, `PDBM-P` performed better than `MBQP` on the larger instances: `PDBM-P` not only provided approximate solutions of better quality in many cases but also stopped in significantly shorter CPU times.

5 Unit-commitment problems

Unit-Commitment (UC) problems determine the optimal production schedule of a set of generation units in a power system, meeting system-wide constraints and units' dynamics. These are notoriously difficult optimization problems that typically involve mixed 0-1 variables in a large scale setting, but need to be solved accurately in short times. Decomposition methods are therefore particularly suited to UC problems, as explained in [Tah+15], see also [Bor+03], [Fei+15], [SFF18].

In order to be able to compare a direct solution of (1) with the primal-dual bundle Algorithm 2, for our numerical experiments we consider a simplified UC formulation. More precisely, the power system is purely thermal, there is no network, and all data is deterministic. Furthermore, in our UC model given in (27), there are linear generating costs and startup costs spanning a single period. The units have ramp constraints, minimum up and down-times, and startup/shutdown capabilities. The system with I units is optimized over an horizon with T time steps. For unit i, p_i^t is the power generated at time t (MW) and u_i^t its status (0 or 1). Variables $v_i^t, w_i^t \in \{0,1\}$ indicate if the unit was turned on or off at time t. The vector $p_i \in \mathbb{R}^T$ gathers the unit generation over all the horizon, and similarly for the other variables. At each time step, the unit has constant minimum and maximum output power, $\underline{p}_i$ and $\overline{p}_i$ (MW). The ramp rate is Δ_i (MW/h), while TU_i and TD_i are the minimum up and downtimes (hours). Technological constraints are represented by requiring (p_i, u_i, v_i, w_i) to be in $X_i := \mathscr{P}_i \cap \left[0, \overline{p}_i\right]^T \times \{0,1\}^{3T}$, for the sets

$$\mathscr{P}_i := \left\{ (p_i, u_i, v_i, w_i) : \begin{array}{rr} |p_i^t - p_i^{t-1}| \le \Delta p_i\,, & t \in [1,T] \\ p_i^t \le (\overline{p}_i - \underline{p}_i)(u_i^t - v_i^t - w_i^{t+1})\,, & t \in [2,T-1] \\ \sum_{j=t-TU_i+1}^{t} v_i^t \le u_i^t\,, \quad \sum_{j=t-TD_i+1}^{t} w_i^t \le 1 - u_i^t, u_i^t - u_i^{t-1} = v_i^t - w_i^t\,, & t \in [2,T] \end{array} \right\},$$

defined for $i = 1, \ldots, I$. For additional details we refer to [MLR13]. The cost of each unit over the whole horizon is $\varphi_i(p_i, v_i) := \sum_{t=1}^T (C_i p_i^t + F_i v_i^t)$, for a linear variable cost C_i (\$/MWh), and a fixed cost F_i (\$/h).

A unit labeled with subindex 0 accounts for load shedding, modeled as a thermal power plant with very high generating cost and capacity, always switched on. This is represented by the inclusion $p_0 \in \mathscr{P}_0 := \left[0, \overline{p}_0\right]^T$, the "generation" cost $\varphi_0(p_0) := \sum_{t=1}^T C_0 p_0^t$.

Finally, if D^t is the system load at time t (MW), the UC problem is

$$\left\{ \begin{array}{ll} \min & \varphi_0(p_0) + \sum_{i=1}^I \varphi_i(p_i, v_i) \\ \text{s.t.} & p_0 \in \mathscr{P}_0 \\ & (p_i, u_i, v_i, w_i) \in X_i \qquad t \in [1,T]\,, i = 1, \ldots, I\,, \\ & \sum_{i=1}^I p_i^t + p_0^t = D^t \qquad t \in [1,T]\,. \end{array} \right. \tag{27}$$

A well known feature of UC problems is that usually they present a small set of system-wide constraints, "tying" the variables of different units, like the market clearing constraint above. Rather than directly relaxing this constraint, we duplicate the variables p_i^t, as in [Lem+96], and consider

$$\left\{ \begin{array}{ll} \min & \varphi_0(p_0) + \sum_{i=1}^I \varphi_i(p_i, v_i) \\ \text{s.t.} & p_0 \in \mathscr{P}_0 \\ & (p_i, u_i, v_i, w_i) \in X_i \qquad t \in [1,T]\,, i = 1, \ldots, I\,, \\ & pa_i^t \in \left[0, \overline{p}_i\right], \qquad t \in [1,T]\,, i = 1, \ldots, I\,, \\ & pa_i^t - p_i^t = 0\,, \qquad t \in [1,T]\,, i = 1, \ldots, I\,, \\ & \sum_{i=1}^m pa_i^t + p_0^t = D^t \qquad t \in [1,T]\,. \end{array} \right.$$

With this formulation, the coupling constraint in (1) is $h(x) = pa - p$, with dimension $m = TI$. The decision variable x and set X are defined by subcomponents: for $i = 1,\dots,I$, $x_i := (p_i, u_i, v_i, w_i) \in X_i$, and

$$x_0 := (p_0, pa) \in X_0 := \left\{ x_0 \in \mathscr{P}_0 \times \prod_{i=1}^{I} [0, \overline{p}_i]^T : \sum_{i=1}^{I} pa_i^t + p_0^t = D^t, t \in [1,T] \right\}.$$

Accordingly, the primal problem (1) to be solved is

$$\begin{cases} \min \sum_{i=0}^{I} \varphi_i(x_i) \\ \text{s.t. } x_i \in X_i, i = 0,\dots,I \\ \quad h(x) = 0. \end{cases}$$

Then, for dual variables $y \in \mathbb{R}^{TI}$ with components y_i^t and $r \in \mathbb{R}_+$, the proximal Lagrangian gives in (3)

$$\begin{aligned} L(x,y,r) &= \sum_{i=0}^{I} \varphi_i(x_i) + \langle y, h(x) \rangle + r\sigma(h(x)) \\ &= \varphi_0(p_0) + \sum_{i=1}^{I} \left(\varphi_i(p_i, v_i) + \langle y_i, pa_i - p_i \rangle + \frac{r}{2} \|pa_i - p_i\|_2^2 \right) \\ &= \underbrace{\varphi_0(p_0) + \sum_{i=1}^{I} \langle y_i, pa_i \rangle}_{L_0(x_0,y)} + \sum_{i=1}^{I} \underbrace{\varphi_i(p_i, v_i) - \langle y_i, p_i \rangle}_{L_i(x_i,y)} + \frac{r}{2} \sum_{i=1}^{I} \|pa_i - p_i\|_2^2, \end{aligned}$$

to be minimized over the mixed-integer set $X = \prod_{i=0}^{I} X_i$, and where the partial Lagrangians above were defined to explain the approximating procedure in Algorithm 3. The sequential solution put in place induces separability for the augmented Lagrangian in a manner resembling the setting of the *alternating direction method of multipliers* (ADMM), introduced in [GM76]. There are two important differences, however. First, ADMM works with a classical augmented Lagrangian, considering r a penalization, maintained fixed throughout iterations. Second, the dual variable y in ADMM is updated using a subgradient step, instead of a (potentially more reliable) bundle update.

Algorithm 3 SEPARABLE PROCEDURE FOR INEXACT MINIMIZATION OF THE UC LAGRANGIAN

Input and starting point. Given $(y^k, r^k) \in \mathbb{R}^m \times \mathbb{R}_+$, let $pa^0 = pa^\nu$ from iteration $k-1$ if $k > 1$. Otherwise, solve the linear relaxation of (27) and let pa^0 as the solution components corresponding to p.

Repeat for $\nu = 1, 2, \dots$:

Mixed 0-1 subproblems: For $i = 1,\dots,I$ solve $\begin{cases} \min\ L_i(x_i, y^k) + \frac{r^k}{2} \|pa_i^{\nu-1} - p_i\|_2^2 \\ \text{s.t. } x_i = (p_i, u_i, v_i, w_i) \in X_i. \end{cases}$

Let p_i^ν be the solution components corresponding to p_i.

Continuous subproblem: Let (p_0^ν, pa^ν) solve $\begin{cases} \min\ L_0(x_0, y^k) + \frac{r^k}{2} \sum_{i=1}^{I} \|pa_i - p_i^\nu\|_2^2 \\ \text{s.t. } x_0 = (p_0, pa) \in X_0. \end{cases}$

Set $x^\nu := (p_0^\nu, pa^\nu, p_1^\nu, \dots, p_I^\nu)$.

Until $\left| L(x^\nu, y^k, r^k) - L(x^{\nu-1}, y^k, r^k) \right| \le 5 \times 10^{-4} |L^0|$.

Inexact Primal Iterate. x^ν is the approximate solution of $\min_{x \in X} L(x, y^k, r^k)$ in Step 2 of Algorithm 2.

To initialize the dual variables recall that, by construction, the iterates r^k can only increase, so tuning well r^0 is fundamental. We let $y^0 = 0 \in \mathbb{R}^m$, and use the steps below to determine r^0:

1. Given $0 < r_1 < r_2 \ll r_3$, for $\ell = 1,2,3$ let $s_\ell := \sigma(h(x^{\nu(\ell)}))$ where $x^{\nu(\ell)}$ is the output of Algorithm 3 with the pair (y^0, r_ℓ).
2. The coefficients (r_ℓ, s_ℓ) fit the exponential function $\mathscr{S}(r) = ae^{-br} + c$, for $\begin{cases} a = \frac{s_1}{e^{-bs_1}}, \\ b = \frac{\log(s_2) - \log(s_1)}{r_1 - r_2}, \\ c = r_3. \end{cases}$

   The function $\mathscr{R}(r) := r\mathscr{S}(r)$ attains its maximum at $r^* = \frac{1}{b}$.

3. We aim at finding r^0 such that $\mathscr{R}(r^0) = K\varphi^{\texttt{lin}}$, for some constant $K > 0$ and where $\varphi^{\texttt{lin}}$ is the unit-commitment cost obtained when solving the linear relaxation of (27). Accordingly, we set $r^0 = r^*$ if $\mathscr{R}(r^*) < K\varphi^{\texttt{lin}}$. Otherwise, r^0 is the root of $\mathscr{R}(r^*) - K\varphi^0 = 0$ in the interval $[r^*, r_3]$.

In order to run the model in different settings, we generated load profiles spanning from one to seven days, with 1-hour discretization ($T = 24, 48, \ldots, 168$). Additionally, the thermal power system was defined in a manner close to real-life configurations, varying $I \in [3, 10]$. Specifically,

1. A nuclear-like group (20% of the total number of units), with low cost, large maximum power output, but a small range of operation (minimum power output equal to 75% of maximum power output).
2. A slow-dynamics coal group (50%) with medium cost, wider range of operation (minimum power output equal to 8% of maximum power output), but small ramp ($0.1 \cdot (\overline{p}_i - \underline{p}_i)$).
3. A gas-peaker group (30%) with high cost, the widest range of operation (minimum power output equal to 5% of maximum power output), faster dynamics (ramp of $0.25 \cdot (\overline{p}_i - \underline{p}_i)$), but smaller maximum power output.

The benchmark compares `MSM` and `PDBM`, with Python implementations of Algorithms 2 and 3. Both solvers stop with a tolerance measured in MW, $\texttt{Tol} := 5 \cdot 10^{-3} \sum_{i=1}^{T} D^t$. For our instances, this amounts to stopping when the difference between p^k and pa^k is inferior to

$$\{4.6, 8.8, 13.3, 17.7, 22.4, 26.6, 31.0\} \text{ MW, over } \{24, 48, 72, 96, 120, 144, 168\} \text{ hours, respectively.}$$

Table 7 in the appendix reports the optimality gaps and feasibility found with both solvers for 56 instances (the exact value of `PV` is computable for these problems), and a summary is given in Table 3.

Table 3: Mean and standard deviation for Table 7

	PDBM		MSM	
	avg	stdev	avg	stdev
$Gap(\%)$	4.5	6.5	14.6	12.1
$h(\bar{x})$	4.5E-03	7.2E-04	5.3E-03	1.1E-03
$\#X_0$	105	52	208	131
CPU (s)	96	121	148	140

For `PDBM` we observe a gap smaller than 5% in 37 runs, while `MSM` attained such condition in 20 runs, as shown on the left plot in Figure 1. `PDBM` consistently had shorter CPU times and fewer oracle calls. Regarding the former, the performance on the right of Figure 1 shows that `PDBM` performed better than `MSM` in around 80% of the instances.

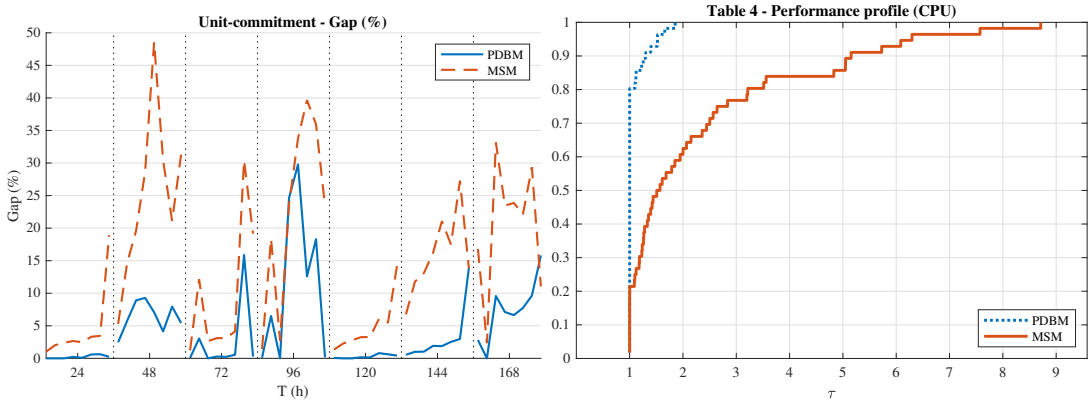

Figure 1: Optimality gap (left) and profile of CPU time (right) for unit-commitment problems.

6 Conclusions

It is interesting to relate our results with those in [FK00, Section 6], stated in an exact setting as in our conceptual scheme. Theorem 6.2(ii) therein shows that the relabeled bundle elements $(\mathtt{a}^i, x^i)$ solve in the limit the following relaxation of (1), on variables $(\tilde{\mathtt{a}}^i, \tilde{x}^i)$ for $i = 1, \dots, m+2$:

$$\min \sum_{i=1}^{m+2} \tilde{\mathtt{a}}^i \varphi(\tilde{x}^i) \;\text{ s.t. }\; \tilde{x}^i \in X\,,\; \sum_{i=1}^{m+2} \tilde{\mathtt{a}}^i h(\tilde{x}^i) = 0\,, \tilde{\mathtt{a}} \in \Delta^{m+2}\,.$$

By examining the proof in Theorem 7, we see that our analysis not also extends the result to inexact calculations but also, thanks to the special Lagrangian (3), improves the statement. More precisely, in our setting eventually the elements $(\mathtt{a}^i, x^i)$ minimize, approximately if calculations are inexact, the relaxation

$$\min \sum_{i=1}^{m+2} \tilde{\mathtt{a}}^i \varphi(\tilde{x}^i) \;\text{ s.t. }\; \tilde{x}^i \in X\,, h(\tilde{x}^i) = 0 \text{ for } i = 1, \dots, m+2\,, \tilde{\mathtt{a}} \in \Delta^{m+2}\,.$$

Moreover, if calculations are asymptotically exact, then Corollary 2 yields primal optimality for the limit of the serious primal points, $\hat{x}^\infty$.

This perspective reveals primal feasibility as one of the strengths of the augmented Lagrangian. Its weakness is the loss of separability, that we addressed somehow in Algorithm 2 by working with inexact primal iterates. Our take-away advice is to work with a classical Lagrangian if primal feasibility is less of a concern than separability, and use the augmented Lagrangian (3) otherwise.

References

[BOK19] A. M. Bagirov, G. Ozturk and R. Kasimbeyli. 'A sharp augmented Lagrangian-based method in constrained non-convex optimization'. In: *Optimization Methods and Software* 34.3 (2019), pp. 462–488.

[BE15] N. L. Boland and A. C. Eberhard. 'On the augmented Lagrangian dual for integer programming'. In: *Mathematical Programming* 150.2 (2015), pp. 491–509.

[Bon+06] J. Bonnans, J. Gilbert, C. Lemaréchal and C. Sagastizábal. *Numerical Optimization. Theoretical and Practical Aspects*. Universitext. Second edition, xiv+490 pp. Berlin: Springer-Verlag, 2006.

[Bor+03] A. Borghetti, A. Frangioni, F. Lacalandra and C. Nucci. 'Lagrangian Heuristics Based on Disaggregated Bundle Methods for Hydrothermal Unit Commitment'. In: *IEEE Transactions on Power Systems* 18 (2003), pp. 313–323.

[BIM13] R. Burachik, A. Iusem and J. Melo. 'An Inexact Modified Subgradient Algorithm for Primal-Dual Problems via Augmented Lagrangians'. In: *Journal of Optimization Theory Applications* 157.1 (2013), pp. 108–131.

[Bur11] R. S. Burachik. 'On primal convergence for augmented Lagrangian duality'. In: *Optimization* 60.8-9 (2011), pp. 979–990.

[Bur+06] R. S. Burachik, R. N. Gasimov, N. A. Ismayilova and C. Y. Kaya. 'On a Modified Subgradient Algorithm for Dual Problems via Sharp Augmented Lagrangian'. In: *J. Global Optimization* 34.1 (2006), pp. 55–78.

[Eve63] H. Everett. 'Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources'. In: *Operations Research* 11.3 (1963), pp. 399–417.

[FAS17] M. J. Feizollahi, S. Ahmed and A. Sun. 'Exact augmented Lagrangian duality for mixed integer linear programming'. In: *Mathematical Programming* 161.1 (2017), pp. 365–387.

[Fei+15] M. J. Feizollahi, M. Costley, S. Ahmed and S. Grijalva. 'Large-scale decentralized unit commitment'. en. In: *International Journal of Electrical Power & Energy Systems* 73 (Dec. 2015), pp. 97–106.

[FK00] S. Feltenmark and K. Kiwiel. 'Dual Applications of Proximal Bundle Methods, Including Lagrangian Relaxation of NonConvex Problems'. In: *SIAM Journal on Optimization* 10.3 (2000), pp. 697–721.

[Fra02] A. Frangioni. 'Generalized Bundle Methods'. In: *SIAM Journal on Optimization* 13.1 (2002), pp. 117–156.

[GM76] D. Gabay and B. Mercier. 'A dual algorithm for the solution of nonlinear variational problems via finite element approximation'. In: *Computers & Mathematics with Applications* 2.1 (1976), pp. 17–40.

[Gas02] R. N. Gasimov. 'Augmented Lagrangian Duality and Nondifferentiable Optimization Methods in Nonconvex Programming'. In: *Journal of Global Optimization* 24.2 (2002), pp. 187–203.

[GAD20] X. Gu, S. Ahmed and S. S. Dey. 'Exact Augmented Lagrangian Duality for Mixed Integer Quadratic Programming'. In: *SIAM Journal on Optimization* 30.1 (2020), pp. 781–797.

[Hin01] M. Hintermüller. 'A Proximal Bundle Method Based on Approximate Subgradients'. In: *Computational Optimization and Applications* 20 (3 2001), pp. 245–266.

[HL96] J. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. 2nd. Grundlehren der mathematischen Wissenschaften 306. Springer-Verlag, 1996, p. 347.

[JP94] J. J. Júdice and F. M. Pires. 'A block principal pivoting algorithm for large-scale strictly monotone linear complementarity problems'. In: *Computers & Operations Research* 21.5 (1994), pp. 587–596.

[Kiw06] K. Kiwiel. 'A proximal bundle method with approximate subgradient linearizations'. In: *SIAM Journal on Optimization* 16.4 (2006), pp. 1007–1023.

[Kiw] K. C. Kiwiel. 'A Method for Solving Certain Quadratic Programming Problems Arising in Nonsmooth Optimization'. In: *IMA Journal of Numerical Analysis* 6.2 (), pp. 137–152.

[Lem+96] C. Lemaréchal, F. Pellegrino, A. Renaud and C. Sagastizábal. 'Bundle methods applied to the unit-commitment problem'. In: *System Modelling and Optimization*. Ed. by J. Doležal and J. Fidler. Chapman and Hall, 1996, pp. 395–402.

[MLR13] G. Morales-España, J. M. Latorre and A. Ramos. 'Tight and Compact MILP Formulation for the Thermal Unit Commitment Problem'. In: *IEEE Transactions on Power Systems* 28.4 (Nov. 2013). Conference Name: IEEE Transactions on Power Systems, pp. 4897–4908.

[NOR07] A. Nedić, A. Ozdaglar and A. Rubinov. 'Abstract convexity for nonconvex optimization duality'. In: *Optimization* 56.5-6 (2007), pp. 655–674.

[OSL] W. Oliveira, C. Sagastizábal and C. Lemaréchal. 'Convex proximal bundle methods in depth: a unified analysis for inexact oracles'. In: *Mathematical Programming* 148.1-2 (), pp. 241–277.

[OS14] W. de Oliveira and C. Sagastizábal. 'Level bundle methods for oracles with on-demand accuracy'. In: *Optimization Methods and Software* 29.6 (2014), pp. 1180–1209.

[OS20] W. de Oliveira and M. Solodov. 'Bundle Methods for Inexact Data'. In: *Numerical Nonsmooth Optimization: State of the Art Algorithms*. Ed. by A. M. Bagirov, M. Gaudioso, N. Karmitsa, M. M. Mäkelä and S. Taheri. Cham: Springer International Publishing, 2020, pp. 417–459.

[OT19] W. de Oliveira and M. P. Tcheou. 'An Inertial Algorithm for DC Programming'. In: *Set-Valued and Variational Analysis* 27.4 (2019), pp. 895–919.

[Pan+17] L. Pan, S. Zhou, N. Xiu and H. Qi. 'A convergent iterative hard thresholding for nonnegative sparsity optimization'. In: *Pacific Journal of Optimization* 13.2 (2017), pp. 325–353.

[RW98] R. Rockafellar and R. Wets. *Variational Analysis*. Die Grundlehren der mathematischen Wissenschaften in Einzeldarstellungen. Springer, 1998.

[Sag12] C. Sagastizábal. 'Divide to Conquer: Decomposition Methods for Energy Optimization'. In: *Mathematical Programming* 134.1 (2012), pp. 187–222.

[SFF18] M. R. Scuzziato, E. C. Finardi and A. Frangioni. 'Comparing Spatial and Scenario Decomposition for Stochastic Hydrothermal Unit Commitment Problems'. In: *IEEE Transactions on Sustainable Energy* 9.3 (2018), pp. 1307–1317.

[SNH83] J.-J. Strodiot, V. H. Nguyen and N. Heukemes. 'ε-Optimal solutions in nondifferentiable convex programming and some related questions'. In: *Mathematical Programming* 25.3 (1983), pp. 307–328.

[Tah+15] M. Tahanan, W. v. Ackooij, A. Frangioni and F. Lacalandra. 'Large-scale Unit Commitment under uncertainty'. en. In: *4OR* 13.2 (Jan. 2015), pp. 115–171.

[Tuy16] H. Tuy. *Convex Analysis and Global Optimization*. 2nd. Springer Optimization and Its Applications. Springer, 2016, pp. XVI, 505.

A Tables with results

Table 4: Case 1 - enumerative strategy versus solvers PDBM and MSM equipped with proximal Lagrangian ($aE-b$ stands for $a\,10^{-b}$ and (500*) that the maximum number of iterations was reached).

		$\varphi(\bar{x})$			$h(\bar{x})$		$\#X_0$		CPU(s)		
	N	ΔPDBM	ΔMSM	ENUM	PDBM	MSM	PDBM	MSM	PDBM	MSM	ENUM
$n=50$											
	100	-0.01	-0.01	-738.71	9E−8	1E−7	15	135	0.2	0.9	0.4
	100	0	0	-818.16	5E−8	9E−8	15	107	0.1	0.7	0.4
	500	0	0	-898.40	4E−8	9E−8	15	108	0.7	4.5	1.8
	500	0	-0.63	-877.03	5E−8	3E−2	142	500*	6.6	20.5	1.7
$n=100$											
	100	0	0	-54.96	2E−8	9E−8	13	64	0.1	0.2	0.7
	100	0	0	-42.71	1E−8	9E−8	13	61	0.1	0.2	0.7
	500	-0.01	-0.01	-1698.29	7E−8	9E−8	15	133	0.7	6.2	3.7
	500	0	0	-1649.18	4E−8	9E−8	15	98	0.7	4.7	3.6
$n=200$											
	100	0	0	-46.10	6E−9	8E−8	13	58	0.1	0.4	1.6
	100	0	0	-36.92	1E−8	8E−8	13	60	0.1	0.4	1.6
	500	-0.01	-0.01	-3542.61	5E−8	9E−8	15	102	0.9	5.3	8.5
	500	0	0	-2741.57	4E−8	9E−8	15	96	0.8	4.9	8.6
$n=300$											
	100	0	0	-29.75	9E−9	9E−8	13	59	0.2	0.8	6.5
	100	0	0	-37.79	1E−8	1E−7	13	59	0.2	0.7	6.6
	500	0	0	-4343.54	6E−8	9E−8	15	106	1.0	6.2	31.9
	500	0	-1.2	-3941.44	5E−8	1E−2	199	500*	12.1	29.4	31.4
$n=500$											
	100	0	0	-34.31	9E−9	9E−8	13	59	0.5	2.3	14.2
	100	0	0	-27.76	8E−9	8E−8	13	59	0.5	2.2	14.3
	500	0	0	-34.31	9E−9	9E−8	13	59	1.2	5.1	71.2
	500	0	0	-30.98	8E−9	8E−8	13	59	1.2	5.1	72.6
$n=1000$											
	100	0	0	-32.54	6E−9	9E−8	13	57	2.4	10.9	68.7
	100	0	0	-30.14	6E−9	9E−8	13	57	2.6	10.7	67.5
	500	0	0	-41.24	6E−9	9E−8	13	57	7.1	32.2	344.4
	500	0	0	-34.16	6E−9	1E−7	13	57	7.0	30.8	339.3
$n=2000$											
	100	0	0	-91.48	9E−8	1E−7	12	54	13.3	60.9	409.7
	100	0	0	-66.76	1E−7	9E−8	12	55	14.5	65.4	383.6
	500	0	0	-91.48	9E−8	1E−7	12	54	29.3	133.3	2038.8
	500	0	0	-71.46	1E−7	9E−8	12	55	29.1	133.7	1914.2

Table 5: Case 2 - enumerative strategy versus solver PDBM and two starting points.

		$\varphi(\bar{x})$			$h(\bar{x})$		$\#X_0$		CPU(s)		
	N	ΔPDBM10	ΔPDBM100	ENUM	PDBM10	PDBM100	PDBM10	PDBM100	PDBM10	PDBM100	ENUM
$n = 50$											
	100	0	0.01	-202.98	5E−8	7E−8	130	19	5.8	1.3	0.6
	100	0	0.01	-145.81	6E−8	8E−9	135	19	6.3	1.2	0.6
	500	0	0.01	-273.46	2E−8	8E−8	26	19	5.3	5.9	3.0
	500	0.01	0	-203.64	6E−8	1E−8	214	19	46.0	7.0	2.9
$n = 100$											
	100	0	0	-29.94	8E−8	2E−8	13	17	0.9	1.4	1.6
	100	0	0	-25.64	3E−8	2E−8	14	17	1.0	1.3	1.7
	500	0	-0.01	-390.20	8E−8	8E−8	168	311	167.4	498.7	8.1
	500	0	0	-384.83	1E−8	4E−9	193	19	144.3	29.8	8.0
$n = 200$											
	100	0	0	-23.34	2E−8	1E−8	13	17	2.5	4.4	4.8
	100	0	0	1.68	3E−8	1E−8	13	17	2.4	3.3	4.7
	500	0	0	-699.89	7E−8	5E−10	17	19	29.3	150.4	23.3
	500	0	0	-718.81	1E−8	8E−10	20	19	31.5	71.3	22.2
$n = 300$											
	100	0.01	0.01	19.20	3E−8	1E−8	13	17	4.3	5.8	13.1
	100	0	0	19.71	3E−8	1E−8	13	17	4.0	5.4	12.3
	500	0	0	-994.26	4E−8	1E−9	21	19	52.5	99.9	65.1
	500	0.01	0	-1232.18	1E−7	8E−10	17	19	44.1	124.6	61.4
$n = 500$											
	100	0	0	31.94	7E−8	1E−8	182	17	162.6	15.4	36.9
	100	0	0	37.70	2E−8	1E−8	13	17	13.0	17.5	37.3
	500	0	0	30.31	2E−8	1E−7	13	16	55.9	70.2	184.1
	500	0	0	34.16	2E−8	9E−9	13	17	66.7	85.3	187.8
$n = 1000$											
	100	0	0	74.94	2E−8	8E−8	13	16	60.2	73.9	223.7
	100	0	0	105.36	7E−8	9E−9	291	17	1303.6	78.1	220.8
	500	0	0	69.36	2E−8	8E−8	13	16	286.0	361.0	1115.7
	500	0	0	102.49	2E−8	9E−9	13	17	277.7	372.2	1109.3
$n = 2000$											
	100	0.01	0	128.43	8E−9	4E−8	13	16	243.7	302.4	1434.6
	100	0	0	148.07	9E−9	4E−8	13	16	242.8	296.4	1456.7
	500	0	0	125.81	8E−9	4E−8	13	16	1149.5	1400.4	7197.8
	500	0	0	136.29	9E−9	4E−8	13	16	1151.1	1414.4	7397.7

Table 6: Comparison of 5 solvers to compute a solution of (21).

		$\Delta\varphi(\bar{x})$						$h(\bar{x})$			$\#X_\varepsilon$			CPU(s)			
	$p\%$	best	PDBM-P	PDBM-S	MSM	MBQP	MBQP-T	PDBM-P	PDBM-S	MSM	PDBM-P	PDBM-S	MSM	PDBM-P	PDBM-S	MSM	MBQP
$n = 500$																	
	10	-9.4	0.4	4.6	2.5	-	5.8	9E−6	5E−10	1E−3	28	21	882	3	6	180*	180*
	15	-15.8	-	3.8	2.6	1	2.3	9E−6	9E−7	2E−4	68	20	777	7	6	180*	180*
	20	-21.2	-	1.7	2.3	-	2	7E−6	6E−8	7E−4	230	19	712	41	6	180*	180*
	25	-25.8	-	1.1	1.7	0.1	1.4	9E−6	1E−9	5E−4	198	19	648	38	6	180*	180*
$n = 600$																	
	10	0.5	0.4	5.2	2.5	-	8.2	7E−6	6E−7	5E−4	31	20	417	3	8	180*	180*
	15	-12	0.1	1.2	0.6	-	21.7	2E−6	2E−9	2E−4	29	20	653	3	7	180*	180*
	20	-15.8	-	0.6	0.5	-	0.3	5E−7	2E−6	1E−4	148	20	648	21	7	180*	180*
	25	-19.3	-	0.3	0.7	0.2	0.8	7E−6	1E−7	4E−5	45	19	701	5	6	180*	180*
$n = 700$																	
	10	-5.8	0.1	1.1	0.2	-	0.2	1E−5	5E−6	3E−4	278	20	826	93	10	180*	180*
	15	-10.8	0.4	0.6	0.6	-	0.4	1E−5	6E−7	4E−4	213	20	829	47	9	180*	180*
	20	-10.7	0.3	0.4	0.3	-	7	9E−7	3E−9	2E−4	10	20	655	1	8	180*	180*
	25	-11.9	0.8	0.9	0.9	-	0.2	8E−6	5E−6	4E−5	298	19	532	60	8	180*	180*
$n = 800$																	
	10	-7.7	-	2.4	1.6	0.7	3.2	5E−6	5E−7	8E−4	70	21	526	18	14	180*	180*
	15	-13.2	-	1.4	1	0.2	0.7	1E−5	5E−6	3E−4	79	19	549	20	13	180*	180*
	20	-16.4	-	0.4	0.3	-	0.4	5E−6	4E−7	4E−4	132	20	590	25	12	181*	180*
	25	-19	0.7	0.7	0.7	-	0.9	5E−6	2E−7	2E−4	47	20	517	13	11	181*	180*
$n = 900$																	
	10	-8.4	0.2	0.7	0.1	-	-	8E−5	1E−8	4E−4	232	21	618	182*	17	180*	180*
	15	-11.5	0.1	0.4	0.1	-	0.1	1E−5	2E−7	3E−4	142	21	498	44	17	180*	180*
	20	-15.7	1.2	1.3	1.2	-	1	9E−6	3E−6	2E−4	151	20	450	51	14	180*	180*
	25	-16.4	1.3	1.5	1.4	-	-	7E−6	4E−6	2E−4	253	20	439	107	14	181*	2
$n = 1000$																	
	10	-8.1	-	1.7	0.1	0.5	0.8	9E−6	8E−7	4E−4	89	21	418	31	22	180*	180*
	15	-12.6	0.2	0.5	0.1	-	0.3	6E−6	1E−6	4E−4	120	21	381	54	19	181*	180*
	20	-16.3	1.6	1.9	1.6	-	0.3	1E−5	6E−6	2E−4	174	19	353	112	17	181*	180*
	25	-17.7	1.4	1.5	1.4	-	-	7E−6	5E−6	3E−5	162	20	335	81	17	181*	2
$n = 1100$																	
	10	-10.2	-	4.2	3	2.6	2.6	2E−5	2E−6	8E−4	319	21	120	180*	28	181*	180*
	15	-18.2	-	2.8	2	2	2	4E−6	6E−7	9E−4	198	21	112	175	27	182*	180*
	20	-25.6	-	1.1	1	1	1.5	3E−6	8E−7	3E−4	31	21	107	17	26	181*	180*
	25	-31	-	1.3	0.8	2.4	3.2	9E−6	2E−7	2E−4	151	21	96	97	26	182*	180*
$n = 1200$																	
	10	-11.8	-	5.4	3.6	5.1	5.1	1E−5	7E−6	6E−4	115	20	83	105	19	180*	180*
	15	-19.9	-	4.4	2.5	2.9	2.9	2E−4	5E−7	4E−4	60	21	74	181*	29	180*	180*
	20	-34.5	-	7.4	5.2	7	6.1	1E−4	1E−7	4E−4	70	21	71	185*	32	181*	181*
	25	-48.3	-	6.6	7.1	10.1	10.1	4E−5	7E−10	2E−4	67	21	69	181*	36	181*	180*
$n = 1300$																	
	10	-10.7	-	2.2	1.3	0.3	3.3	9E−7	9E−7	4E−4	20	22	108	15	40	182*	180*
	15	-17	-	1.3	1.1	0.3	0.5	6E−6	8E−7	1E−4	83	21	86	69	38	182*	180*
	20	-23	0.4	0.8	1	-	-	1E−5	3E−6	9E−4	118	21	81	182*	36	181*	181*
	25	-27.2	-	1	1.3	0.6	5.7	3E−6	9E−7	4E−4	27	21	75	19	34	183*	180*
$n = 1400$																	
	10	-12.1	-	6.4	1.9	1.8	24	2E−6	1E−20	1E−3	25	2	72	18	11	184*	181*
	15	-19.9	0.3	4.9	0.5	-	0.1	1E−5	4E−13	9E−4	50	2	66	77	11	180*	180*
	20	-31.9	2.5	6.4	3.1	-	0.9	5E−6	1E−6	1E−3	25	22	49	49	30	180*	181*
	25	-41.9	-	1.3	1.1	1.1	1.1	2E−4	6E−6	9E−4	73	20	50	185*	41	184*	181*
$n = 1500$																	
	10	-11.8	-	3.8	0.5	0.7	0.7	9E−5	1E−8	1E−3	116	2	77	181*	11	182*	181*
	15	-21	-	4.5	0.9	2.1	3.7	1E−6	6E−6	2E−3	44	21	57	54	27	180*	181*
	20	-30.7	0.5	4.7	2.1	-	2.4	7E−6	9E−7	4E−3	28	22	45	57	43	180*	181*
	25	-42.1	0.2	4.7	-	5.5	51.5	1E−5	2E−6	5E−3	20	21	42	20	54	183*	181*

Table 7: Relative optimality gap and feasibility for unit-commitment problems (27).

		Gap (%)		$h(\bar{x})$		$\#X_\varepsilon$		CPU (s)	
	I	PDBM	MSM	PDBM	MSM	PDBM	MSM	PDBM	MSM
$T = 24$h									
	3	0.0	1.0	2.0E−3	5.0E−3	8	88	0.7	5.3
	4	0.0	2.0	4.5E−3	5.0E−3	98	91	7.4	8.3
	5	0.0	2.4	4.3E−3	4.9E−3	16	123	2.1	10.6
	6	0.2	2.7	4.8E−3	5.0E−3	181	153	17.5	19.2
	7	0.1	2.5	4.9E−3	5.0E−3	162	168	20.4	22.2
	8	0.6	3.3	4.9E−3	5.0E−3	166	198	21.1	26.7
	9	0.6	3.5	4.9E−3	5.0E−3	178	215	24.7	31.6
	10	0.2	18.9	4.9E−3	9.6E−3	65	500	15.3	93.1
$T = 48$h									
	3	2.5	5.3	4.9E−3	5.0E−3	72	121	12.7	15.6
	4	5.8	14.6	3.7E−3	4.5E−3	70	220	17.6	43.0
	5	8.9	19.6	4.6E−3	5.0E−3	91	363	24.1	77.5
	6	9.3	28.6	4.5E−3	5.8E−3	82	500	29.4	142.0
	7	7.1	48.6	4.9E−3	1.0E−2	59	500	28.0	176.3
	8	4.1	30.4	4.6E−3	7.4E−3	56	500	21.4	186.4
	9	7.9	21.1	4.9E−3	6.0E−3	112	500	54.1	192.6
	10	5.4	31.3	4.4E−3	9.7E−3	170	504	79.3	209.3
$T = 72$h									
	3	0.1	1.2	2.5E−3	5.0E−3	7	55	2.0	10.1
	4	3.0	12.2	3.7E−3	5.0E−3	107	110	45.9	40.9
	5	0.0	2.7	4.8E−3	5.0E−3	98	123	23.5	29.6
	6	0.3	3.1	4.8E−3	5.0E−3	134	148	38.7	52.4
	7	0.2	3.1	4.8E−3	5.0E−3	128	180	42.9	64.7
	8	0.6	4.1	4.8E−3	5.0E−3	142	216	56.2	100.5
	9	15.9	30.4	4.5E−3	5.0E−3	190	323	131.1	220.7
	10	0.3	19.2	4.8E−3	5.0E−3	62	341	65.7	230.9
$T = 96$h									
	3	0.1	1.4	2.1E−3	4.7E−3	7	50	2.6	13.4
	4	6.5	18.4	4.9E−3	5.0E−3	107	158	69.6	97.0
	5	0.0	2.7	4.8E−3	5.0E−3	121	106	40.0	35.8
	6	24.6	23.7	3.4E−3	5.0E−3	134	135	166.9	109.6
	7	29.8	33.8	4.5E−3	5.0E−3	139	137	215.6	116.3
	8	12.6	39.7	4.9E−3	5.0E−3	79	284	121.9	345.7
	9	18.3	36.0	4.1E−3	5.0E−3	213	253	256.0	256.5
	10	0.2	23.4	4.9E−3	5.0E−3	83	317	113.4	291.0
$T = 120$h									
	3	0.1	1.3	2.4E−3	4.4E−3	8	51	3.3	18.9
	4	0.0	2.3	4.9E−3	5.0E−3	66	69	35.7	32.4
	5	0.0	2.8	4.8E−3	5.0E−3	117	87	48.9	39.9
	6	0.2	3.3	4.7E−3	5.0E−3	113	123	63.4	78.4
	7	0.1	3.3	4.7E−3	5.0E−3	139	151	83.1	110.7
	8	0.8	6.1	4.7E−3	5.0E−3	149	218	115.1	224.1
	9	0.6	5.5	4.8E−3	5.0E−3	164	260	133.9	268.0
	10	0.4	14.2	4.9E−3	5.0E−3	54	294	122.8	392.9
$T = 144$h									
	3	0.6	6.8	4.8E−3	5.0E−3	58	29	21.4	12.9
	4	1.0	11.8	4.9E−3	5.0E−3	60	93	42.5	50.3
	5	1.0	13.1	4.9E−3	5.0E−3	69	129	55.5	89.4
	6	1.9	16.2	4.9E−3	5.0E−3	83	190	85.5	183.9
	7	1.9	21.1	4.9E−3	5.0E−3	91	175	103.0	190.4
	8	2.5	17.5	5.0E−3	4.9E−3	101	227	121.5	304.1
	9	3.0	27.3	5.0E−3	5.0E−3	115	270	184.1	380.5
	10	13.9	13.8	4.9E−3	4.7E−3	193	208	446.5	343.4
$T = 168$h									
	3	2.8	16.8	4.3E−3	5.0E−3	38	59	33.0	46.9
	4	0.0	2.3	4.9E−3	5.0E−3	69	57	52.1	37.3
	5	9.6	33.1	4.8E−3	5.0E−3	88	211	103.0	243.1
	6	7.1	23.4	4.8E−3	5.0E−3	125	105	228.6	150.2
	7	6.6	23.9	5.0E−3	5.0E−3	126	170	220.7	259.9
	8	7.7	22.1	4.9E−3	5.0E−3	138	232	300.7	432.7
	9	9.6	29.3	5.0E−3	5.0E−3	153	327	369.4	579.7
	10	15.8	11.0	4.9E−3	5.0E−3	196	243	652.1	519.9