Deriving Solution Value Bounds from the ADMM*

Jonathan Eckstein †

November 23, 2019

Abstract

This short paper describes a simple subgradient-based techniques for deriving bounds on the optimal solution value when using the ADMM to solve convex optimization problems. The technique requires a bound on the magnitude of some optimal solution vector, but is otherwise completely general. Some computational examples using LASSO problems demonstrate that the technique can produce steadily converging bounds in situations in which standard Lagrangian bounds yield little or no useful information.

1 Introduction

Consider a structured convex optimization problem of the form

$$\min_{x\in\mathbb{R}^n} f(x) + g(Mx) \tag{1}$$

where M is an $r \times n$ real matrix and $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $g : \mathbb{R}^r \to \mathbb{R} \cup \{+\infty\}$ are closed proper convex functions. A currently prominent method for solving such problems is the alternating direction method of multipliers (ADMM) [7, 8, 5, 6], which, for some fixed scalar parameter $c > 0$, generates sequences $\{x^k\} \subset \mathbb{R}^n$ and $\{z^k\}, \{p^k\} \subset \mathbb{R}^r$ according to the following recursions:

$$x^{k+1} \in \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \left\{ f(x) + \langle p, Mx\rangle + \tfrac{c}{2}\left\|Mx - z^k\right\|^2 \right\} \tag{2}$$

$$z^{k+1} = \operatorname*{arg\,min}_{z\in\mathbb{R}^r} \left\{ g(z) - \langle p, z\rangle + \tfrac{c}{2}\left\|Mx^{k+1} - z\right\|^2 \right\} \tag{3}$$

$$p^{k+1} = p^k + c(Mx^{k+1} - z^{k+1}). \tag{4}$$

Various forms of overrelaxation are possible for this method [7, 5], but for simplicity we disregard them here.

*This work was supported in part by NSF grant CCF-1617617, Computing and Communications Foundations, CISE directorate, and Air Force Office of Scientific Research (AFOSR) grant FA95550-15-1-0251. The author would also like to thank Patrick Johnstone and Paulo J.S. Silva for some discussions relating to the LASSO example.

†Department of Management Science and Information Systems and RUTCOR, 100 Rockafeller Road, Livingston Campus, Rutgers University, Piscataway NJ 08854 USA, jeckstei@business.rutgers.edu.

Compared to other Lagrangian-based methods, one drawback of the ADMM is that it does not immediately provide lower bounds on the optimal solution value, which can be desirable in many applications. By adding a possibly significant amount of work to the ADMM iteration, it is in some cases possible to compute a Lagrangian bound, as described below in Section 2. In many other cases, however, this technique does not yield useful information. This short paper describes an alternative procedure that can provide useful and steadily converging bounds, so long as there is some upper limit available on the magnitude of some optimal solution vector. The key inequality used is equivalent to (3.11) in [2]; however, its potential was not fully developed there.

2 Lagrangian Bounds

The dual function $q : \mathbb{R}^r \to \mathbb{R} \cup \{-\infty\}$ of (1) is given by

$$\begin{aligned} q(p) &= \min_{\substack{x\in\mathbb{R}^n \\ z\in\mathbb{R}^r}} \big\{ f(x) + g(z) + \langle p, Mx - z\rangle \big\} \\ &= \min_{x\in\mathbb{R}^n} \big\{ f(x) + \langle M^\top p, x\rangle \big\} + \min_{z\in\mathbb{R}^r} \big\{ g(z) - \langle p, z\rangle \big\} \qquad (5) \\ &= -f^*(-M^\top p) - g^*(p), \qquad (6) \end{aligned}$$

where f^* and g^* are the respective convex conjugate functions of f and g. By weak duality, or equivalently Fenchel's inequality, $q(p)$ is a lower bound on the value of any solution of (1) for any $p \in \mathbb{R}^r$. When $q(p)$ is finite, we say that p is *dual feasible*; however, for an arbitrary choice of $p \in \mathbb{R}^r$ it is possible that either $f^*(M^\top p) = +\infty$ or $g^*(p) = +\infty$, in which case $q(p) = -\infty$ and $q(p)$ does not provide a useful bound on the optimal value.

The most straightforward approach to enhancing the ADMM to provide solution bounds is to compute $q(p^k)$ for each dual iterate p^k computed in step (4). In the context of the progressive hedging (PH) algorithm for stochastic programming [11], this is essentially the strategy proposed in [9], with some refinements developed subsequently in [1] in the case of integer variables; the PH algorithm may be viewed as an application of the ADMM in a space with an appropriately formulated weighted inner product.

Letting ∂g denote the subgradient map of the function g, a standard convex-analytic necessary and sufficient optimality condition for the minimization in (3) is

$$0 \in \partial g(z^{k+1}) - p^k + c(z^{k+1} - Mx^{k+1}) = \partial g(z^{k+1}) - p^{k+1}, \qquad (7)$$

where the equality uses (4). It follows that z^{k+1} solves the second minimization in (3) for $p = p^{k+1}$ and

$$-g^*(p^{k+1}) = g(z^{k+1}) - \langle p^{k+1}, z^{k+1}\rangle. \qquad (8)$$

Applying a similar analysis to (2) results in

$$\begin{aligned} 0 &\in \partial f(x^{k+1}) + M^\top p + cM^\top(Mx^{k+1} - z^k) \\ &= \partial f(x^{k+1}) + M^\top\big(p + c(Mx^{k+1} - z^k)\big) \\ &= \partial f(x^{k+1}) + M^\top p^{k+1/2}, \qquad (9) \end{aligned}$$

where we define $p^{k+1/2} = p^k + c(Mx^{k+1} - z^k)$ for all $k \geq 0$. It follows that x^{k+1} solves the first minimization in (5) for $p = p^{k+1/2}$ and

$$-f^*(M^\top p^{k+1/2}) = f(x^{k+1}) + \langle M^\top p^{k+1/2}, x^{k+1}\rangle = f(x^{k+1}) + \langle p^{k+1/2}, Mx^{k+1}\rangle. \tag{10}$$

As an aside, neither (8) nor (10) is a coincidence; both are consequences of the ADMM being an application of the Douglas-Rachford splitting algorithm to minimize $-q = f^* \circ (-M^\top) + g^*$, as first pointed out in [8]; see also [5], as well as [6] for a simplified presentation.

Regrettably, $p^{k+1/2}$ and p^{k+1} are different vectors unless $z^{k+1} = z^k$, so (10) is generally not helpful in computing $q(p^{k+1})$. Instead, we must directly compute

$$\hat{f}_k \doteq \inf_{x \in \mathbb{R}^n} \left\{ f(x) + \langle M^\top p^{k+1}, x\rangle \right\} \tag{11}$$

and then

$$q(p^{k+1}) = \hat{f}_k + g(z^{k+1}) - \langle p^{k+1}, z^{k+1}\rangle, \tag{12}$$

which then provides a lower bound on the optimal solution value F^* of (1). However, it may happen that $\hat{f}_k = -\infty$, in which case one does not obtain a useful bound. Furthermore, computing (11) may add considerable effort to the ADMM iteration, depending on the function f.

A complementary strategy would be to leverage (10) to compute $q(p^{k+1/2})$. In this case we must instead compute

$$\hat{g}_k \doteq \inf_{z \in \mathbb{R}^r} \left\{ g(z) - \langle p^{k+1/2}, z\rangle \right\} \tag{13}$$

and then

$$q(p^{k+1/2}) = f(x^{k+1}) + \langle p^{k+1/2}, Mx^{k+1}\rangle + \hat{g}_k, \tag{14}$$

thus obtaining a bound on F^*. This paper will henceforth call $q(p^{k+1/2})$ the *mid-step* Lagrangian bound, to distinguish it from the Lagrangian bound $q(p^{k+1})$. Clearly the same potential drawbacks apply to this bound, but with respect to the function g rather than f: calculating it might add significantly to the amount of computation at each iteration, and it will not yield useful information when $\hat{g}_k = -\infty$.

2.1 Problematic Functions for Lagrangian Bounds

2.1.1 Quadratic Loss Functions

In many applications, f is chosen to be a loss function of the form

$$f(x) = \tfrac{1}{2} \|Ax - b\|^2, \tag{15}$$

where A is an $m \times n$ real observation matrix and $b \in \mathbb{R}^m$ is a response vector. In this case, the calculation (11) is involves minimizing the positive semidefinite quadratic function

$$\begin{aligned} f_k(x) &\doteq \tfrac{1}{2} \|Ax - b\|^2 + \langle M^\top p^{k+1}, x\rangle \\ &= \tfrac{1}{2} x^\top A^\top A x + \langle M^\top p^{k+1} - A^\top b, x\rangle + \|b\|^2. \end{aligned}$$

This function is convex and continuous, so minimizing it is equivalent to finding a critical point, which in turns is equivalent to solving the linear system

$$A^\top Ax = A^\top b - M^\top p^{k+1}. \tag{16}$$

If this system has a solution $\bar{x}$, then we have $\hat{f}_k = \frac{1}{2}\|A\bar{x} - b\|^2 + \langle M^\top p^{k+1}, \bar{x}\rangle > -\infty$. But if $A^\top A$ does not have full rank, as is guaranteed to happen when the number of features n is larger than the number of observations m, then it is possible that (16) has no solution. In this case, the vector $A^\top b - M^\top p^{k+1}$ has a nonzero projection v onto the orthogonal complement of the row space of $A^\top A$, which is the null space of $(A^\top A)^\top = A^\top A$. It is then easily shown that v is a direction of recession of f_k, that is, a direction along which the value of f_k may be driven to $-\infty$. Therefore, f_k is unbounded below, $\hat{f}_k = -\infty$, and the Lagrangian bound $q(p^{k+1})$ must also be $-\infty$. The same would obviously apply to the mid-step Lagrangian bound if a function of the form (15) were to be used in the role of g.

Furthermore, even when (16) has a solution, obtaining one may require significant additional computation beyond that involved in finding x^{k+1} in (2). In fact, solving (16) may be significantly more time consuming than (2), because it involves a possibly indefinite system, whereas (2) involves a positive-definite system to which one may be able to apply the Sherman-Morrison-Woodbury matrix inversion lemma to reduce the dimension of the factorizations needed; see for example [2, Section 4.2.4].

2.1.2 L_1 Penalties

A common choice for g in machine learning applications is an L_1 penalty function of the form $g(z) = \lambda\|z\|_1$. The convex conjugate of this function is

$$\begin{aligned} g^*(p) &= \sup_{z\in\mathbb{R}^r}\{\langle p, z\rangle - g(z)\} = \sup_{z\in\mathbb{R}^r}\{\langle p, z\rangle - \lambda\|z\|_1\} \\ &= \begin{cases} 0, & \text{if } -\lambda \le p_j \le \lambda \;\forall\, j = 1,\ldots,r \\ +\infty, & \text{otherwise.} \end{cases} \end{aligned} \tag{17}$$

Computing $\hat{g}_k = -g^*(p^{k+1/2})$ should therefore not be time consuming, but if the vector $p^{k+1/2}$ has any component whose magnitude exceeds λ, then $\hat{g}_k = -\infty$ and the mid-step Lagrangian bound $q(p^{k+1/2})$ is also $-\infty$. The same would obviously apply to the Lagrangian bound $q(p^{k+1})$ were an L_1 penalty to take the role of f.

2.1.3 Subspace Indicator Functions

Another common choice for g is the convex indicator function of some linear subspace V of $\mathbb{R}^r$, that is

$$g(z) = \begin{cases} 0, & \text{if } z \in V \\ +\infty, & \text{if } z \notin V. \end{cases}$$

Some common examples of this form include the progressive hedging algorithm [11] (in a space with a specially formulated inner product) and consensus problems; see for example

[2, Section 7]. In this case, we have

$$g^*(p) = \sup_{z\in\mathbb{R}^r}\{\langle p, z\rangle - g(z)\} = \sup_{z\in V}\{\langle p, z\rangle\} = \begin{cases} 0, & \text{if } p \in V^\perp \\ +\infty, & \text{if } p \notin V^\perp, \end{cases}$$

which is simply the convex indicator function of the complementary subspace $V^\perp$. Thus, if $p^{k+1/2} \notin V^\perp$, then $\hat{g}_k = -\infty$ and the mid-step Lagrangian bound $q(p^{k+1/2})$ is $-\infty$; the same would obviously apply to the Lagrangian bound $q(p^{k+1})$ were a subpace indicator function to take the role of f. The difficulty of detecting whether $p^{k+1/2} \in V^\perp$ or $p^{k+1} \in V^\perp$ depends on the form of V, but will typically involve numerical tolerance issues.

2.2 Applicability of Lagrangian Bounds

There are situations in which the infimum needed to calculate $\hat{f}_k$ in (11) is guaranteed to be finite and attained. Examples include when the effective domain $\operatorname{dom} f$ on which f takes finite values is compact, or more generally when the image $\operatorname{rge}(\partial f)$ of the subgradient map ∂f is all of $\mathbb{R}^n$ (more specifically, it should be sufficient for $\operatorname{rge}(\partial f)$ to contain the row space of M). In such sitations, the Lagrangian bound calculation in (11)-(12) should always yield a finite bound on the optimal solution. Thus, the Lagrangian bound should be applicable, its main potential drawback being the extra calculation required for (11).

Similarly, if the infimum needed to calculate $\hat{g}_k$ in (13) is guaranteed to be finite and attained, then the calculations in (13)-(14) should always yield a finite bound; this will happen, for example if $\operatorname{rge}(\partial g) = \mathbb{R}^r$. In such cases, the mid-step Lagrangian bound is applicable, its main possible drawback being the extra calculation required for (13), when it is significant.

However, there are cases when neither $\hat{f}_k$ nor $\hat{g}_k$ is guaranteed to be finite, so neither form of Lagrangian bound may be used reliably. A simple example of this kind is the standard LASSO problem [12]

$$\min_{x\in\mathbb{R}^n} \left\{ \tfrac{1}{2} \|Ax - b\|^2 + \lambda \|z\|_1 \right\}, \tag{18}$$

where $\lambda > 0$ is a given scalar, and A and b are respectively an observation matrix and response vector, as in Section 2.1.1. We now propose a different bound calculation that may be used in such situations, provided it is possible to bound the magnitude of some optimal solution vector (as is indeed the case for LASSO, as we shall establish below).

3 A Subgradient-Based Bound Calculation

The alternative bounding procedure relies on the following simple result:

Proposition 1. *Suppose x^* is an optimal solution of (1) with value $F^* \doteq f(x^*) + g(Mx^*)$. Then at every iteration $k \geq 1$ of the ADMM algorithm (2)-(4), one has*

$$F^* \geq f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^{k+1} - z^{k+1}\rangle - c\langle z^k - z^{k+1}, Mx^* - Mx^{k+1}\rangle. \tag{19}$$

Proof. The respective optimality conditions (7) and (9) for (2) and (3) are equivalent to

$$M^\top p^{k+1/2} \in \partial f(x^{k+1}) \qquad\qquad p^{k+1} \in \partial g(z^{k+1}). \tag{20}$$

Therefore, we have

$$f(x^*) \geq f(x^{k+1}) - \langle M^\top p^{k+1/2}, x^* - x^{k+1}\rangle \tag{21}$$
$$g(Mx^*) \geq g(z^{k+1}) + \langle p^{k+1}, Mx^* - z^{k+1}\rangle \tag{22}$$

Rewriting (21) as

$$f(x^*) \geq f(x^{k+1}) - \langle p^{k+1} + c(z^k - z^{k+1}), Mx^* - Mx^{k+1}\rangle$$

and adding it to (22), one obtains

$$\begin{aligned} F^* &\geq f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^* - z^{k+1} - Mx^* + Mx^{k+1}\rangle \\ &\qquad - \langle c(z^k - z^{k+1}), Mx^* - Mx^{k+1}\rangle \\ &= f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^{k+1} - z^{k+1}\rangle - c\langle z^k - z^{k+1}, Mx^* - Mx^{k+1}\rangle. \end{aligned}$$ □

Naturally, (19) does not in general provide a concrete bound on F^* because the optimal solution x^* is necessarily unknown. However, if some bound is known on the magnitude of x^* or Mx^*, then then one may bound the inner product $\langle z^k - z^{k+1}, Mx^* - Mx^{k+1}\rangle$ and then obtain a well behaved bound from (19).

Proposition 2. *Suppose there is an optimal solution x^* of problem (1), along with scalars $B > 0$ and $s \in [1, \infty]$ such that $\|x^*\|_s \leq B$. Then at every iteration $k \geq 1$ of the ADMM algorithm (2)-(4), the optimal value F^* of (1) obeys the inequality*

$$\begin{aligned} F^* \geq f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^{k+1} - z^{k+1}\rangle \\ + c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - cB\left\|M^\top(z^k - z^{k-1})\right\|_{s^*}, \end{aligned} \tag{23}$$

where s^* is such that $1/s + 1/s^* = 1$, with the convention that $1/\infty = 0$. The values

$$\beta_k \doteq f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^{k+1} - z^{k+1}\rangle + c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - cB\left\|M^\top(z^k - z^{k-1})\right\|_{s^*}$$

on the right-hand side of this inequality are finite for all $k \geq 1$ and form a sequence converging to F^*.

Proof. Considering the last term in (19), we have by first applying the Hölder inequality and then the hypothesis $\|x^*\|_s \leq B$ that

$$\begin{aligned} -c\langle z^k - z^{k+1}, Mx^* - Mx^{k+1}\rangle &= c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - c\langle M^\top(z^k - z^{k+1}), x^*\rangle \\ &\geq c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - c\|x^*\|_s \left\|M^\top(z^k - z^{k+1})\right\|_{s^*} \\ &\geq c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - cB\left\|M^\top(z^k - z^{k+1})\right\|_{s^*}. \end{aligned}$$

Substituting this inequality into (19) yields (23). As f and g were assumed closed proper convex, the minimizers in (2) and (3) cannot be at points where f or g have infinite values, so finiteness of the sequence $\{\beta_k\}$ follows immediately.

In this context, the ADMM algorithm (2)-(4) is known, as shown for example in [6] to have the properties that $Mx^k - z^k \to 0$, $f(x^k) + g(z^k) \to F^*$, and $\{p^k\}$ and $\{z^k\}$ are convergent. Denoting the limits of $\{p^k\}$ and $\{z^k\}$ by p^∞ and z^∞ respectively, we may use the properites while taking the limit in the definition of β_k to obtain

$$\beta_k \to F^* + \langle p^\infty, z^\infty - z^\infty\rangle + \langle z^\infty - z^\infty, z^\infty\rangle - cB \left\|M^\top(z^\infty - z^\infty)\right\|_{s^*} = F^*. \qquad \square$$

The first three terms $f(x^{k+1}) + g(z^{k+1}) + \langle p^{k+1}, Mx^{k+1} - z^{k+1}\rangle$ in the formula for β_k would constitute a Lagrangian bound on F^* if (x^{k+1}, z^{k+1}) were a joint minimizer of the Lagrangian with the dual vector p^{k+1}. In fact, although z^{k+1} minimizes the Lagrangian with respect to z with the dual vector p^{k+1}, in the ADMM we instead have that x^{k+1} minimizes the Lagrangian with respect to x at the generally different dual vector $p^{k+1/2}$. The final terms $c\langle z^k - z^{k+1}, Mx^{k+1}\rangle - cB \left\|M^\top(z^k - z^{k-1})\right\|_{s^*}$ in (23) may be considered as a "corrector" bounding the impact of this discrepancy.

By a slight modification of the above analysis, if we instead have a scalar B' such that $\|Mx^*\| \le B'$, the last term of (23) may clearly be replaced by $cB' \left\|z^k - z^{k+1}\right\|_{s^*}$.

It should be noted that an inequality equivalent to (19) appears as (3.11) in [2], and the following unnumbered inequality in [2] partially explores its potential to provide a bound on F^*. However, it does not appear that this kind of technique has been used in practice. Instead, [2] uses this analysis mainly to justify terminating the method when (in the current setting and notation) $\|Mx^k - b\|$ and $\|M^\top(z^k - z^{k-1})\|$ are small. But without the specific calculations embodied in (23), it is this author's experience that such termination criteria can lead to termination at lower accuracy than many users might expect.

4 An Application Example: LASSO

Consider once again the LASSO problem (18), for which neither form of Lagrangian bound may be readily generated by the ADMM. We first establish that it is relatively easy to calculate a bound on the 1-norm of any optimal solution to this problem, meaning that it meets the conditions needed by Proposition 2.

Proposition 3. *Considering the LASSO problem* (18)*, let* $\bar{x} \in \arg\min_{x\in\mathbb{R}^n} \{\, \|Ax - b\|\}$*. Then for any optimal solution* x^* *of* (18)*, one has* $\|x^*\|_1 \le \|\bar{x}\|_1$*.*

Proof. Consider any $\tilde{x} \in \mathbb{R}^n$ with $\|\tilde{x}\|_1 > \|\bar{x}\|_1$. Since $\bar{x}$ minimizes $\|Ax - b\|$, we have $\frac{1}{2}\|A\tilde{x} - b\|^2 \ge \frac{1}{2}\|A\bar{x} - b\|^2$. Multiplying $\|\tilde{x}\|_1 > \|\bar{x}\|_1$ by $\lambda > 0$ and adding inequalities, we obtain

$$\tfrac{1}{2}\|A\tilde{x} - b\|^2 + \lambda\|\tilde{x}\|_1 > \tfrac{1}{2}\|A\bar{x} - b\|^2 + \lambda\|\bar{x}\|_1.$$

Thus, $\tilde{x}$ cannot be optimal for (18), and the result follows. $\square$

Thus, if we compute any minimizer $\bar{x}$ of $\|Ax - b\|$ and set $s = 1$ and $B = \|\bar{x}\|_1$, the conditions needed by Proposition 2 will be satisfied.

We now describe some computational experiments evaluating the bound procedures. These experiments set $\bar{x}$ by using MATLAB's `lsqminnorm` function to find minimizer of $\|Ax - b\|$ having the smallest 2-norm. This is a relatively simple calculation involving only standard matrix operations.

Dataset	m	n	c	Iterations	Time Without LB	Time With LB	Ratio
`brain`	42	5597	5	420	0.37	9.85	26.6
`colon`	62	2000	5	595	0.24	2.43	10.1
`leukemia`	72	3571	5	399	0.36	5.13	14.3
`lymphoma`	62	4026	5	585	0.47	7.81	16.6
`prostate`	102	6033	5	377	0.87	13.24	15.2
`srbct`	63	2308	5	354	0.18	2.09	11.6
`ball64`	1638	4096	10	168	6.74	12.03	1.8
`logo64`	1638	4096	10	167	6.76	12.19	1.8
`mug128`	4770	16384	10	234	144.38	538.90	3.7
`mug32`	410	1024	10	87	0.16	0.38	2.4
`pems`	238	138672	20	2479	314.90	—	—

Table 1: Datasets, iterations, and seconds of run time

The datasets tested are described in Table 1. The first six are cancer microarray datasets from [3], the next four are imaging-related datasets from [4], and the last, `pems`, is a transportation-related datset from UCI repository [10]. All datasets were normalized as in [6], with $\|b\|$ and the norm of each column of A all scaled to 1. The L_1 penalty parameter λ was set to 0.1 in all cases.

When solving the problem instances with the ADMM, M is the $n \times n$ identity matrix, $f(x) = \frac{1}{2}\|Ax - b\|^2$, and $g(z) = \lambda \|z\|_1$. The proximal parameter c was set to approximately optimize convergence for each class of instances and is shown in the fourth column of the table. The algorithm was terminated when the subgradient set $\nabla f(z^k) + \partial g(z^k)$ contained any element with norm less than or equal to $\epsilon \doteq 10^{-4}$, that is,

$$\operatorname{dist}\big(\nabla f(z^k) + \partial g(z^k), 0\big) \leq \epsilon. \tag{24}$$

The implementation was in MATLAB and based on the implementation in [6], and the repective primal and dual starting points z^0 and p^0 were both the zero vector in $\mathbb{R}^n$.

At each iteration, the implementation computed the bound β_k of Proposition 2, which in this case reduces to

$$\beta_k = \tfrac{1}{2}\left\|Ax^{k+1} - b\right\|^2 + \lambda \left\|z^{k+1}\right\|_1 + c\langle z^k - z^{k+1}, x^{k+1}\rangle - c\,\|\bar{x}\|_1 \left\|z^k - z^{k+1}\right\|_\infty.$$

At each iteration, the implementation also computed the mid-step Lagrangian bound $q(p^{k+1/2})$ as in (14). However, this bound can be $-\infty$ since $\hat{g}_k = -g^*(p^{k+1/2})$ can by $+\infty$ as in (17). To make this bound somewhat more practical, the implementation applied a numerical tolerance of $\bar{\epsilon} \doteq 10^{-5}$ to the calculation of $\hat{g}_k$, resulting in

$$\hat{g}_k = \begin{cases} 0, & -(\lambda + \epsilon) \leq z_j^{k+1} \leq \lambda + \epsilon \quad \forall\, j = 1, \ldots, n \\ -\infty, & \text{otherwise.} \end{cases}$$

The implementation had a second, optional mode of operation in which it also attempted to calculate the Lagrangian bound $q(p^{k+1})$ at every iteration. This option can add significantly to run time because it requires the calculation of $\hat{f}_k$ described in Section 2.1.1, involving the attempted solution of the singular system of linear equations (16). To detect the solvability of this system, and produce a solution when possible, the implementation used a simple technique based on the singular value decomposition of A; a numerical tolerance of $\bar{\epsilon} = 10^{-5}$ was also used in assessing the solvability of (16). The last three columns of Table 1 show the number of seconds to meet the termination criterion without the calculation of $q(p^{k+1})$ ("Without LB"), the number of seconds to attain convergence with the calculation of $q(p^{k+1})$ ("With LB"), and the ratio of the two times.

All computations were performed on a Windows PC with a 3.2GHz Core i5-4460 CPU. For the cancer microarray datasets, attempting to compute the Lagrangian bound $q(p^{k+1})$ at every iteration increases run time by at least an order of magnitude; for the imaging datasets, the effect is not as strong, likely because the aspect ratios n/m of these datasets are smaller. For `pems`, it was not practical to compute $q(p^{k+1})$ on the testing system because the singular value decomposition of A would have required 143GB of memory, and so the data in the last two columns are omitted.

For the six cancer microarray datasets, Figure 1 plots the evolution of the objective $f(z^k) + g(z^k)$ and the three bounds as a function o the iteration k. When the Lagrangian bounds are $-\infty$, they are depicted as being zero, since the solution value of the LASSO problem must be nonnegative. Generally, the three bounds evolve similar for the first few iterations, after which β_k converges to the optimal objective value while Lagrangian bounds become $-\infty$ for a large number of iterations. For `brain`, the two Lagrangian bounds meet the expanded numerical tolerance for being finite after about 400 iteration, and then "jump" to the neighborhood of the optimal objective value. However, such a jump never occurs for the other five datasets. The divergence of β_k from the other bounds occurs when $\left\|p^{k+1/2}\right\|_1 > \lambda$

Figure 2 depicts the evolution of the objective function and bounds for the remaining datasets. For the three imaging datasets other than `mug32`, the mid-step Lagrangian bound performs comparatively well, tracking β_k closely until quite near the optimal objective value, after which the threshold $\left\|p^{k+1/2}\right\|_1 > \lambda$ is reached and it becomes $-\infty$. The conventional Lagrangian bound was of no utility for any of the imaging instances, even in the first few iterations. The Lagrangian bound does not appear in the plot for `pems` since it was not practical to compute in this instance.

4.1 Concluding Observations

In summary, the bound β_k appears far more useful and reliable in the context of the ADMM than Lagrangian-style bounds. Its main drawback is the need for some bound on the optimal solution magnitude, as with $B = \|\bar{x}\|_1$ in the case of the LASSO problem. The difficulty of computing such a bound depends both on both the type and size of the problem being solved. For example, for very large-scale LASSO problems finding such a bound could be challenging, even though it was straightforward for the smaller problems tested here.

Inspecting the plots in figures 1 and 2, it appears that the bound β_k and the objective function can become essentially equal long before the subgradient-based termination criterion (24) is met. This phenomenon suggests that the bound could be used in an alternative

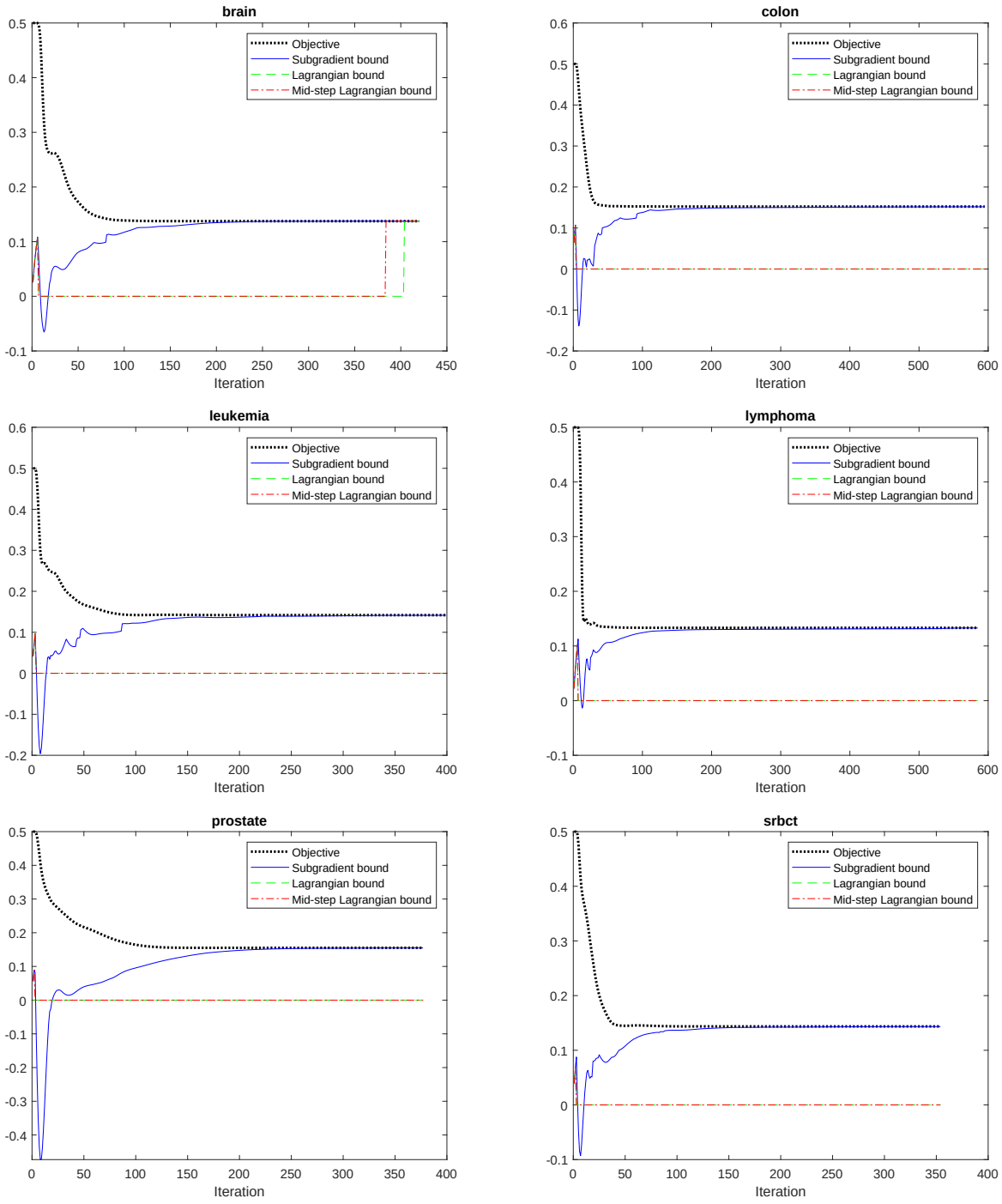

Figure 1: Bound and objective evolution for the cancer microarray datasets

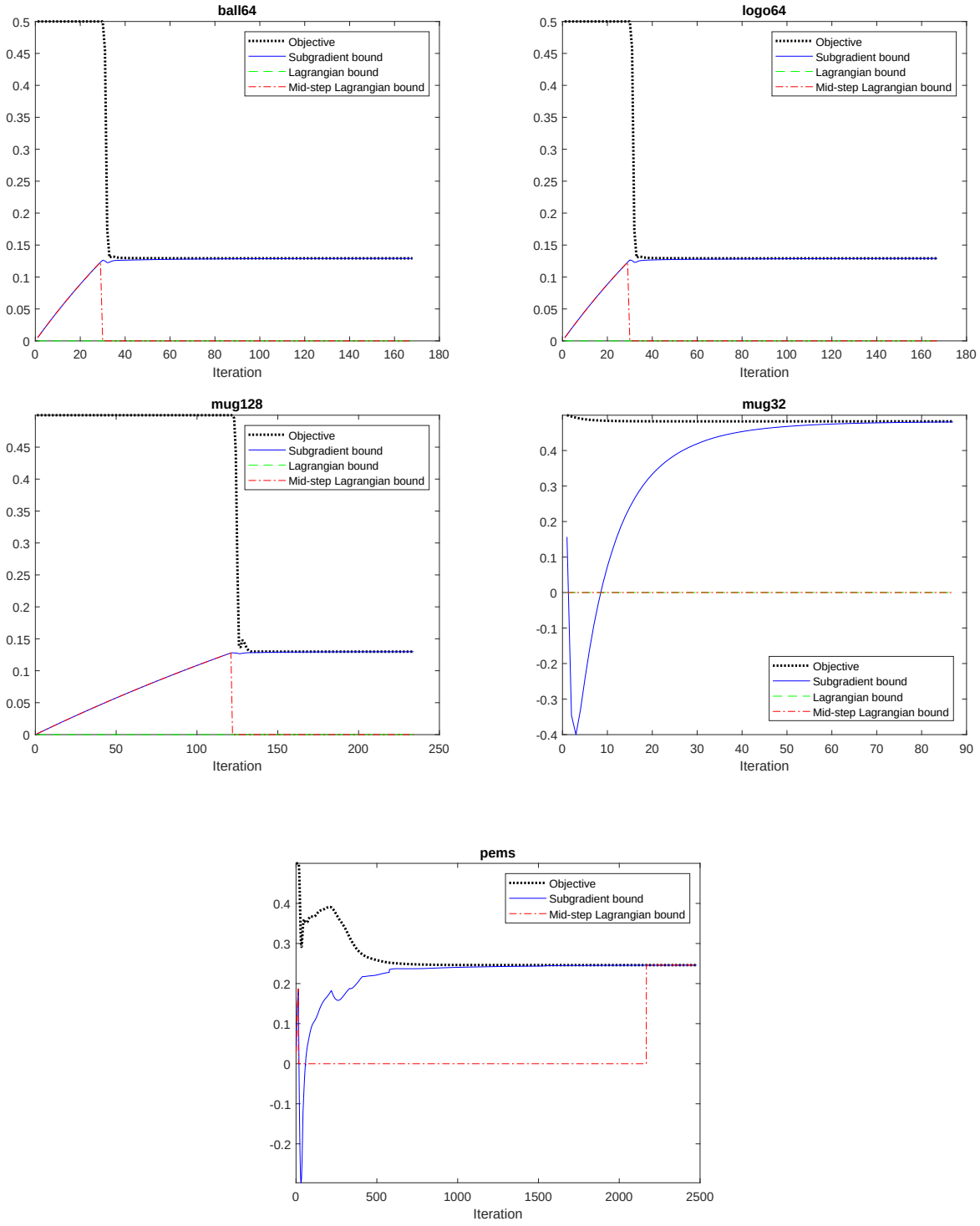

Figure 2: Bound and objective evolution for the other datasets

termination criterion, such as, for some tolerance $\delta > 0$,

$$\begin{aligned} f(x^k) + g(Mx^k) &\leq \beta_k + \delta \qquad \text{for general } M, \text{or} \\ f(z^k) + g(z^k) &\leq \beta_k + \delta \qquad \text{for } M = I. \end{aligned}$$

The experiments also show that the convergence of the bounds β_k to the optimal objective value need not be monotone. Monotonicity could of course be enforced by simply tracking the largest value generated so far, $\bar{\beta}_k \doteq \max_{0 \leq \ell \leq k} \{\beta_\ell\}$.

References

[1] N. Boland, J. Christiansen, B. Dandurand, A. Eberhard, J. Linderoth, J. Luedtke, and F. Oliveira. Combining progressive hedging with a Frank-Wolfe method to compute Lagrangian dual bounds in stochastic mixed-integer programming. *SIAM J. Optim.*, 28(2):1312–1336, 2018.

[2] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1), 2011.

[3] M. Dettling and P. Bühlmann. Finding predictive gene groups from microarray data. *J. Multivariate Anal.*, 90(1):106–131, 2004.

[4] M. F. Duarte, M. A. Davenport, D. Takbar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling: Building simpler, smaller, and less-expensive digital cameras. *IEEE Sig. Proc. Mag.*, 25(2):83–91, 2008.

[5] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Math. Program.*, 55(3):293–318, 1992.

[6] J. Eckstein and W. Yao. Understanding the convergence of the alternating direction method of multipliers: theoretical and computational perspectives. *Pac. J. Optim.*, 11(4):619–644, 2015.

[7] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian methods: Applications to the numerical solution of boundary-value problems*, volume 15 of *Studies in Mathematics and its Applications*, pages 97–146. North-Holland Publishing Co., Amsterdam, 1983.

[8] D. Gabay. Applications of the method of multipliers to variational inequalities. In M. Fortin and R. Glowinski, editors, *Augmented Lagrangian Methods: Applications to the Solution of Boundary Value Problems*, chapter IX, pages 299–340. North-Holland, Amsterdam, 1983.

[9] D. Gade, G. Hackebeil, S. M. Ryan, J.-P. Watson, R. J.-B. Wets, and D. L. Woodruff. Obtaining lower bounds from the progressive hedging algorithm for stochastic mixed-integer programs. *Math. Program.*, 157(1):47–67, 2016.

[10] M. Lichman. UCI machine learning repository, 2013.

[11] R. T. Rockafellar and R. J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Math. Oper. Res.*, 16(1):119–147, 1991.

[12] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.