SIAM J. OPTIM.
Vol. 21, No. 1, pp. 361–390

AN AFFINE-SCALING INTERIOR-POINT METHOD FOR CONTINUOUS KNAPSACK CONSTRAINTS WITH APPLICATION TO SUPPORT VECTOR MACHINES*

MARIA D. GONZALEZ-LIMA[†], WILLIAM W. HAGER[‡], AND HONGCHAO ZHANG[§]

Abstract. An affine-scaling algorithm (ASL) for optimization problems with a single linear equality constraint and box restrictions is developed. The algorithm has the property that each iterate lies in the relative interior of the feasible set. The search direction is obtained by approximating the Hessian of the objective function in Newton's method by a multiple of the identity matrix. The algorithm is particularly well suited for optimization problems where the Hessian of the objective function is a large, dense, and possibly ill-conditioned matrix. Global convergence to a stationary point is established for a nonmonotone line search. When the objective function is strongly convex, ASL converges R-linearly to the global optimum provided the constraint multiplier is unique and a nondegeneracy condition holds. A specific implementation of the algorithm is developed in which the Hessian approximation is given by the cyclic Barzilai-Borwein (CBB) formula. The algorithm is evaluated numerically using support vector machine test problems.

Key words. interior-point, affine-scaling, cyclic Barzilai-Borwein methods, global convergence, linear convergence, support vector machines

AMS subject classifications. 90C06, 90C26, 65Y20

DOI. 10.1137/090766255

1. Introduction.

In this paper we develop an interior-point algorithm for a box-constrained optimization problem with a linear equality constraint:

$$\min\ f(\mathbf{x}) \quad \text{subject to } \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{l} \le \mathbf{x} \le \mathbf{u}, \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = b. \tag{1.1}$$

Here f is a real-valued, continuously differentiable function, $\mathbf{a} \in \mathbb{R}^n$, $\mathbf{a} \neq \mathbf{0}$, $b \in \mathbb{R}^1$, and $\mathbf{l} < \mathbf{u}$ with possibly, $l_i = -\infty$ or $u_i = \infty$. We refer to the constraints in (1.1) as knapsack constraints since they represent a continuous version of the constraints which arise in the discrete knapsack problem. Initially, to simplify the exposition, we focus on the special case $\mathbf{u} = \infty$ and $\mathbf{l} = \mathbf{0}$:

$$\min\ f(\mathbf{x}) \quad \text{subject to } \mathbf{x} \in \mathcal{F}, \tag{1.2}$$

where

$$\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \ge \mathbf{0}, \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = b\}. \tag{1.3}$$

The gradient projection algorithm can be applied to (1.2) since the projection on the feasible set (1.3) can be evaluated quickly. A general framework for a nonmonotone spectral gradient projection algorithm is developed by Birgin, Martínez,

*Received by the editors July 27, 2009; accepted for publication (in revised form) February 2, 2011; published electronically March 17, 2011. This material is based upon work supported by the National Science Foundation under grants 0619080, 0620286, and 1016204, by the Office of Naval Research under grant N00014-11-1-0068, and by the Universidad Simón Bolívar.

http://www.siam.org/journals/siopt/21-1/76625.html

[†] Departamento de Cómputo Científico y Estadística, Universidad Simón Bolívar, Apdo. 89000, Caracas 1080-A, Venezuela (mgl@cesma.usb.ve).

[‡] Department of Mathematics, University of Florida, PO Box 118105, Gainesville, FL 32611-8105 (hager@math.ufl.edu, http://www.math.ufl.edu/∼hager).

[§] Department of Mathematics, Center for Computation and Technology, Louisiana State University, 140 Lockett Hall, Baton Rouge, LA 70803-4918 (hozhang@math.lsu.edu, http://www.math.lsu.edu/∼hozhang).

and Raydan in [4, 5, 6, 7], while Dai and Fletcher [17] develop a version tailored to a knapsack constraint such as (1.3). In this paper, we develop a new affine-scaling algorithm, denoted ASL, tailored to the knapsack constraint (1.3), and compare it to the algorithm of Dai and Fletcher. Our algorithm is an extension of an earlier affine-scaling algorithm [34] for box-constrained optimization. The algorithm starts at a point $\mathbf{x}_1$ in the relative interior of the feasible set $\mathcal{F}$ and generates a sequence $\mathbf{x}_k$, $k \geq 2$ by the following rule:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + s_k \mathbf{d}_k, \tag{1.4}$$

where $s_k \in (0, 1]$ is a positive stepsize, and the ith component of $\mathbf{d}_k$ is given by

$$d_{ki} = -\left(\frac{1}{\lambda_k + \nabla_x L_i(\mathbf{x}_k, \mu_k)^+ / x_{ki}} \right) \nabla_x L_i(\mathbf{x}_k, \mu_k). \tag{1.5}$$

Here $L : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is the Lagrangian defined by

$$L(\mathbf{x}, \mu) = f(\mathbf{x}) + \mu(b - \mathbf{a}^{\mathsf{T}}\mathbf{x}),$$

and $\nabla_x L_i$ is the ith component of the gradient of L with respect to variable $\mathbf{x}$. In (1.5), λ_k is a positive scalar, $t^+ = \max\{0, t\}$ for any scalar t, and the parameter μ_k is chosen so that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$. We give numerical results for a specific implementation of ASL in which λ_k is computed using a cyclic version of the Barzilai-Borwein (CBB) stepsize rule [3, 18]. In the CBB method, the same BB step is reused in several iterations. Since the BB method does not monotonically reduce the value of the cost function, a nonmonotone line search is needed to ensure that the stepsize in the line search is 1 when the iterates converge [18, 19, 54].

We now motivate the search direction $\mathbf{d}_k$ in (1.5). The first-order optimality conditions (KKT conditions) for (1.2) can be expressed as

$$\mathbf{X}^1(\mathbf{x}, \mu) \circ \nabla_x L(\mathbf{x}, \mu) = \mathbf{0}, \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = b, \quad \mathbf{x} \geq \mathbf{0}, \tag{1.6}$$

where

$$X_i^1(\mathbf{x}, \mu) = \begin{cases} 1 & \text{if} \quad \nabla_x L_i(\mathbf{x}, \mu) \leq 0, \\ x_i & \text{otherwise.} \end{cases} \tag{1.7}$$

Here "$\circ$" denotes the Hadamard (or componentwise) product of two vectors. That is, if $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, then $\mathbf{x} \circ \mathbf{y} \in \mathbb{R}^n$ and $(\mathbf{x} \circ \mathbf{y})_i = x_i y_i$, where x_i is the ith component of $\mathbf{x}$. The parameter μ is the Lagrange multiplier associated with the linear equality constraint. For a convex optimization problem, the KKT conditions (1.6) are necessary and sufficient for optimality. Our algorithm amounts to an iterative method for solving (1.6).

The ASL iterates are chosen to lie in the relative interior of the feasible set $\mathcal{F}$ in (1.3). Let $\mathbf{x}_k$ denote the current iterate, and let us substitute $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k$ in (1.6) and linearize around $\mathbf{x}_k$ to obtain

$$\mathbf{H}(\mathbf{x}_k, \mu)\mathbf{d}_k = -\mathbf{X}^1(\mathbf{x}_k, \mu) \circ \nabla L(\mathbf{x}_k, \mu), \quad \mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0, \tag{1.8}$$

where

$$\mathbf{H}(\mathbf{x}_k, \mu) = \mathbf{diag}(\mathbf{X}^1(\mathbf{x}_k, \mu))\, \nabla^2 f(\mathbf{x}_k) + \mathbf{diag}(\nabla_x L(\mathbf{x}_k, \mu)^+). \tag{1.9}$$

Here $\mathbf{diag}(\mathbf{x})$ is an n by n diagonal matrix with ith diagonal element x_i. In situations where $\nabla^2 f(\mathbf{x})$ is a huge, dense matrix, it can be time consuming to solve the linear system (1.8). In the ASL algorithm, we make the approximation $\nabla^2 f(\mathbf{x}_k) \approx \lambda_k \mathbf{I}$, where $\lambda_k > 0$. With this approximation, the matrix in (1.9) is diagonal. For each choice of μ, there is an associated solution $\mathbf{d}_k(\mu)$ to the first equation in (1.8) with $\nabla^2 f(\mathbf{x}_k)$ replaced by $\lambda_k \mathbf{I}$. In the ASL algorithm, we choose μ such that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k(\mu) = 0$; in other words, μ is chosen so that the solution to the first equation in (1.8) satisfies the second equation in (1.8). As we will see, there is a unique choice for μ with the property that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k(\mu) = 0$. In the ASL iteration (1.4), an additional stepsize parameter s_k is introduced to ensure global convergence; moreover, for each $s_k \in (0, 1]$, if $\mathbf{x}_k$ lies in the relative interior of the feasible set $\mathcal{F}$, then so does $\mathbf{x}_{k+1} = \mathbf{x}_k + s_k \mathbf{d}_k$.

ASL generalizes the affine-scaling algorithm in [34] by allowing for an additional linear equality constraint $\mathbf{a}^{\mathsf{T}}\mathbf{x} = b$. There are many important applications that involve both bound constraints and an additional linear equality constraint. Examples include the maximum clique problem [29, 48], the graph partitioning problem [33], and the support vector machine (SVM) [10, 15, 64]. In this paper, we will compare the performance of ASL, when applied to SVM test problems, to that of the SVM codes LIBSVM [14, 23] and GPDT [60, 61, 67, 68]. SVM has been used in many real life applications including pattern recognition and classification problems such as isolated handwritten digit recognition [11, 12, 15, 58, 59], object recognition [8], speaker identification [57], face detection [50, 51], text categorization [38], and nonlinear least squares problems arising in inverse density estimation [65].

Our paper is organized as follows: In section 2 an implementation of the ASL algorithm is presented; the line search is based on the nonmonotone scheme of Grippo, Lampariello, and Lucidi (GLL) [32]. Note, though, that the numerical experiments employ the nonmonotone line search in [34] for which the analysis is more complex, but which often yields better performance in practice. In section 3 various continuity properties of the ASL algorithm are developed. Section 4 establishes global convergence to a stationary point, while section 5 gives a global linear convergence result for strongly convex functions. Section 6 discusses the BB choice for the parameter λ_k. The generalization of ASL to box constraints is given in section 7, while section 8 gives a Newton–Secant scheme for computing the parameter μ_k in (1.5). Some of the most effective algorithms for SVM problems have been block coordinate descent methods in which each iteration involves an optimization over a working set. In section 9 we give a procedure for selecting a working set, and we compare it to a scheme of Joachims [39]. Finally, in section 10 we evaluate the performance of a version of ASL, denoted svmASL, which is specialized to SVM problems.

Notation. For any scalar t, $t^+ = \max\{0, t\}$, while for any vector $\mathbf{v} \in \mathbb{R}^n$, $\mathbf{v}^+$ is the vector whose ith component is v_i^+. $\nabla f(\mathbf{x})$ denotes the gradient of f, a row vector. The gradient of $f(\mathbf{x})$, arranged as a column vector, is $\mathbf{g}(\mathbf{x})$. The subscript k often represents the iteration number in an algorithm, and $\mathbf{g}_k$ stands for $\mathbf{g}(\mathbf{x}_k)$. If $\mathbf{x}^*$ is an optimal solution of (1.2), then $\mathbf{g}^*$ denotes $\mathbf{g}(\mathbf{x}^*)$. We let x_{ki} denote the ith component of the iterate $\mathbf{x}_k$. The Hadamard (or componentwise) product $\mathbf{x} \circ \mathbf{y}$ of two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ is the vector in $\mathbb{R}^n$ defined by $(\mathbf{x} \circ \mathbf{y})_i = x_i y_i$. $\mathbf{diag}(\mathbf{x})$ is an n by n diagonal matrix with ith diagonal element x_i. $\|\cdot\|$ is the Euclidean norm and $|\mathcal{S}|$ is the number of elements in the set $\mathcal{S}$.

2. The ASL algorithm with GLL linear search.

The general framework for the ASL algorithm (1.4)–(1.5), with the nonmonotone GLL line search [32], is the following:

Affine-Scaling Interior-Point Method for Knapsack Constraints (ASL)

```
Given $\lambda_0 > 0$, $\delta$ and $\eta \in (0,1)$, and an integer $M \geq 0$.
Choose $\mathbf{x}_1 > 0$ with $\mathbf{a}^{\mathsf{T}}\mathbf{x}_1 = b$ and set $k = 1$.
  Step 1.   Choose $\lambda_k \geq \lambda_0$ and find $\mu_k$ such that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$ where $\mathbf{d}_k$
            is defined in (1.5).
  Step 2.   If $\mathbf{d}_k = \mathbf{0}$, stop.
  Step 3.   Choose $s_k = \eta^j$ with $j \geq 0$ the smallest integer such that
            $f(\mathbf{x}_k + s_k\mathbf{d}_k) \leq f_k^R + \delta s_k \nabla f(\mathbf{x}_k)\mathbf{d}_k$,
            where $f_k^R = \max\{f(\mathbf{x}_{k-j}) : 0 \leq j \leq \min(k-1, M)\}$.
  Step 4.   Set $\mathbf{x}_{k+1} = \mathbf{x}_k + s_k\mathbf{d}_k$.
  Step 5.   Set $k = k+1$ and go to step 1.
```

The parameter f_k^R is the reference function value. For the ordinary Armijo line search [1], $f_k^R = f(\mathbf{x}_k)$. For the GLL line search, f_k^R is the local maximum function value defined in step 3. The choice for f_k^R used in the numerical experiments appears in [34].

We first show that if the ASL iterate $\mathbf{x}_k$ lies in the relative interior of the feasible set, then so does $\mathbf{x}_{k+1}$.

Proposition 2.1. *If $\mathbf{x}_k$ lies in the relative interior of the feasible set $\mathcal{F}$ and $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$, then $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{d}_k$ lies in the relative interior of $\mathcal{F}$ for all $\alpha_k \in [0,1]$.*

Proof. In order to show that $\mathbf{x}_{k+1} > 0$ if $\mathbf{x}_k > 0$, it suffices to prove that if $d_{ki} < 0$, then $d_{ki} > -x_{ki}$, which implies that $x_{ki} + \alpha_k d_{ki} > 0$ because $\alpha_k \in [0,1]$. By definition of d_{ki} in (1.5) and the requirement that $\lambda_k > 0$, we have

$$d_{ki} = \begin{cases} -\dfrac{g_i(\mathbf{x}_k) - \mu_k a_i}{\lambda_k} \geq 0 & \text{if} \quad g_i(\mathbf{x}_k) - \mu_k a_i \leq 0, \\ -\dfrac{g_i(\mathbf{x}_k) - \mu_k a_i}{\lambda_k + (g_i(\mathbf{x}_k) - \mu_k a_i)/x_{ki}} > -x_{ki} & \text{if} \quad g_i(\mathbf{x}_k) - \mu_k a_i > 0. \end{cases}$$

Hence, $\mathbf{x}_{k+1} > 0$. Furthermore, $\mathbf{a}^{\mathsf{T}}\mathbf{x}_{k+1} = \mathbf{a}^{\mathsf{T}}(\mathbf{x}_k + \alpha_k\mathbf{d}_k) = \mathbf{a}^{\mathsf{T}}\mathbf{x}_k = b$ since $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$. □

We now show that there exists a unique μ_k such that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$. Given $\mathbf{x} > \mathbf{0}$ and $\lambda > 0$, let us introduce the functions

$$d_i(\mu) = -\frac{g_i - \mu a_i}{\lambda + (g_i - \mu a_i)^+/x_i}, \tag{2.1}$$

$$r_i(\mu) = a_i d_i(\mu), \quad \text{and} \quad r(\mu) = \sum_{i=1}^{n} r_i(\mu) = \mathbf{a}^{\mathsf{T}}\mathbf{d}(\mu). \tag{2.2}$$

Finding μ_k such that $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$ amounts to finding a zero of $r(\cdot) = \mathbf{a}^{\mathsf{T}}\mathbf{d}(\cdot)$ in the case $\lambda = \lambda_k$, $\mathbf{x} = \mathbf{x}_k$, and $\mathbf{g} = \mathbf{g}(\mathbf{x}_k)$.

Proposition 2.2. *Suppose that $\mathbf{x} > 0$ and $\lambda > 0$, and define*

$$\mu_0 = \min\{g_i/a_i : 1 \leq i \leq n, \quad a_i \neq 0\},$$
$$\mu_1 = \max\{g_i/a_i : 1 \leq i \leq n, \quad a_i \neq 0\}.$$

The function $r(\mu)$ has a unique zero on the interval $[\mu_0, \mu_1]$, $r(\mu) < 0$ for all $\mu < \mu_0$, and $r(\mu) > 0$ for all $\mu > \mu_1$. Moreover, r is continuously differentiable and monotone with $r'(\mu) > 0$ for every μ.

Proof. Since

$$r_i(\mu) = \begin{cases} -\dfrac{x_i a_i(g_i - \mu a_i)}{x_i\lambda + (g_i - \mu a_i)} & \text{if} \quad g_i - \mu a_i > 0, \\ -\dfrac{a_i(g_i - \mu a_i)}{\lambda} & \text{if} \quad g_i - \mu a_i \leq 0, \end{cases}$$

it is clear that $r_i(\mu)$ is continuously differentiable when $\mu a_i \neq g_i$. In the case that $\mu a_i = g_i$ and $a_i \geq 0$, we have

$$\begin{aligned}\lim_{h\to 0^+} \frac{r_i(\mu+h) - r_i(\mu)}{h} &= \lim_{h\to 0^+} \frac{a_i(ha_i)}{\lambda h} = \frac{a_i^2}{\lambda},\\ \lim_{h\to 0^-} \frac{r_i(\mu+h) - r_i(\mu)}{h} &= \lim_{h\to 0^-} \frac{x_i h a_i^2}{h(x_i\lambda - ha_i)} = \frac{a_i^2}{\lambda}.\end{aligned}$$

If $a_i < 0$, the same result holds. Therefore, $r_i(\mu)$ is differentiable for all μ and its derivative is continuous with

$$r_i'(\mu) = \begin{cases} \dfrac{\lambda a_i^2 x_i^2}{(x_i\lambda + g_i - \mu a_i)^2} & > 0 \quad \text{if} \quad g_i - \mu a_i > 0,\\ \dfrac{a_i^2}{\lambda} & > 0 \quad \text{if} \quad g_i - \mu a_i \leq 0.\end{cases}$$

If $a_i \neq 0$, then $r_i'(\mu) > 0$ for all μ, and $r_i(\cdot)$ is strictly increasing. Since $r_i(g_i/a_i) = 0$, it follows that $r_i(\mu) < 0$ for $\mu < g_i/a_i$ and $r_i(\mu) > 0$ for $\mu > g_i/a_i$. Consequently, $r_i(\mu) < 0$ for $\mu < \mu_0$ and $r_i(\mu) > 0$ for $\mu > \mu_1$. In the case $a_i = 0$, $r_i(\cdot) = 0$. Since $\mathbf{a} \neq \mathbf{0}$ by assumption, it follows from the definition of r that $r(\mu) < 0$ for $\mu < \mu_0$, and $r(\mu) > 0$ for $\mu > \mu_1$. □

Proposition 2.2 implies that the ASL algorithm is well defined. Next, we show that the search direction (1.5) associated with the ASL algorithm is a descent direction whenever $\|\mathbf{d}(\mu)\| \neq 0$. This implies that the Armijo line search in step 3 of ASL terminates at a finite j.

Proposition 2.3. *If* $\mathbf{x} > 0$, $\lambda > 0$, *and* μ *is chosen such that* $\mathbf{a}^{\mathsf{T}}\mathbf{d}(\mu) = 0$, *then we have*

$$\mathbf{g}^{\mathsf{T}}\mathbf{d}(\mu) \leq -\lambda\|\mathbf{d}(\mu)\|^2,$$

where $\mathbf{d}$ *is defined in* (2.1). *Therefore,* $\mathbf{g}^{\mathsf{T}}\mathbf{d}(\mu) < 0$ *whenever* $\mathbf{d}(\mu) \neq \mathbf{0}$.

Proof. Since $\mathbf{a}^{\mathsf{T}}\mathbf{d}(\mu) = 0$, we have

$$\begin{aligned}\mathbf{g}^{\mathsf{T}}\mathbf{d}(\mu) = (\mathbf{g} - \mu\mathbf{a})^{\mathsf{T}}\mathbf{d}(\mu) &= \sum_{i=1}^n (g_i - \mu a_i) d_i(\mu)\\ &= -\sum_{i=1}^n (\lambda + (g_i - \mu a_i)^+/x_i) d_i(\mu)^2 \qquad (2.3)\\ &\leq -\lambda\|\mathbf{d}(\mu)\|^2.\end{aligned}$$

The equality (2.3) follows from (2.1). □

3. Continuity properties.

In this section, some continuity properties of ASL are established. The proofs of Propositions 3.1 and 3.2 and Lemma 3.3 are analogous to proofs given in [34]. They are included for completeness.

Proposition 3.1. *If* f *is continuously differentiable and* $\mathbf{X}^1(\cdot)$ *is defined by* (1.7), *then the map*

$$\mathbf{X}^1(\cdot,\cdot) \circ \nabla_x L(\cdot,\cdot) : \mathbb{R}^n_+ \times \mathbb{R} \to \mathbb{R}^n, \quad \mathbb{R}^n_+ := \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0}\},$$

is continuous.

Proof. Since ∇f is continuous, $\nabla_x L_i(\cdot,\cdot)$ is continuous. If either $\nabla_x L_i(\mathbf{x},\mu) < 0$ or $\nabla L(\mathbf{x},\mu)_i > 0$, then $X_i^1(\cdot,\cdot)$ is continuous at $\mathbf{x}$; consequently, the product

$\nabla_x L_i(\cdot,\cdot)X_i^1(\cdot,\cdot)$ is continuous at $(\mathbf{x},\mu)$. Suppose that $\nabla_x L_i(\mathbf{x},\mu) = 0$. By the definition of $\mathbf{X}^1$, we have

$$|\nabla_x L_i(\mathbf{y},\nu)X_i^1(\mathbf{y},\nu)| \le \max\{1, y_i\}|\nabla_x L_i(\mathbf{y},\nu)|$$

for any $\mathbf{y} \ge \mathbf{0}$ and $i \in [1,n]$. Since $\nabla_x L_i(\cdot,\cdot)$ is continuous and $\nabla_x L_i(\mathbf{x},\mu) = 0$, it follows that $\nabla_x L_i(\mathbf{y},\nu)X_i^1(\mathbf{y},\nu)$ approaches zero as $(\mathbf{y},\nu)$ approaches $(\mathbf{x},\mu)$. Therefore, the product $\nabla_x L(\cdot,\cdot) \circ \mathbf{X}^1(\cdot,\cdot)$ is continuous everywhere. □

Now we show that the ASL search directions approach zero if and only if the KKT conditions are satisfied in an asymptotic sense.

Proposition 3.2. *Suppose f is continuously differentiable on the feasible set $\mathcal{F}$, and the parameter λ_k in* (1.5) *is bounded from above and below by finite positive constants:*

$$0 < \lambda_0 := \inf_{k\ge 1} \lambda_k \le \sup_{k\ge 1} \lambda_k := \lambda_{\max} < \infty. \tag{3.1}$$

If $\{(\mathbf{x}_k,\mu_k) : 1 \le k < \infty\}$ is uniformly bounded and $\mathbf{x}_k$ is in the relative interior of $\mathcal{F}$ for each k, then

$$\lim_{k\to\infty} \mathbf{d}_k = \mathbf{0} \quad \text{if and only if} \quad \lim_{k\to\infty} \mathbf{X}^1(\mathbf{x}_k,\mu_k) \circ \nabla_x L(\mathbf{x}_k,\mu_k) = \mathbf{0},$$

where $\mathbf{d}_k$ and $\mathbf{X}^1$ are defined in (1.5) *and* (1.7)*, respectively.*

Proof. We will show that, for any $i \in [1,n]$,

$$\lim_{k\to\infty} d_{ki} = 0 \quad \text{if and only if} \quad \lim_{k\to\infty} X_i^1(\mathbf{x}_k,\mu_k)\nabla_x L_i(\mathbf{x}_k,\mu_k) = 0, \tag{3.2}$$

in which case the proposition follows immediately. By (1.5), we have

$$d_{ki} = -\frac{x_{ki}\nabla_x L_i(\mathbf{x}_k,\mu_k)}{\lambda_k x_{ki} + \nabla_x L_i(\mathbf{x}_k,\mu_k)^+}. \tag{3.3}$$

Since the $\mathbf{x}_k$ lie in the relative interior of $\mathcal{F}$ and the λ_k are positive by (3.1), the denominator of (3.3) is positive for each k. Since $\mathbf{x}_k$, μ_k, and λ_k are bounded and f is continuously differentiable, the denominator on the right-hand side of (3.3) is bounded. Consequently, if d_{ki} tends to zero, the numerator $x_{ki}\nabla_x L_i(\mathbf{x}_k,\mu_k)$ tends to zero. If $\nabla_x L_i(\mathbf{x}_k,\mu_k) > 0$, then $\mathbf{X}_i^1(\mathbf{x}_k,\mu_k) = x_{ki}$; hence, along the subsequence of iterates where $\nabla_x L_i(\mathbf{x}_k,\mu_k) > 0$, $\mathbf{X}_i^1(\mathbf{x}_k,\mu_k)\nabla_x L_i(\mathbf{x}_k,\mu_k) = x_{ki}\nabla_x L_i(\mathbf{x}_k,\mu_k)$, which tends to zero. Along the subsequence of iterates where $\nabla_x L_i(\mathbf{x}_k,\mu_k) \le 0$, we have $d_{ki} = -\nabla_x L_i(\mathbf{x}_k,\mu_k)/\lambda_k$. Hence, if d_{ki} tends to zero, then $\nabla_x L_i(\mathbf{x}_k,\mu_k)$ tends to zero, which implies that $X_i^1(\mathbf{x}_k,\mu_k)\nabla_x L_i(\mathbf{x}_k,\mu_k)$ tends to zero since $\mathbf{X}_i^1(\mathbf{x}_k,\mu_k)$ is bounded.

Conversely, suppose that $X_i^1(\mathbf{x}_k,\mu_k)\nabla_x L_i(\mathbf{x}_k,\mu_k)$ tends to zero. In this case, we can write

$$\{1,2,\ldots\} = \mathcal{K}_1 \cup \mathcal{K}_2,$$

where either $\mathcal{K}_1$ or $\mathcal{K}_2$ may be empty and

$$\text{(a)} \ \lim_{k\in\mathcal{K}_1} \nabla L_i(\mathbf{x}_k,\mu_k) = 0 \quad \text{and} \quad \text{(b)} \ \lim_{k\in\mathcal{K}_2} X_i^1(\mathbf{x}_k,\mu_k) = 0.$$

If $\mathcal{K}_1$ has an infinite number of elements, then (3.1) and (3.3) imply that d_{ki} tends to zero for $k \in \mathcal{K}_1$ approaching ∞. If $\mathcal{K}_2$ has an infinite number of elements, then for $k \in \mathcal{K}_2$ with k sufficiently large, we have $X_i^1(\mathbf{x}_k, \mu_k) = x_{ki}$ and $\nabla_x L_i(\mathbf{x}_k, \mu_k)^+ = \nabla_x L_i(\mathbf{x}_k, \mu_k) > 0$. Consequently, (3.3) can be rewritten

$$d_{ki} = -\frac{x_{ki}}{1 + \lambda_k x_{ki}/\nabla_x L_i(\mathbf{x}_k, \mu_k)}, \quad k \in \mathcal{K}_2. \tag{3.4}$$

By (b) $X_i^1(\mathbf{x}_k, \mu_k) = x_{ki}$ tends to zero as $k \in \mathcal{K}_2$ tends to ∞. By (3.4), d_{ki} tends to zero as $k \in \mathcal{K}_2$ tends to ∞. Hence, the entire sequence $\{d_{ki} : k \ge 1\}$ approaches zero, which completes the proof of (3.2). □

The next lemma shows that if $\mathbf{x}^*$ is a KKT point with corresponding multiplier μ^*, then $\mathbf{d}_k$ approaches $\mathbf{0}$ as $(\mathbf{x}_k, \mu_k)$ approaches $(\mathbf{x}^*, \mu^*)$.

LEMMA 3.3. *Suppose f is Lipschitz continuously differentiable in a neighborhood of a KKT point $\mathbf{x}^*$ for* (1.2)*. If μ^* is the multiplier associated with $\mathbf{x}^*$, then there exists a constant c such that, for all $(\mathbf{x}_k, \mu_k)$ near $(\mathbf{x}^*, \mu^*)$ with $\mathbf{x}_k > 0$, and for all $\lambda_k \ge \lambda_0 > 0$, we have*

$$\|\mathbf{d}_k\| \le c(\|\mathbf{x}^* - \mathbf{x}_k\| + |\mu^* - \mu_k|). \tag{3.5}$$

Proof. Since $\mathbf{x}^*$ is a KKT point for (1.2) with corresponding multiplier μ^*, we have $\mathbf{X}^1(\mathbf{x}^*, \mu^*) \circ \nabla_x L(\mathbf{x}^*, \mu^*) = \mathbf{0}$. Hence, for each i, either

(a) $g_i(\mathbf{x}^*) - \mu^* a_i = 0$, or (b) $g_i(\mathbf{x}^*) - \mu^* a_i > 0$ and $X_i^1(\mathbf{x}^*, \mu^*) = x_i^* = 0$.

From the definition of $\mathbf{d}_k$, it follows that, for any $\mathbf{x}_k > \mathbf{0}$, we have

$$|d_{ki}| \le |g_i(\mathbf{x}_k) - \mu_k a_i|/\lambda_0.$$

Hence, for those indices i where (a) holds,

$$\begin{aligned} |d_{ki}| &\le (|g_i(\mathbf{x}_k) - g_i(\mathbf{x}^*)| + |(\mu_k - \mu^*)a_i|)/\lambda_0 \\ &\le \left(\frac{\kappa + |a_i|}{\lambda_0}\right)(\|\mathbf{x}^* - \mathbf{x}_k\| + |\mu^* - \mu_k|), \end{aligned} \tag{3.6}$$

where $\kappa > 0$ is the Lipschitz constant of ∇f. For any index i where (b) holds and for $(\mathbf{x}_k, \mu_k)$ in a neighborhood of $(\mathbf{x}^*, \mu^*)$ with $\mathbf{x}_k > \mathbf{0}$, we have $(g_i(\mathbf{x}_k) - \mu_k a_i)^+ = g_i(\mathbf{x}_k) - \mu_k a_i > 0$ and

$$|d_{ki}| \le x_{ki} \le \|\mathbf{x}_k - \mathbf{x}^*\|. \tag{3.7}$$

Combining (3.6) and (3.7) gives (3.5) for a suitable choice of c. □

We now show that the stepsize s_k is bounded away from zero.

PROPOSITION 3.4. *If s_k is chosen in accordance with step 3 of ASL, and ∇f is Lipschitz continuous with Lipschitz constant κ on the line segment connecting $\mathbf{x}_k$ and $\mathbf{x}_k + \mathbf{d}_k$, then we have*

$$s_k \ge \min\left\{1, \frac{2\eta(1-\delta)\lambda_k}{\kappa}\right\}. \tag{3.8}$$

Proof. If step 3 of ASL terminates with $j = 0$, then $s_k = 1$, which satisfies (3.8). On the other hand, if step 3 terminates with $j > 0$, the we must have that

$$f(\mathbf{x}_k + \eta^{j-1}\mathbf{d}_k) \ge f_k^R + \delta\eta^{j-1}\mathbf{g}_k^\mathsf{T}\mathbf{d}_k \ge f(\mathbf{x}_k) + \delta\eta^{j-1}\mathbf{g}_k^\mathsf{T}\mathbf{d}_k. \tag{3.9}$$

By the fundamental theorem of calculus,

$$f(\mathbf{x}_k + \eta^{j-1}\mathbf{d}_k) - f(\mathbf{x}_k) = \eta^{j-1}\nabla f(\mathbf{x}_k)\mathbf{d}_k + \int_0^{\eta^{j-1}} (\nabla f(\mathbf{x}_k + t\mathbf{d}_k) - \nabla f(\mathbf{x}_k))\mathbf{d}_k dt.$$

By the Lipschitz continuity of ∇f,

$$f(\mathbf{x}_k + \eta^{j-1}\mathbf{d}_k) - f(\mathbf{x}_k) \le \eta^{j-1}\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k + 0.5\kappa\eta^{j-1}\|\mathbf{d}_k\|^2. \tag{3.10}$$

Combining (3.9) and (3.10) gives

$$(\delta - 1)\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le 0.5\kappa\eta^{j-1}\|\mathbf{d}_k\|^2. \tag{3.11}$$

Multiplying by η, we obtain

$$s_k = \eta^j \ge 2\eta(1-\delta)|\mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k|/(\kappa\|\mathbf{d}_k\|^2).$$

Utilizing Proposition 2.3, the proof is complete. □

4. Global convergence. We now prove the global convergence of ASL. According to step 3 of the ASL algorithm,

$$f(\mathbf{x}_{k+1}) \le f_k^R \tag{4.1}$$

for each k. Hence, we have

$$f_{k+1}^R \le f_k^R \le \cdots \le f_1^R = f(\mathbf{x}_1). \tag{4.2}$$

We assume that the following level set $\mathcal{L}$ is bounded:

$$\mathcal{L} = \{\mathbf{x} \in \mathcal{F} : f(\mathbf{x}) \le f(\mathbf{x}_1)\}. \tag{4.3}$$

Combining (4.1) and (4.2), it follows that $f(\mathbf{x}_k) \le f(\mathbf{x}_1)$ and $\mathbf{x}_k \in \mathcal{L}$ for each k. Hence, the iterates $\mathbf{x}_k$ lie in a bounded set. Since f is continuously differentiable, the gradients $\mathbf{g}_k$ are bounded. By Proposition 2.2, μ_k is bounded uniformly in k. These uniform bounds for $\|\mathbf{x}_k\|$ and μ_k together with the lower bound $\lambda_k \ge \lambda_0 > 0$ in (3.1) imply that $\|\mathbf{d}_k\|$ is uniformly bounded by some finite constant β.

THEOREM 4.1. *If $\lambda_k \ge \lambda_0 > 0$ for all k, the level set $\mathcal{L}$ is bounded, and f is Lipschitz continuously differentiable on the set*

$$\overline{\mathcal{L}} := \{\mathbf{x} \in \mathcal{F} : \|\mathbf{x} - \mathbf{y}\| \le \beta \textit{ for some } \mathbf{y} \in \mathcal{L}\}, \tag{4.4}$$

then ASL either terminates in a finite number of iterations at a KKT point, or

$$\lim_{k\to\infty} \mathbf{d}_k = \mathbf{0}. \tag{4.5}$$

Proof. If algorithm ASL terminates at iteration k, then $\mathbf{d}_k = \mathbf{0}$ and $\nabla_x L(\mathbf{x}_k, \mu_k) = \mathbf{0}$, which implies that the KKT conditions hold at $\mathbf{x}_k$. In the case that $\mathbf{d}_k \ne \mathbf{0}$ for all k, we show that $\mathbf{d}_k$ approaches $\mathbf{0}$. By Proposition 3.4 and the lower bound $\lambda_k \ge \lambda_0 > 0$, there exists a constant C such that $s_k \ge C$ for all k. By step 3 of ASL and Proposition 2.3, we have

$$f(\mathbf{x}_{k+1}) \le f_k^R + \delta s_k \mathbf{g}_k^{\mathsf{T}}\mathbf{d}_k \le f_k^R - \delta C\lambda_0\|\mathbf{d}_k\|^2. \tag{4.6}$$

By (4.2) and the fact that f is bounded from below on $\mathcal{L}$, we conclude that f_k^R monotonically approaches a limit denoted f_∞^R. We now show that

$$\lim_{k\to\infty} f(\mathbf{x}_k) = f_\infty^R. \tag{4.7}$$

Since $f_k^R = \max\{f(\mathbf{x}_{k-j}) : 0 \le j \le \min(k-1, M)\}$, it follows that, for each k, there exists an index l such that

$$f_k^R = f(\mathbf{x}_l), \quad \text{where } k - M \le l \le k. \tag{4.8}$$

Since l depends on k, we let $l(k)$ denote the index associated with k as in (4.8). Since $f_k^R = f(\mathbf{x}_{l(k)})$, it follows that

$$\lim_{k\to\infty} f(\mathbf{x}_{l(k)}) = \lim_{k\to\infty} f_k^R = f_\infty^R. \tag{4.9}$$

We prove by induction that, for all $j \ge 0$,

$$\lim_{k\to\infty} f(\mathbf{x}_{l(k)-j}) = f_\infty^R. \tag{4.10}$$

This holds for $j = 0$ by (4.9). Assume that (4.10) holds for each j between zero and i; we will show that (4.10) holds for $j = i+1$. By (4.6) with $k = l(k) - i - 1$, we have

$$f(\mathbf{x}_{l(k)-i}) \le f_{l(k)-i-1}^R - \delta C \lambda_0 \|\mathbf{d}_{l(k)-i-1}\|^2.$$

By the induction hypothesis, $f(\mathbf{x}_{l(k)-i})$ approaches f_∞^R as k tends to ∞. Since f_k^R also approaches f_∞^R, we conclude that $\mathbf{d}_{l(k)-i-1}$ tends to $\mathbf{0}$. Since $s_k \le 1$, it follows that

$$\lim_{k\to\infty} \|\mathbf{x}_{l(k)-i} - \mathbf{x}_{l(k)-i-1}\| = 0.$$

The Lipschitz continuity of f over $\overline{\mathcal{L}}$ implies that

$$\lim_{k\to\infty} |f(\mathbf{x}_{l(k)-i}) - f(\mathbf{x}_{l(k)-i-1})| = 0.$$

Hence, by (4.10) for $j = i$, we have

$$f_\infty^R = \lim_{k\to\infty} f(\mathbf{x}_{l(k)-i}) = \lim_{k\to\infty} f(\mathbf{x}_{l(k)-i-1}).$$

Consequently, (4.10) holds for $j = i+1$. This completes the induction and (4.10) holds for all $j \ge 0$.

For $j = 0, 1, \ldots, M$, define the sets

$$I_j = \cup_{k\ge1}\{l(k) - j\}.$$

The identity (4.10) is equivalent to

$$\lim_{\substack{k \,\in\, I_j \\ k \,\to\, \infty}} f(\mathbf{x}_k) = f_\infty^R.$$

Since $k - l(k) \le M$, it follows that each k lies in one of the sets I_j, $0 \le j \le M$. For the indices k in any of these sets I_j, the function values $f(\mathbf{x}_k)$ approach f_∞^R. This establishes (4.7), and by (4.6) it follows that $\mathbf{d}_k$ approaches $\mathbf{0}$ as k tends to ∞. $\square$

Remark. Together, Theorem 4.1 and Proposition 3.2 imply that, if the parameter λ_k in ASL is uniformly bounded from above, then the KKT conditions are satisfied in an asymptotic sense. In particular, section 6 shows that the BB choice for λ_k is uniformly bounded.

5. Global and local convergence for strongly convex functions. In this section, we develop a global linear convergence result in the case that f is strongly convex over the feasible set $\mathcal{F}$. Recall that f is strongly convex if there exists a constant $\gamma > 0$ such that

$$f(\mathbf{y}) \geq f(\mathbf{x}) + \nabla f(\mathbf{x})(\mathbf{y} - \mathbf{x}) + \gamma \|\mathbf{x} - \mathbf{y}\|^2 \tag{5.1}$$

for all $\mathbf{x}$ and $\mathbf{y} \in \mathcal{F}$. We first show that, if f also satisfies the hypotheses of Theorem 4.1, then the iterates $\mathbf{x}_k$ converge to the unique solution of (1.2).

THEOREM 5.1. *Suppose f is strongly convex over the feasible set $\mathcal{F}$, and the parameter λ_k is uniformly bounded away from zero and ∞ in accordance with* (3.1). *If f is Lipschitz continuously differentiable over the set* (4.4)*, then the iterates $\mathbf{x}_k$ generated by ASL either converge in a finite number of iterations to the unique solution $\mathbf{x}^*$ of* (1.2)*, or*

$$\lim_{k \to \infty} \mathbf{x}_k = \mathbf{x}^*. \tag{5.2}$$

Proof. If ASL converges in finite number of iterations to a point $\mathbf{x}^*$, then by Theorem 4.1, the KKT conditions hold at $\mathbf{x}^*$. Due to the strong convexity of f and the convexity of the feasible set $\mathcal{F}$, it follows that $\mathbf{x}^*$ is the unique solution of (1.2). Conversely, suppose that ASL does not terminate in a finite number of iterations. Since f is strongly convex, the level set $\mathcal{L}$ is bounded; that is,

$$\|\mathbf{x} - \mathbf{x}_1\| \leq \|\nabla f(\mathbf{x}_1)\|/\gamma$$

for all $\mathbf{x} \in \mathcal{L}$. By (4.1) and (4.2), each of the ASL iterates $\mathbf{x}_k$ lies in $\mathcal{L}$. Therefore, the gradients $\mathbf{g}_k$ are uniformly bounded, which together with Proposition 2.2 ensures μ_k is also uniformly bounded. Hence, the sequence $\{(\mathbf{x}_k, \mu_k)\}$ is uniformly bounded, and there exists a subsequence $\{(\mathbf{x}_{k_i}, \mu_{k_i})\}$ which converges to a pair $(\mathbf{x}^*, \mu^*) \in \mathcal{F} \times \mathbb{R}$. By Theorem 4.1, $\mathbf{d}_{k_i}$ approaches $\mathbf{0}$ as i tends to ∞. By Proposition 3.2,

$$\lim_{i \to \infty} X^1(\mathbf{x}_{k_i}, \mu_{k_i}) \circ \nabla_x L(\mathbf{x}_{k_i}, \mu_{k_i}) = \mathbf{0}.$$

By Proposition 3.1, we have

$$X^1(\mathbf{x}^*, \mu^*) \circ \nabla_x L(\mathbf{x}^*, \mu^*) = \mathbf{0}.$$

Hence $\mathbf{x}^*$ is a KKT point for (1.2) with corresponding multiplier μ^*. Again, since f is strongly convex over the convex set $\mathcal{F}$, $\mathbf{x}^*$ is the unique solution of (1.2), which achieves the global minimum. During the proof of Theorem 4.1, we show that the entire sequence $f(\mathbf{x}_k)$ approaches a limit. Since the subsequence $f(\mathbf{x}_{k_i})$ approaches the global minimum $f(\mathbf{x}^*)$, it follows that the entire sequence $f(\mathbf{x}_k)$ approaches the global minimum. By the first-order optimality condition

$$\nabla f(\mathbf{x}^*)(\mathbf{x} - \mathbf{x}^*) \geq 0$$

for all $\mathbf{x} \in \mathcal{F}$, and by the strong convexity condition (5.1), we have

$$\gamma \|\mathbf{x}_k - \mathbf{x}^*\|^2 \leq f(\mathbf{x}_k) - f(\mathbf{x}^*). \tag{5.3}$$

Hence, $\mathbf{x}_k$ approaches $\mathbf{x}^*$ as $f(\mathbf{x}_k)$ approaches $f(\mathbf{x}^*)$. This completes the proof. □

Next, we show that $f(\mathbf{x}_k)$ converges to $f(\mathbf{x}^*)$ at least R-linearly when the following multiplier uniqueness and nondegeneracy assumptions hold. As a result, by (5.3), the iterates $\mathbf{x}_k$ converge linearly to $\mathbf{x}^*$:

Multiplier uniqueness: There exists an index i such that $a_i \neq 0$ and $x_i^* > 0$, where $\mathbf{x}^*$ is the global minimizer of (1.2).

Let μ^* be a multiplier associated with $\mathbf{x}^*$ and the linear equality constraint. The KKT conditions imply that $g_i(\mathbf{x}^*) - \mu^* a_i = 0$ when $x_i^* > 0$. Hence, $\mu^* = g_i(\mathbf{x}^*)/a_i$ when the multiplier uniqueness condition holds. In other words, the multiplier is uniquely determined.

Nondegeneracy: the strict complementarity condition holds at $\mathbf{x}^*$; that is,

$$g_i(\mathbf{x}^*) - \mu^* a_i > 0 \text{ whenever } x_i^* = 0.$$

The following lemma shows that μ_k converges to μ^* at least as fast as $\mathbf{x}_k$ approaches $\mathbf{x}^*$. Moreover, the search direction $\mathbf{d}_k$ provides a local error bound.

LEMMA 5.2. *Suppose the multiplier uniqueness and nondegeneracy conditions hold, f is strongly convex over the feasible set $\mathcal{F}$, and f is Lipschitz continuously differentiable over the convex hull of the level set* (4.3)*. If λ_k is bounded away from zero and ∞ in accordance with* (3.1)*, then there exist positive constants τ_1, τ_2, ζ_1, and ζ_2 such that*

$$|\mu_k - \mu^*| \le \tau_1 \|\mathbf{x}_k - \mathbf{x}^*\|, \tag{5.4}$$

$$\|\mathbf{x}_k - \mathbf{x}^*\| \le \tau_2 \|\mathbf{d}_k\|, \text{ and} \tag{5.5}$$

$$-\zeta_1 \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \le \mathbf{g}_k^{\mathsf{T}} (\mathbf{x}_k - \mathbf{x}^*) \le -\zeta_2 \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \tag{5.6}$$

for all k sufficiently large.

Proof. We first show that the parameters μ_k converge to the unique Lagrange multiplier μ^* associated with $\mathbf{x}^*$ and the linear constraint $\mathbf{a}^{\mathsf{T}}\mathbf{x} = b$. By Theorem 5.1, $\mathbf{x}_k$ converges to $\mathbf{x}^*$. By Proposition 2.2, the μ_k are uniformly bounded. Suppose that a subsequence of the $\{\mu_k\}$ approaches a limit ν. As seen in the proof of Theorem 5.1, $\mathbf{x}^*$ is a KKT point for (1.2) with corresponding multiplier ν. By the multiplier uniqueness condition, $\nu = \mu^*$. Hence, the entire sequence μ_k converges to μ^*.

Since $\mathbf{x}_k \to \mathbf{x}^*$ and $\mu_k \to \mu^*$ as $k \to \infty$, the first-order optimality conditions along with the nondegeneracy condition imply that

$$\lim_{k\to\infty} (g_{ki} - \mu_k a_i) = \begin{cases} g_i^* - \mu^* a_i = 0 & \text{if} \quad i \notin \mathcal{A}, \\ g_i^* - \mu^* a_i > 0 & \text{if} \quad i \in \mathcal{A}, \end{cases} \tag{5.7}$$

where $g_i^* = g_i(\mathbf{x}^*)$ and

$$\mathcal{A} = \{i \in [1, n] : x_i^* = 0\}.$$

We rearrange the identity $\mathbf{a}^{\mathsf{T}}\mathbf{d}_k = 0$ and take k large enough that $g_{ki} - \mu_k a_i > 0$ for all $i \in \mathcal{A}$ to obtain

$$\begin{aligned} \left| \sum_{i \notin \mathcal{A}} \frac{a_i (g_{ki} - \mu_k a_i)}{\lambda_k + (g_{ki} - \mu_k a_i)^+ / x_{ki}} \right| &= \left| \sum_{i \in \mathcal{A}} \frac{a_i x_{ki}}{(\lambda_k / (g_{ki} - \mu_k a_i)) x_{ki} + 1} \right| \\ &\le \sum_{i \in \mathcal{A}} |a_i x_{ki}| \le \|\mathbf{a}\| \|\mathbf{x}_k - \mathbf{x}^*\|. \end{aligned} \tag{5.8}$$

In the last inequality, we utilize the Schwarz inequality and fact that $x_i^* = 0$ for $i \in \mathcal{A}$. Since $g_i^* - \mu^* a_i = 0$ when $i \notin \mathcal{A}$, we have

$$\sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - \mu_k a_i)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} = \sum_{i \notin \mathcal{A}} \frac{a_i((g_{ki} - \mu_k a_i) - (g_i^* - \mu^* a_i))}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} = \sum_{i \notin \mathcal{A}} \frac{a_i^2(\mu^* - \mu_k)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} + \sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - g_i^*)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}}. \tag{5.9}$$

For $i \notin \mathcal{A}$, $g_{ki} - \mu_k a_i$ approaches zero and x_{ki} is bounded away from zero. Since λ_k is bounded away from zero, it follows that

$$\frac{(g_{ki} - \mu_k a_i)^+}{x_{ki}} \le \lambda_k \text{ for all } i \notin \mathcal{A} \text{ and } k \text{ sufficiently large,} \tag{5.10}$$

which implies that

$$\sum_{i \notin \mathcal{A}} \frac{a_i^2}{2\lambda_{\max}} \le \sum_{i \notin \mathcal{A}} \frac{a_i^2}{2\lambda_k} \le \sum_{i \notin \mathcal{A}} \frac{a_i^2}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}}, \tag{5.11}$$

where $\lambda_{\max}$ is the upper bound for λ_k. We rearrange (5.9) and combine with (5.11) to obtain

$$\begin{aligned} |\mu_k - \mu^*| \sum_{i \notin \mathcal{A}} \frac{a_i^2}{2\lambda_{\max}} &\le \sum_{i \notin \mathcal{A}} \frac{a_i^2 |\mu_k - \mu^*|}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} \\ &\le \left| \sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - g_i^*)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} \right| + \left| \sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - \mu_k a_i)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} \right| \\ &\le \sum_{i \notin \mathcal{A}} \frac{|a_i(g_{ki} - g_i^*)|}{\lambda_0} + \left| \sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - \mu_k a_i)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} \right| \\ &\le \frac{\kappa \|\mathbf{a}\| \|\mathbf{x}_k - \mathbf{x}^*\|}{\lambda_0} + \left| \sum_{i \notin \mathcal{A}} \frac{a_i(g_{ki} - \mu_k a_i)}{\lambda_k + (g_{ki} - \mu_k a_i)^+/x_{ki}} \right| \\ &\le \left(1 + \frac{\kappa}{\lambda_0}\right) \|\mathbf{a}\| \|\mathbf{x}_k - \mathbf{x}^*\|. \end{aligned}$$

Here, κ is the Lipschitz constant for $\mathbf{g}$ and the last inequality is from (5.8). This establishes (5.4).

Next, we establish the error bound condition $\|\mathbf{x}_k - \mathbf{x}^*\| \le \tau_2 \|\mathbf{d}_k\|$. The strong convexity assumption (5.1) implies that

$$\|\mathbf{x}_k - \mathbf{x}^*\|^2 \le (\mathbf{g}_k - \mathbf{g}^*)^\mathsf{T}(\mathbf{x}_k - \mathbf{x}^*)/(2\gamma). \tag{5.12}$$

By the first-order optimality conditions, $g_i^* - \mu^* a_i = 0$ for $i \notin \mathcal{A}$. Since $\mathbf{x}_k$ is feasible in (1.2), we have $\mathbf{a}^\mathsf{T}(\mathbf{x}_k - \mathbf{x}^*) = 0$. Consequently, we have

$$\begin{aligned} (\mathbf{g}_k - \mathbf{g}^*)^\mathsf{T}(\mathbf{x}_k - \mathbf{x}^*) &= ((\mathbf{g}_k - \mu_k \mathbf{a}) - (\mathbf{g}^* - \mu^* \mathbf{a}))^\mathsf{T}(\mathbf{x}_k - \mathbf{x}^*) \\ &= \sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)(x_{ki} - x_{ki}^*) \\ &\quad + \sum_{i \in \mathcal{A}} [(g_{ki} - g_{ki}^*) + a_i(\mu^* - \mu_k)] x_{ki}. \end{aligned} \tag{5.13}$$

The Schwarz inequality yields

$$\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)(x_{ki}-x_{ki}^*) \le \|\mathbf{x}_k-\mathbf{x}^*\|\sqrt{\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2}. \tag{5.14}$$

By the Lipschitz continuity of $\mathbf{g}$ and by the estimate (5.4) for $\mu_k-\mu^*$, it follows that

$$\sum_{i\in\mathcal{A}}[(g_{ki}-g_{ki}^*)+a_i(\mu^*-\mu_k)]x_{ki} \le (\kappa+\tau_1\|\mathbf{a}\|)\|\mathbf{x}_k-\mathbf{x}^*\|\sqrt{\sum_{i\in\mathcal{A}}x_{ki}^2}. \tag{5.15}$$

Combining (5.12)–(5.15) gives

$$\|\mathbf{x}_k-\mathbf{x}^*\| \le \frac{1}{2\gamma}\left(\sqrt{\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2}+(\kappa+\tau_1\|\mathbf{a}\|)\sqrt{\sum_{i\in\mathcal{A}}x_{ki}^2}\right). \tag{5.16}$$

We now give an upper bound for the right side of (5.16) in terms of $\|\mathbf{d}_k\|$.

Choose k large enough that

$$g_{ki}-\mu_k a_i > 0 \quad \text{and} \quad \frac{x_{ki}\lambda_k}{g_{ki}-\mu_k a_i} \le 1 \text{ for all } i\in\mathcal{A}. \tag{5.17}$$

By the nondegeneracy condition and the fact that $g_{ki}-\mu_k a_i$ approaches $g_i^*-\mu^* a_i>0$ for all $i\in\mathcal{A}$, it is always possible to choose k in this way. By the definition of $\mathbf{d}$, it follows from (5.17) that

$$d_{ki} \ge x_{ki}/2 \text{ for all } i\in\mathcal{A} \tag{5.18}$$

for k large enough. For $i\notin\mathcal{A}$, choose k large enough that (5.10) holds. Hence, we have

$$d_{ki} \ge |g_{ki}-\mu_k a_i|/(2\lambda_k) \ge |g_{ki}-\mu_k a_i|/(2\lambda_{\max}) \text{ for all } i\notin\mathcal{A}. \tag{5.19}$$

Combining (5.16), (5.18), and (5.19) gives (5.5).

Next, we establish the upper bound in (5.6). Utilizing the identity $\mathbf{a}^{\mathsf{T}}(\mathbf{x}_k-\mathbf{x}^*)=0$ gives

$$\begin{aligned}\mathbf{g}_k^{\mathsf{T}}(\mathbf{x}_k-\mathbf{x}^*) &= (\mathbf{g}_k-\mu_k\mathbf{a})^{\mathsf{T}}(\mathbf{x}_k-\mathbf{x}^*)\\ &= \sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)(x_{ki}-x_{ki}^*)+\sum_{i\in\mathcal{A}}(g_{ki}-\mu_k a_i)(x_{ki}-x_{ki}^*)\\ &\le \sqrt{\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2}\|\mathbf{x}_k-\mathbf{x}^*\|+\sum_{i\in\mathcal{A}}(g_{ki}-\mu_k a_i)x_{ki}. \end{aligned} \tag{5.20}$$

By (5.16) we have

$$\sqrt{\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2}\|\mathbf{x}_k-\mathbf{x}^*\| \le$$

$$c\left(\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2+\sqrt{\sum_{i\notin\mathcal{A}}(g_{ki}-\mu_k a_i)^2}\sqrt{\sum_{i\in\mathcal{A}}x_{ki}^2}\right) \tag{5.21}$$

for a suitable choice of the constant c. Since $g_{ki} - \mu_k a_i$ approaches zero for all $i \in \mathcal{A}$, it follows from the nondegeneracy assumption that

$$\sqrt{\sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2} \le \min_{i \in \mathcal{A}} \{g_{ki} - \mu_k a_i\} \tag{5.22}$$

for k large enough. Since the 2-norm is bounded by the 1-norm, it follows from (5.22) that

$$\sqrt{\sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2} \sqrt{\sum_{i \in \mathcal{A}} x_{ki}^2} \le \min_{i \in \mathcal{A}} \{g_{ki} - \mu_k a_i\} \sum_{i \in \mathcal{A}} x_{ki} \tag{5.23}$$

for k sufficiently large. Hence, we have

$$\min_{i \in \mathcal{A}} \{g_{ki} - \mu_k a_i\} \sum_{i \in \mathcal{A}} x_{ki} + \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki} \le 2 \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki}. \tag{5.24}$$

Combining (5.20)–(5.24) gives

$$\mathbf{g}_k^{\mathsf{T}} (\mathbf{x}_k - \mathbf{x}^*) \le c \left(\sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 + \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki} \right) \tag{5.25}$$

for a suitable choice of c and for k sufficiently large. Also, choose k large enough that (5.10) and (5.17) hold. Since $\mathbf{a}^{\mathsf{T}} \mathbf{d}_k = 0$, it follows from the definition of $\mathbf{d}_k$ that

$$\begin{aligned} -\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k &= -(\mathbf{g}_k - \mu_k \mathbf{a})^{\mathsf{T}} \mathbf{d}_k \\ &= \sum_{i \notin \mathcal{A}} \frac{(g_{ki} - \mu_k a_i)^2}{\lambda_k + (g_{ki} - \mu_k a_i)^+ / x_{ki}} + \sum_{i \in \mathcal{A}} \frac{(g_{ki} - \mu_k a_i) x_{ki}}{(x_{ki} \lambda_k / (g_{ki} - \mu_k a_i)) + 1} && (5.26) \\ &\ge \frac{1}{2\lambda_k} \sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 + \frac{1}{2} \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki} \\ &\ge \frac{1}{2\lambda_{\max}} \sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 + \frac{1}{2} \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki}. && (5.27) \end{aligned}$$

Combining (5.25) and (5.27) yields the upper bound in (5.6).

Finally, we focus on the lower bound in (5.6). The convexity inequality (5.12) and the first half of (5.13) can be rearranged as

$$(\mathbf{g}_k - \mu_k \mathbf{a})^{\mathsf{T}} (\mathbf{x}_k - \mathbf{x}^*) \ge 2\gamma \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \sum_{i \in \mathcal{A}} (g_i^* - \mu^* a_i) x_{ki}.$$

Since $\mathbf{a}^{\mathsf{T}} (\mathbf{x}_k - \mathbf{x}^*) = 0$, it follows that

$$\mathbf{g}_k^{\mathsf{T}} (\mathbf{x}_k - \mathbf{x}^*) \ge 2\gamma \|\mathbf{x}_k - \mathbf{x}^*\|^2 + \sum_{i \in \mathcal{A}} (g_i^* - \mu^* a_i) x_{ki}. \tag{5.28}$$

Again, take k large enough that (5.17) holds. If the denominators in (5.26) are replaced by the respective lower bounds λ_k and 1, we obtain

$$\begin{aligned} -\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k = -(\mathbf{g}_k - \mu_k \mathbf{a})^{\mathsf{T}} \mathbf{d}_k &\le \frac{1}{\lambda_k} \sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 + \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki} \\ &\le \frac{1}{\lambda_0} \sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 + \sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki}. && (5.29) \end{aligned}$$

Also, choose k large enough that

$$g_{ki} - \mu_k a_i \le 2(g_i^* - \mu^* a_i) \text{ for all } i \in \mathcal{A}.$$

Hence, we have

$$\sum_{i \in \mathcal{A}} (g_{ki} - \mu_k a_i) x_{ki} \le 2 \sum_{i \in \mathcal{A}} (g_i^* - \mu^* a_i) x_{ki}. \tag{5.30}$$

Since $g_i^* - \mu^* a_i = 0$ for $i \notin \mathcal{A}$, we obtain

$$\begin{aligned}\sum_{i \notin \mathcal{A}} (g_{ki} - \mu_k a_i)^2 &= \sum_{i \notin \mathcal{A}} ((g_{ki} - \mu_k a_i) - (g_i^* - \mu^* a_i))^2 \\ &\le (\|\mathbf{g}_k - \mathbf{g}^*\| + |\mu_k - \mu^*| \|\mathbf{a}\|)^2 \\ &\le (\kappa + \tau_1 \|\mathbf{a}\|)^2 \|\mathbf{x}_k - \mathbf{x}^*\|^2. \qquad (5.31)\end{aligned}$$

Combining (5.28)–(5.31), the lower bound in (5.6) is established. □

We now show that the function value $f(\mathbf{x}_k)$ generated by ASL converge R-linearly when f is strongly convex.

Theorem 5.3. *Suppose the multiplier uniqueness and nondegeneracy conditions hold, f is strongly convex over the feasible set $\mathcal{F}$, and f is Lipschitz continuously differentiable over the convex hull of the level set* (4.3) *and over the set* (4.4). *If λ_k is bounded away from zero and ∞ in accordance with* (3.1), *then either ASL converges in a finite number of iterations, or there exists $\theta \in (0,1)$ and an integer $K > 0$ such that*

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \theta^k (f(\mathbf{x}_1) - f(\mathbf{x}^*)) \tag{5.32}$$

for all $k > K$.

Proof. Suppose that ASL does not converge in finite number of iterations, otherwise the proof is complete. Choose $K \ge M + 1$ large enough that (5.4)–(5.6) hold for all $k \ge K$. Recall that the ASL iterates are given by $\mathbf{x}_{k+1} = \mathbf{x}_k + s_k \mathbf{d_k}$ with $s_k \in (0,1]$. Let κ denote the Lipschitz constant for $\mathbf{g}$. Observe that

$$\begin{aligned}&|\mathbf{g}_{k+1}^{\mathsf{T}}(\mathbf{x}_{k+1} - \mathbf{x}^*) - \mathbf{g}_k^{\mathsf{T}}(\mathbf{x}_k - \mathbf{x}^*)| \\ &\quad \le |(\mathbf{g}_{k+1} - \mathbf{g}_k)^{\mathsf{T}}(\mathbf{x}_{k+1} - \mathbf{x}^*)| + |\mathbf{g}_k^{\mathsf{T}}(\mathbf{x}_{k+1} - \mathbf{x}_k)| \\ &\quad \le \kappa \|\mathbf{x}_{k+1} - \mathbf{x}_k\| \|\mathbf{x}_{k+1} - \mathbf{x}^*\| + |s_k| |\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k| \\ &\quad \le \kappa \|\mathbf{x}_{k+1} - \mathbf{x}_k\| (\|\mathbf{x}_{k+1} - \mathbf{x}_k\| + \|\mathbf{x}_k - \mathbf{x}^*\|) + |s_k| |\mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k| \\ &\quad \le \kappa (s_k^2 + \tau_2) \|\mathbf{d}_k\|^2 - s_k \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \qquad (5.33) \\ &\quad \le -\frac{\kappa(1 + \tau_2)}{\lambda_0} \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k - \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k \qquad (5.34) \\ &\quad = -\left(\frac{\kappa(1 + \tau_2) + \lambda_0}{\lambda_0} \right) \mathbf{g}_k^{\mathsf{T}} \mathbf{d}_k. \qquad (5.35)\end{aligned}$$

Here (5.33) is due to (5.5) and the relation $\mathbf{x}_{k+1} - \mathbf{x}_k = s_k \mathbf{d}_k$; (5.34) is a consequence of Proposition 2.3 and the condition $s_k \in (0,1]$. By (5.6) and (5.35), we have, for $k \ge K$,

$$
\begin{aligned}
-\mathbf{g}_{k+1}^{\mathsf T}\mathbf{d}_{k+1} &\le \mathbf{g}_{k+1}^{\mathsf T}(\mathbf{x}_{k+1}-\mathbf{x}^*)/\zeta_1\\
&\le \frac{1}{\zeta_1}\left(\mathbf{g}_k^{\mathsf T}(\mathbf{x}_k-\mathbf{x}^*)+|\mathbf{g}_{k+1}^{\mathsf T}(\mathbf{x}_{k+1}-\mathbf{x}^*)-\mathbf{g}_k^{\mathsf T}(\mathbf{x}_k-\mathbf{x}^*)|\right)\\
&\le \frac{1}{\zeta_1}\left(\mathbf{g}_k^{\mathsf T}(\mathbf{x}_k-\mathbf{x}^*)-\frac{\kappa(1+\tau_2)+\lambda_0}{\lambda_0}\mathbf{g}_k^{\mathsf T}\mathbf{d}_k\right)\\
&\le -\left(\frac{\kappa(1+\tau_2)+\lambda_0(1+\zeta_2)}{\zeta_1\lambda_0}\right)\mathbf{g}_k^{\mathsf T}\mathbf{d}_k := -\tau\mathbf{g}_k^{\mathsf T}\mathbf{d}_k. \qquad (5.36)
\end{aligned}
$$

In Theorem 4.1, in the proof of (4.6), it was shown that there exists a constant $\rho > 0$ such that

$$f(\mathbf{x}_{k+1}) \le f_k^R + \rho\mathbf{g}_k^{\mathsf T}\mathbf{d}_k \qquad (5.37)$$

for each k. In step 3 of the ASL algorithm, we set $f_k^R = \max\{f(\mathbf{x}_{k-j}) : 0 \le j \le \min(k-1, M)\}$. Hence, for any $k > M$, there exists l such that $k-1-M \le l \le k-1$ and

$$f(\mathbf{x}_k) \le f(\mathbf{x}_l) + \rho\mathbf{g}_{k-1}^{\mathsf T}\mathbf{d}_{k-1}. \qquad (5.38)$$

Note that $k-l \le M+1$. Combining (5.36) and (5.38) gives

$$f(\mathbf{x}_k) \le f(\mathbf{x}_l) + \left(\frac{\rho}{\tau}\right)\mathbf{g}_k^{\mathsf T}\mathbf{d}_k. \qquad (5.39)$$

On the other hand, by the convexity of f and by (5.6), we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \mathbf{g}(\mathbf{x}_k)^{\mathsf T}(\mathbf{x}_k-\mathbf{x}^*) \le -\zeta_2\mathbf{g}_k^{\mathsf T}\mathbf{d}_k \text{ for all } k \ge K. \qquad (5.40)$$

Subtracting $f(\mathbf{x}^*)$ from both sides of (5.39) and using (5.40) to bound the $\mathbf{g}_k^{\mathsf T}\mathbf{d}_k$ term, we have

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le f(\mathbf{x}_l) - f(\mathbf{x}^*) - \frac{\rho}{\tau\zeta_2}(f(\mathbf{x}_k) - f(\mathbf{x}^*)).$$

Rearranging this gives

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \psi(f(\mathbf{x}_l) - f(\mathbf{x}^*)), \quad \psi = \frac{\tau\zeta_2}{\tau\zeta_2+\rho} < 1.$$

We apply this decay formula in a recursive fashion. The recursion is stopped when reaching an index l for which $l < K$. When this occurs, we utilize the trivial estimate

$$f(\mathbf{x}_l) - f(\mathbf{x}^*) \le f(\mathbf{x}_1) - f(\mathbf{x}^*)$$

based on (4.1) and (4.2). Since $k-l \le M+1$, there are at least $\lceil (k-K)/(M+1)\rceil$ steps in the recursion:

$$
\begin{aligned}
f(\mathbf{x}_k) - f(\mathbf{x}^*) &\le \psi^{(k-K)/(M+1)}(f(\mathbf{x}_1) - f(\mathbf{x}^*))\\
&= (\psi^{(1-K/k)/(M+1)})^k(f(\mathbf{x}_1) - f(\mathbf{x}^*)).
\end{aligned}
$$

For $k > 2K$, we have $1 - K/k \ge 1/2$, which implies

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \le \theta^k(f(\mathbf{x}_1) - f(\mathbf{x}^*)),$$

where $\theta = \psi^{1/(2M+2)} < 1$ for $k > 2K$. This completes the proof. □

6. The BB choice of $\boldsymbol{\lambda_k}$. Our choice of λ_k in the numerical experiments is based on the CBB method where the same Hessian estimate is reused for multiple iterations. The BB method is due to Barzilai and Borwein [3]. It was further developed by Raydan [54]. The first cyclic idea, which includes the BB method as a special case, was presented by Friedlander et al. in [27] for unconstrained quadratic optimization. Dai [16] and Raydan and Svaiter [55] presented cyclic versions of the BB algorithm with cycle length two. Later, Dai et al. [18] introduced the CBB method where the cycle length is arbitrary and proved local linear convergence at a local minimizer of a quadratic with a positive definite Hessian.

The BB method is a quasi-Newton method, where the Hessian $\nabla^2 f(\mathbf{x}_k)$ is replaced by $\lambda_k I$ at each iteration $k \geq 2$. The parameter λ_k is the solution of the least squares problem

$$\min_{\lambda \in \mathbb{R}} \|\lambda \mathbf{s}_{k-1} - \mathbf{y}_{k-1}\|.$$

Here, $\mathbf{s}_{k-1} = \mathbf{x}_k - \mathbf{x}_{k-1}$, $\mathbf{y}_{k-1} = \mathbf{g}_k - \mathbf{g}_{k-1}$, and $\mathbf{g}_k = \mathbf{g}(\mathbf{x}_k)$. In [34] a lower bound $\lambda_0 > 0$ for λ is introduced to ensure global convergence of the affine-scaling algorithm. The modified least squares problem was

$$\lambda_k^{BB} := \arg\min_{\lambda \geq \lambda_0} \|\lambda \mathbf{s}_{k-1} - \mathbf{y}_{k-1}\|_2 = \max\left\{\lambda_0, \frac{\mathbf{s}_{k-1}^{\mathsf{T}}\mathbf{y}_{k-1}}{\mathbf{s}_{k-1}^{\mathsf{T}}\mathbf{s}_{k-1}}\right\}, \tag{6.1}$$

where $k \geq 2$. The starting parameter value λ_1 can be chosen freely, subject to the constraint $\lambda_1 \geq \lambda_0$. In our algorithm we will also use this choice of the stepsize in conjunction with the cyclic strategy used in [18]. That is, if $m \geq 1$ is the cycle length and $\ell \geq 0$ is the cycle number, then the cyclic choice for λ_k is

$$\lambda_{m\ell+i} = \lambda_{m\ell+1}^{BB} \quad \text{for } i = 1, \cdots, m. \tag{6.2}$$

Of course, when the cycle length is 1, then $\lambda_k = \lambda_k^{BB}$ for each k.

As the following proposition indicates, this choice of λ_k satisfies (3.1). For a proof of this result, see the remark that follows [34, Prop. 3.2].

PROPOSITION 6.1. *Under the hypotheses of Theorem* 4.1*, we have*

$$\lambda_0 \leq \lambda_k^{BB} \leq \overline{\lambda},$$

where $\overline{\lambda}$ is any bound for the spectral radius of the Hessian of f on the convex hull of the level set $\mathcal{L}$ in (4.3)*.*

7. Box constraints. Our analysis has focused on the nonnegativity constraint $\mathbf{x} \geq \mathbf{0}$, however, with small adjustments, ASL can be applied to the box-constrained optimization problem (1.1), similar to [34]. The modifications are as follows: The definition of $\mathbf{X}^1$ in (1.7) should be replaced by

$$X_i(\mathbf{x}, \mu) = \begin{cases} u_i - x_i & \text{if} \quad \nabla_x L_i(\mathbf{x}, \mu) \leq 0, \\ x_i - l_i & \text{otherwise.} \end{cases}$$

With the convention that $\infty \times 0 = 0$, the KKT conditions can be expressed

$$\mathbf{X}(\mathbf{x}, \mu) \circ \nabla_x L(\mathbf{x}, \mu) = \mathbf{0}, \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = b, \quad \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}.$$

With the convention that $1/\infty = 0$, the new approximation to the Newton search direction is

$$d_{ki} = -\frac{1}{\lambda_k + |\nabla_x L_i(\mathbf{x}_k, \mu_k)|/X_i(\mathbf{x}_k, \mu_k)} \nabla_x L_i(\mathbf{x}_k, \mu_k).$$

In the special case considered earlier, where $l_i = 0$ and $u_i = \infty$, we have $X_i(x, \mu) = \infty$ when $\nabla_x L_i(\mathbf{x}, \mu) \le 0$ and $|\nabla_x L_i(\mathbf{x}, \mu)|/X_i(\mathbf{x}, \mu) = 0$, exactly as in (1.5). If $l_i = -\infty$ and $u_i = \infty$ (no bound constraints), and if we make the BB choice for λ_k, then ASL reduces to a CBB method [18] in the null space of the linear constraints $\mathbf{a}^{\mathsf{T}}\mathbf{x} = b$.

8. Computation of $\boldsymbol{\mu_k}$.

The parameter μ_k in ASL can be computed using a safeguarded, Newton–Secant scheme. A Secant–Secant version of this scheme was presented in [35]; however, due to the special structure of r in (2.2), we can compute r' at the same time that we compute r. Hence, we can exploit the derivative at no additional computational cost and replace every other secant step in [35] by a Newton step.

In Figure 8.1 we illustrate the Newton–Secant iteration. We start from an interval $[a, b]$ which brackets the root α of r. By Proposition 2.2, we could take $a = \mu_0$ and $b = \mu_1$ to bracket the root. One step of Newton's method is applied, starting from a or b, whichever has the absolute smallest function value. In Figure 8.1(a), this Newton step moves the right end point b of the bracketing interval to b'. From b', we apply a secant iteration which moves the left end point a of the bracketing interval to a'. The next Newton–Secant iteration starts from a' which now has the smallest function value. The Newton iteration overshoots the root, landing at a point b'', similar to what is seen in Figure 8.1(b). The subsequent Secant iteration generates a''. The reason for alternating between a Newton and a secant step is that the combination typically updates both sides of the bracketing interval. Whenever the Newton step generates an iterate outside the bracketing interval, we should replace this iterate by one generated by a bisection step. Also, when the convergence is slower than linear, a bisection step will ensure linear convergence.

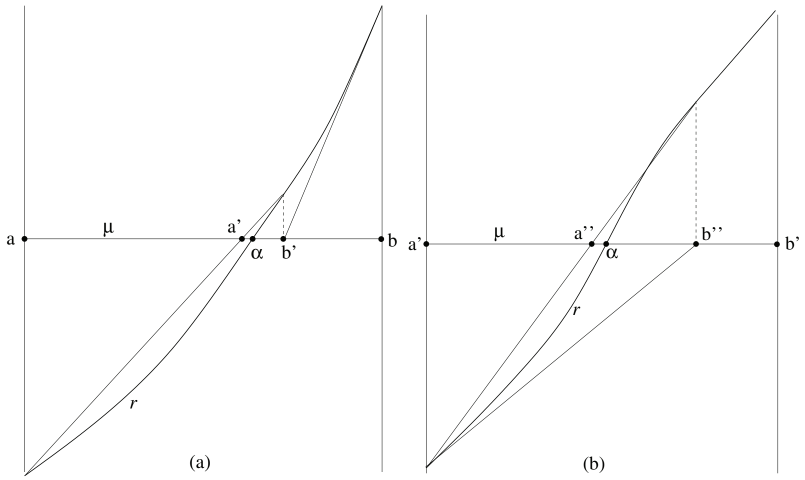

FIG. 8.1. *Newton–Secant iteration used to compute μ_k.*

Let $(a'', b'') = \text{Newton–Secant}^2(a, b)$ denote this iteration which alternates between the use of Newton's method and the secant method. Altogether Newton–Secant^2 constitutes 2 Newton steps and 2 secant steps. The Newton step is applied to the endpoint of the bracketing interval with the absolute smallest function value. The secant iteration is applied to the current bracketing interval.

THEOREM 8.1. *Suppose that $\phi : \mathbb{R} \to \mathbb{R}$ is three times continuously differentiable near a root α. If both $\phi'(\alpha) \ne 0$ and $\phi''(\alpha) \ne 0$, then, for a_0 and b_0 sufficiently close* *to α with $a_0 \le \alpha \le b_0$, the iteration*

$$(a_{k+1}, b_{k+1}) = \textit{Newton–Secant}^2(a_k, b_k)$$

converges to α, and the interval width $b_k - a_k$ tends to zero with root convergence order 7.

Proof. The condition $\phi'(\alpha) \neq 0$ ensures that α is a simple root. The condition $\phi''(\alpha) \neq 0$ ensures that, near α, each step in the iteration Newton–Secant^2 iteration generates a point on the opposite side of the root. Suppose for convenience that $\phi'(a) > 0$ and $\phi''(a) > 0$ when a is sufficiently close to α, and that $0 < \phi(b) < |\phi(a)| = -\phi(a)$; this geometry corresponds to Figure 8.1. The other cases are treated in a similar way. It is well known (e.g., see [2, p. 49]) that the error in a secant step applied to an interval $[a, b]$ bracketing the root α generates a point c which satisfies

$$c - \alpha = (a - \alpha)(b - \alpha)\Phi(a, b), \quad \Phi(a, b) = \frac{\phi''(\xi)}{2\phi''(\overline{\xi})},$$

where $\xi, \overline{\xi} \in [a, b]$. A Newton step from a generates a point c which satisfies

$$\begin{aligned} c - \alpha &= (a - \alpha)^2\Psi(a, \alpha), \\ \Psi(a, \alpha) &= \frac{\phi''(\xi)}{\phi'(\xi)} + \frac{\phi(\xi)}{\phi'(\xi)^2}\left(\phi'''(\xi) - 2\frac{\phi''(\xi)^2}{\phi'(\xi)}\right), \end{aligned} \tag{8.1}$$

where $\xi \in [a, \alpha] \subset [a, b]$. Obviously, if the Newton step starts at b, then the factor $(a - \alpha)^2$ in (8.1) should be replaced by $(b - \alpha)^2$. Since $\phi'(\alpha) > 0$, $\phi''(\alpha) > 0$, and $\phi(\alpha) = 0$, both $\Phi(a, b)$ and $\Psi(a, b)$ are positive when a and b are sufficiently close to α. Hence, a Newton iteration from either a or b generates a point $c > \alpha$, while a secant step applied to a and b generates a point $c < \alpha$. Consequently, alternating Newton and secant steps generate points on opposite sides of α when a and b are sufficiently close to α.

Since $|\phi(b)| < |\phi(a)|$, the Newton, secant, Newton, and secant steps taken by Newton–Secant^2 satisfy

$$\begin{aligned} b' - \alpha &= (b - \alpha)^2\Psi(b, \alpha), \\ a' - \alpha &= (a - \alpha)(b' - \alpha)\Phi(a, b'), \\ b'' - \alpha &= (a' - \alpha)^2\Psi(a', \alpha), \\ a'' - \alpha &= (a' - \alpha)(b'' - \alpha)\Phi(a', b''). \end{aligned}$$

Combining these relations gives

$$\begin{aligned} a'' - \alpha &= (a - \alpha)^3(b - \alpha)^6\Psi(b, \alpha)\Phi(a, b')\Psi(a', \alpha)\Phi(a', b''), \\ b'' - \alpha &= (a - \alpha)^2(b - \alpha)^4\Psi(b, \alpha)\Phi(a, b')\Psi(a', \alpha). \end{aligned}$$

We now identify a and b with a_k and b_k, and a'' and b'' with a_{k+1} and b_{k+1}. The Newton–Secant iteration is $(a_{k+1}, b_{k+1}) = \text{Newton–Secant}^2(a_k, b_k)$. Choose $\lambda \in (0, 1)$ and let $[a_0, b_0]$ be an interval containing α in its interior which is chosen small enough to ensure that both $\Phi(a, b)$ and $\Psi(a, \alpha)$ are nonnegative and bounded, and

$$\begin{aligned} \lambda &\ge (a - \alpha)^2(b - \alpha)^6\Psi(b, \alpha)\Phi(a, b')\Psi(a', \alpha)\Phi(a', b''), \\ \lambda &\ge (a - \alpha)^2(b - \alpha)^3\Psi(b, \alpha)\Phi(a, b')\Psi(a', \alpha) \end{aligned}$$

for all $a, b \in [a_0, b_0]$. It follows that

$$\mathbf{e}_{k+1} \le \lambda \mathbf{e}_k, \quad \mathbf{e}_k = \begin{bmatrix} |a_k - \alpha| \\ |b_k - \alpha| \end{bmatrix}.$$

Hence, the bracketing intervals converge at least linearly to the root.

Choose C and D such that

$$\begin{aligned} &C \ge \Psi(b,\alpha)\Phi(a,b')\Psi(a',\alpha)\Phi(a',b''), \\ &D \ge \Psi(b,\alpha)\Phi(a,b')\Psi(a',\alpha) \end{aligned}$$

for all $a, b \in [a_0, b_0]$. Consider the following recurrence:

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} = \begin{bmatrix} CA_k^3 B_k^6 \\ DA_k^2 B_k^4 \end{bmatrix}, \quad \begin{bmatrix} A_0 \\ B_0 \end{bmatrix} = \begin{bmatrix} |a_0 - \alpha| \\ |b_0 - \alpha| \end{bmatrix}.$$

Clearly, $|a_k - \alpha| \le A_k$ and $|b_k - \alpha| \le B_k$ for each k.

Make the substitution $A_k = p\overline{A}_k$ and $B_k = q\overline{B}_k$, where

$$p = \frac{\sqrt{C}}{D} \quad \text{and} \quad q = \sqrt[3]{\frac{D}{C}}.$$

The new variables satisfy the recurrence

$$\begin{bmatrix} \overline{A}_{k+1} \\ \overline{B}_{k+1} \end{bmatrix} = \begin{bmatrix} \overline{A}_k^3 \overline{B}_k^6 \\ \overline{A}_k^2 \overline{B}_k^4 \end{bmatrix}. \tag{8.2}$$

Finally, we make the change of variables

$$v_k = \log(\overline{A}_k) \quad \text{and} \quad w_k = \log(\overline{B}_k)$$

to obtain the recurrence

$$\begin{bmatrix} v_{k+1} \\ w_{k+1} \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} v_k \\ w_k \end{bmatrix}$$

from (8.2). The eigenvalues of the matrix are zero and 7, and the solution of the recurrence is

$$\begin{bmatrix} v_k \\ w_k \end{bmatrix} = 7^{k-1}(v_0 + 2w_0) \begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad k \ge 1.$$

As a result a_k and b_k converge to α with root convergence order 7. □

Remark. The root convergence rate for the Secant–Secant iteration in [35] was $1 + \sqrt{2}$ for two successive secant steps. Hence, four successive secant steps has the root convergence rate

$$(1 + \sqrt{2})^2 \approx 5.83,$$

which is slightly smaller than the root convergence rate 7 for Newton–Secant^2.

9. Subspace implementation. For problems where the evaluation of the objective function and its gradient is relatively costly, it can be more efficient to solve (1.2) through a sequence of subspace optimization problems. At iteration k, we solve the problem

$$\min\ f(\mathbf{x}) \quad \text{subject to } \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{x} \geq \mathbf{0}, \quad \mathbf{a}^{\mathsf{T}}\mathbf{x} = b, \quad \mathbf{x} \in \mathcal{S}_k, \tag{9.1}$$

where $\mathcal{S}_k$ is a subspace of $\mathbb{R}^n$. In a series of papers [60, 61, 67, 68], Serafini, Zanghirati, and Zanni show that such a subspace approach can be very effective for SVM problems. In the numerical experiments of the next section, we apply ASL to the subspace problem (9.1). In this section, we explain how we choose the subspaces $\mathcal{S}_k$ since our choice is slightly different from the scheme of Joachims [39] that was the basis for the subspace approach of Serafini, Zanghirati, and Zanni.

Let m denote a bound on the subspace size. The subspace selection scheme of Serafini, Zanghirati, and Zanni is given by a solution of the problem

$$\min\ \mathbf{g}_k^{\mathsf{T}}\mathbf{d} \quad \text{subject to} \quad \left\{ \begin{array}{l} \mathbf{a}^{\mathsf{T}}\mathbf{d} = 0, \quad -\mathbf{1} \leq \mathbf{d} \leq \mathbf{1}, \\ d_i \geq 0 \text{ if } x_{ki} = 0, \quad |\{d_i : d_i \neq 0\}| \leq m, \end{array} \right. \tag{9.2}$$

where $|\mathcal{S}|$ denote the number of elements in the set $\mathcal{S}$. The indices of the nonzero components of a solution $\mathbf{d}$ to (9.2) correspond to the subspace. Our subspace selection scheme is based on a solution of the problem

$$\min\ \frac{\lambda_k}{2}\mathbf{d}^{\mathsf{T}}\mathbf{d} + \mathbf{g}_k^{\mathsf{T}}\mathbf{d} \quad \text{subject to} \quad \left\{ \begin{array}{l} \mathbf{a}^{\mathsf{T}}\mathbf{d} = 0, \quad \mathbf{x}_k + \mathbf{d} \geq \mathbf{0}, \\ |\{d_i : d_i \neq 0\}| \leq m, \end{array} \right. \tag{9.3}$$

where $\lambda_k \geq 0$ is chosen so that $\lambda_k \mathbf{I} \approx \nabla^2 f(\mathbf{x}_k)$. An advantage of the problem (9.2) is that it has a very simple solution, as proved by Lin [45]. An advantage of (9.3) is that the objective function and constraints may provide a better model for the nonlinear optimization problem (1.1).

We obtain an approximation to a solution of (9.3) by using a heuristic algorithm modeled on the exact algorithm for (9.2). Let $\overline{\mathbf{d}}$ denote a solution of (9.3) with the sparsity constraint neglected, let ν be the Lagrange multiplier associated with the equality constraint, and let ℓ_i denote the ith term in the Lagrangian:

$$\ell_i = \frac{\lambda_k}{2}\overline{d}_i^2 + (g_{ki} + \nu a_i)\overline{d}_i.$$

We partition the indices of $\overline{\mathbf{d}}$ into two sets, $\mathcal{I}_0$ corresponding to indices i for which $a_i\overline{d}_i \leq 0$ and $\mathcal{I}_1$ corresponding to indices i for which $a_i\overline{d}_i > 0$. We build $\mathcal{S}_k$ by alternately inserting into $\mathcal{S}_k$ indices from one of the sets, say $\mathcal{I}_0$, followed by indices from the other set $\mathcal{I}_1$. We start by initializing $\mathcal{S}_k$ to be the set consisting of an index i associated with the smallest Lagrangian term of ℓ_i. Suppose $i \in \mathcal{I}_0$. Next, we insert indices from $\mathcal{I}_1$ into $\mathcal{S}_k$ using those indices for which ℓ_i is smallest, stopping as soon as

$$\sum_{i \in \mathcal{S}_k} a_i\overline{d}_i > 0. \tag{9.4}$$

The algorithm continues to alternate between $\mathcal{I}_0$ and $\mathcal{I}_1$. We always select from the remaining indices those for which ℓ_i is smallest, and we stop selection when the summation in (9.4) changes sign.

There has been much work concerning the convergence of algorithms based on subspace selections. References include [13, 42, 44, 45, 46, 52, 63]. To ensure convergence, we should include in $\mathcal{S}_k$ the index which makes ℓ_i smallest (as we do), and an index j from the opposite set, either $\mathcal{I}_0$ or $\mathcal{I}_1$, that makes $|a_j\overline{d}_j|$ largest.

10. Numerical experiments.

In this section, we present some numerical experiments to assess the performance of ASL using SVM test problems. The dual of the two-class SVM classification problem is equivalent to the following quadratic programming problem:

$$\min \ \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x} - \mathbf{1}^{\mathsf{T}}\mathbf{x} \quad \text{subject to } \mathbf{a}^{\mathsf{T}}\mathbf{x} = 0, \quad \mathbf{0} \le \mathbf{x} \le C\mathbf{1}, \tag{10.1}$$

where $\mathbf{1}$ is the vector whose entries are all 1, the vector $\mathbf{a}$ corresponds to the two data classes with $a_i = 1$ or $a_i = -1$ for each i, $\mathbf{A}$ is an n by n matrix with $A_{ij} = a_i a_j K(\mathbf{w}_i, \mathbf{w}_j)$, $\mathbf{w}_i \in \mathbb{R}^m$ is the data, $K : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ is a given "kernel function," and C is a scalar connected with the flexibility which is allowed in the data separation. Increasing C increases the penalty associated with a violation in the separation condition. The components of a solution $\mathbf{x}$ of (10.1) are the Lagrange multipliers associated with the separation condition. Generally, $\mathbf{A}$ is positive semidefinite; however, there are some kernel functions which lead to an indefinite matrix. In applications, $\mathbf{A}$ can be huge, dense, and ill-conditioned. In fact, the rank of $\mathbf{A}$ could be tiny compared to n. In particular, for a "linear kernel" $K(\mathbf{w}_i, \mathbf{w}_j) = \mathbf{w}_i^{\mathsf{T}}\mathbf{w}_j$, the rank of $\mathbf{A}$ is at most m, which can be much smaller than n, the number of data points.

Algorithms for SVM include active set methods, primal/dual interior-point methods, semismooth methods, Lagrangian method, and decomposition methods. Some references include [21, 23, 24, 25, 26, 28, 30, 37, 39, 41, 42, 43, 45, 46, 47, 49, 53, 56, 63]. At least for nonlinear kernels, block coordinate descent methods have been particularly popular. One of the reasons for the success of these approaches for nonlinear kernels is that simply evaluating a column of $\mathbf{A}$ can be expensive. If there exists an optimal solution of (10.1) which is sparse (a relatively small number of nonzero components), then by starting from the initial guess $\mathbf{x} = \mathbf{0}$, and by only changing a few components of $\mathbf{x}$ in each iteration, it may be possible to converge to an optimal solution while staying approximately within the sparsity pattern of the optimal solution. Consequently, not all the columns of $\mathbf{A}$ need to be evaluated, leading to a significant computational savings.

The experiments utilized the following codes:

- LIBSVM, version 2.91, by Chih-Chung Chang and Chih-Jen Lin [23]. The code is available at the Website [14]. In each iteration the algorithm optimizes over a working set consisting of two components of $\mathbf{x}$.
- GPDT (gradient projection-based decomposition technique) by Thomas Serafini, Luca Zanni, and Gaetano Zanghirati. The code is available at the Website [62]. The algorithm solves a series of subspace optimization problems by a gradient projection method. The dimension of the subspace can be chosen arbitrarily.

We considered three kernel functions:

1. linear: $K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathsf{T}}\mathbf{y}$,
2. radial basis function: $K(\mathbf{x}, \mathbf{y}) = \exp(-\gamma\|\mathbf{x} - \mathbf{y}\|^2)$,
3. polynomial: $K(\mathbf{x}, \mathbf{y}) = (\gamma\mathbf{x}^{\mathsf{T}}\mathbf{y})^d$,

where $\gamma = 1/m$ and $d = 3$. For these kernels, $\mathbf{A}$ is a positive semidefinite matrix.

We considered the eight test problems listed in Table 10.1. The first seven problems were obtained from the LIBSVM Website [14], while MNIST was obtained from the Website [9] of Léon Bottou. MNIST corresponds to a classification problem for the digit 8. The eight data sets are also posted at the Website [31] for this paper. The data is relatively sparse in the sense that the number of nonzeros in the data is much smaller than the product between m, the number of features, and n, the

Table 10.1
The SVM test set.

Problem Name	m (features)	n (dimension)	nnz (nonzeros in data)
A7A	122	16100	223304
A8A	123	22696	314815
A9A	123	32561	451592
IJCNN1	22	49990	649870
W6A	300	17188	200470
W7A	300	24692	288148
REAL-SIM	20958	72309	3709083
MNIST	780	60000	8994156

number of data points. When the kernel is linear, this sparsity can be exploited in the routines to evaluate the objective function. On the other hand, the **A** matrix is completely nonzero for the radial basis function kernel, and for the poly kernel, **A** is mostly nonzero. In these cases, the evaluation of the objective function and gradient is relatively slow.

The stopping criterion in the numerical experiments was based on the violation of the first-order optimality conditions. In our ASL-based algorithm for solving the SVM problem, we estimated the violation **e** in the first-order optimality conditions at a feasible point **x** and a multiplier μ as follows:

$$e_i = \begin{cases} x_i & \text{if } \nabla_x L_i(\mathbf{x}, \mu) > x_i, \\ \nabla_x L_i(\mathbf{x}, \mu) & \text{if } x_i \geq \nabla_x L_i(\mathbf{x}, \mu) \geq 0, \\ C - x_i & \text{if } -\nabla_x L_i(\mathbf{x}, \mu) > C - x_i, \\ -\nabla_x L_i(\mathbf{x}, \mu) & \text{if } C - x_i \geq -\nabla_x L_i(\mathbf{x}, \mu) \geq 0. \end{cases}$$

Note that $\mathbf{e} = \mathbf{0}$ if and only if the first-order optimality conditions are satisfied. Our stopping criterion was $\|\mathbf{e}\|_\infty \leq 10^{-3}$. With this convergence tolerance, the codes produced nearly the same objective function value to within 6 or 7 digits. Each of the codes seems to use nearly the same stopping criterion.

In our first set of experiments, we compare the performance of ASL to that of the gradient projection algorithm of Dai and Fletcher [17]. This algorithm can be selected in the GPDT code to optimize over the subspace when solving an SVM problem. When the subspace is the entire space, the gradient projection algorithm is applied to the original SVM problem. When the GPDT code is used in this way with the subspace equal to the entire space, the CPU time is abnormally large since the GPDT code is designed to be efficient when the subspaces are relatively small compared to the problem dimension. On the other hand, the number of iterations of the gradient projection algorithm in GPDT can be compared to the number of iterations of ASL since the running time of the two algorithms should, in principle, be proportional to the number of iterations (the time of an iteration would be essentially the time to update the gradient of the objective function).

Table 10.2 shows the comparison between the number of iterations of the gradient projection algorithm and ASL. In 33 of the 36 cases, ASL uses fewer iterations to satisfy the same stopping criterion. There are many cases where ASL uses on the order of half as many iterations as the gradient projection algorithm.

In the next series of experiments, we compare the performance of the SVM codes when they employ the subspace approach to solve the SVM test problems. We wrote a wrapper svmASL which constructs the subspace and which calls ASL to solve the subspace optimization problem. In the experiments, we consider four different choices

Table 10.2

The number of iterations for the ASL compared to the number of iterations for the gradient projection algorithm GP using the same convergence tolerance .001.

Problem	Kernel	C	ASL	GP
Name			Iterations	Iterations
A7A	lin	1	4088	7897
	lin	10	36325	69677
	rbf	1	206	193
	rbf	10	954	1420
	poly	1	28	44
	poly	10	64	69
A8A	lin	1	6624	11409
	lin	10	41755	80785
	rbf	1	192	266
	rbf	10	1236	1869
	poly	1	28	45
	poly	10	62	94
A9A	lin	1	6442	12369
	lin	10	69614	131139
	rbf	1	266	307
	rbf	10	1771	3230
	poly	1	33	53
	poly	10	83	98
IJCNN1	lin	1	1959	3210
	lin	10	17759	26343
	rbf	1	454	457
	rbf	10	2215	2315
	poly	1	68	89
	poly	10	104	192
W7A	lin	1	3019	5332
	lin	10	17275	35605
	rbf	1	411	723
	rbf	10	1929	4753
	poly	1	28	17
	poly	10	69	70
REAL-SIM	lin	1	384	575
	lin	10	3064	4049
	rbf	1	8	14
	rbf	10	15	27
	poly	1	3	2
	poly	10	3	3

for the upper bound in (10.1): $C = 1$, 10, 100, and 1000. The numerical experiments were performed on a single processor of a Rackable Systems shared-memory computer with eight dual-core 2.2 GHz AMD Opteron 875 processors, 1 MB cache for each processor, and 64 GB memory. Default parameter values were used for the codes with the following exception: For svmASL and GPDT, the subspace dimension was 250 for linear kernels and 450 for nonlinear kernels.

The starting guess in the codes was always $\mathbf{x}_1 = \mathbf{0}$. For the affine-scaling method developed in this paper, the iterates should lie in the relative interior of the feasible set. We handle the interior-point requirement as follows: If $x_i = 0$, then we keep $x_i = 0$ until the step along the negative gradient moves x_i into the interior of the feasible set. From that point on, x_i lies in the interior of the bound constraint.

At the Website [31] for this paper we have posted the running times, number of iterations, and final objective function values for the codes. Here we compare the running time performance of the codes using the performance profiles of Dolan and

Moré [20]. A performance profile shows the fraction P of problems for which any given method is within a factor τ of the best time. In a performance profile, the top curve is the method that solved the most problems in a time that was within a factor τ of the best time. The percentage of the test problems for which a method is the fastest is given on the left axis of the plot. The right side of the plot gives the percentage of the test problems that were successfully solved by each of the methods. In essence, the right side is a measure of an algorithm's robustness. Figures 10.1–10.3 compare the CPU time performance of the methods for each of the kernels. These plots indicate that svmASL provided the best running time performance for each of the kernels. For the linear kernel, GPDT gave better performance than LIBSVM. For the nonlinear kernels, GPDT and LIBSVM had similar performance for either small or large τ; for an intermediate range of τ, GPDT had better performance.

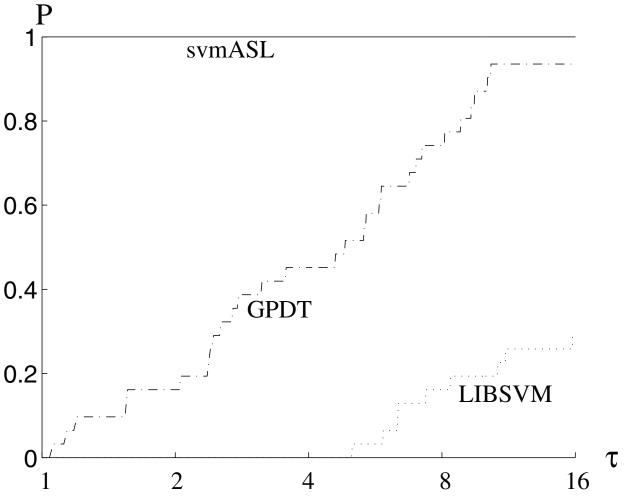

Fig. 10.1. *CPU time performance profiles for linear kernel.*

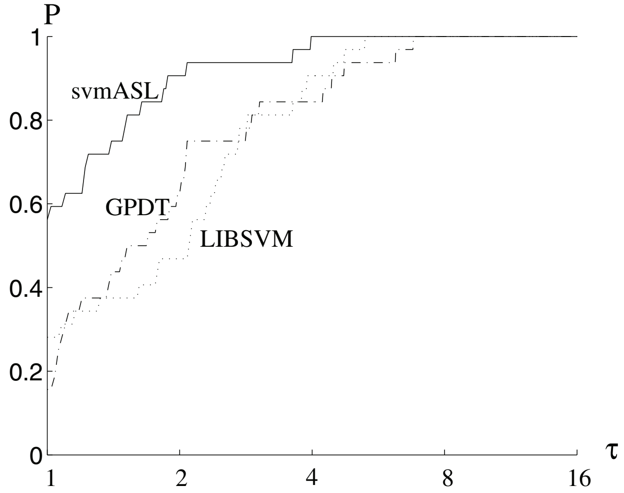

Fig. 10.2. *CPU time performance profiles for radial basis function kernel.*

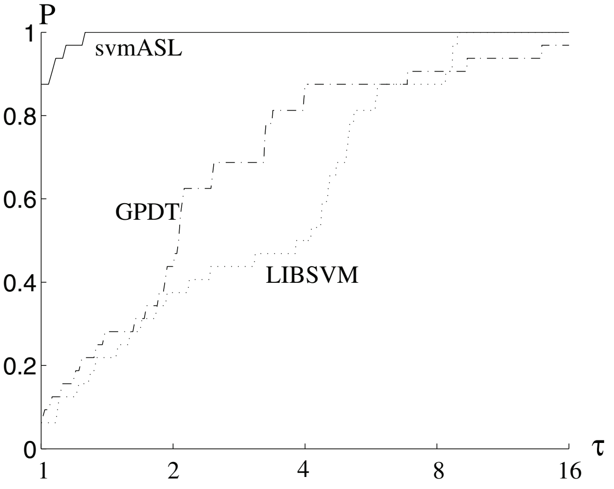

Fig. 10.3. *CPU time performance profiles for polynomial kernel.*

The codes LIBSVM, GPDT, and svmASL solve the dual formulation of the SVM problem, and they handle both linear and nonlinear kernels. Special algorithms have been developed for linear kernels. These include algorithms based on interior-point methods [28, 66], a method LIBLINEAR [22] based on dual coordinate descent, and a scheme SVMperf based on a "structural" formulation [40] of the SVM problem. CPU time comparisons among these three methods are found in [66]. In these comparisons, LIBLINEAR was relatively fast. In Table 10.3 we compare the performance of LIBLINEAR, Version 1.7 (L2 regularization and L1 loss function), to that of svmASL, the affine-scaling gradient-based method developed in this paper, and svmGP, the same code as svmASL except that the affine-scaling direction is replaced by the direction given by our gradient projection algorithm [36].

As can be seen in Table 10.3, for some problems and for C sufficiently small, LIBLINEAR is very fast. However, as C increases, either svmASL or svmGP are faster than LIBLINEAR. The reason that svmGP is faster than svmASL is that,

Table 10.3
CPU time comparisons between an algorithm, LIBLINEAR, specifically tailored to linear kernels, and both svmASL and svmGP.

Problem Name	C	LIBLINEAR Time (secs)	svmASL Time (secs)	svmGP Time (secs)
A9A	1	1.9	7.4	4.3
	10	16.0	24.1	10.3
	100	144.9	144.3	48.8
	1000	1107.7	965.1	315.5
IJCNN1	1	0.8	4.7	4.0
	10	3.0	9.8	6.4
	100	20.0	34.8	15.4
	1000	138.0	206.4	72.0
MNIST	1	greater than 24 hrs	90.8	80.7
	10	greater than 24 hrs	618.3	515.9
	100	greater than 24 hrs	5708.5	5084.5
	1000	greater than 24 hrs	58403.9	53168.5

for a linear kernel, the running time is mostly the time spent solving the subspace problem. And within the subspace, the time spent computing the search direction is a significant fraction of the solution time. Even though the algorithm developed in section 8 for computing the affine-scaling direction is fast, our algorithm for projecting a vector into a knapsack constraint is much faster. Hence, for a linear kernel, svmGP is generally more efficient than svmASL. For nonlinear kernels, svmASL can be faster than svmGP due to a smaller number of outer iterations.

11. Conclusions. The affine-scaling algorithm of [34] for general nonlinear optimization with box constraints was generalized to handle problems with an additional linear constraint $\mathbf{a}^{\mathsf{T}}\mathbf{x} = b$. The ASL was obtained by linearizing the first-order optimality conditions in $\mathbf{x}$ and approximating the Hessian of the objective function by a positive multiple of the identity matrix. This led to a nonlinear system of equations for the search direction $\mathbf{d}_k$ and the associated multiplier μ_k. The nonlinear system has a unique solution according to Proposition 2.2. It is shown in Theorem 4.1 that ASL with a nonmonotone Armijo-type line search is globally convergent to a stationary point. Theorem 5.3 establishes R-linear convergence to the global optimum when the objective function is strongly convex, the constraint multiplier is unique, and a nondegeneracy condition holds. An algorithm denoted Newton–Secant^2 could be used to compute the multiplier μ_k. Typically, successive steps of Newton–Secant^2 bracket μ_k and the width of the bracketing intervals tends to zero with root convergence order 7. We evaluated the performance of ASL using SVM test problems. In Table 10.2 we observed that the affine-scaling approach generally led to a reduction in the number of iterations when compared to a gradient projection algorithm. A subspace implementation of ASL, denoted svmASL, was developed and compared to the SVM codes LIBSVM [14, 23], based on a working set of size 2, and GPDT [60, 61, 67, 68], for which the working set was arbitrary. The profiles in Figures 10.1–10.3 indicated that svmASL gave the best CPU time performance.

Acknowledgment. Constructive comments by the reviewers are gratefully acknowledged.

REFERENCES

[1] L. Armijo, *Minimization of functions having Lipschitz continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[2] K. E. Atkinson, *An Introduction to Numerical Analysis*, John Wiley, New York, 1978.

[3] J. Barzilai and J. M. Borwein, *Two point step size gradient methods*, IMA J. Numer. Anal., 8 (1988), pp. 141–148.

[4] E. G. Birgin and J. M. Martínez, *Large-scale active-set box-constrained optimization method with spectral projected gradients*, Comput. Optim. Appl., 23 (2002), pp. 101–125.

[5] E. G. Birgin, J. M. Martínez, and M. Raydan, *Nonmonotone spectral projected gradient methods for convex sets*, SIAM J. Optim., 10 (2000), pp. 1196–1211.

[6] E. G. Birgin, J. M. Martínez, and M. Raydan, *Algorithm 813: SPG—software for convex-constrained optimization*, ACM Trans. Math. Software, 27 (2001), pp. 340–349.

[7] E. G. Birgin, J. M. Martínez, and M. Raydan, *Inexact spectral projected gradient methods on convex sets*, IMA J. Numer. Anal., 23 (2003), pp. 539–559.

[8] V. Blanz, B. Schölkopf, H. Bülthoffand, C. Burges, V. N. Vapnik, and T. Vetter, *Comparison of view-based object recognition algorithms using realistic* 3d *models*, in Artificial Neural Networks, ICIANN '96, Lecture Notes in Comput. Sci. 1112 J. V. C. von der Malsburg, W. von Seelen, and B. Sendhoff, eds., Springer, Berlin, 1996, pp. 251–256.

[9] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, *Fast kernel classifiers with online and active learning*, 2005. MNIST data set available at http://leon.bottou.org/papers/bordes-ertekin-weston-bottou-2005.

[10] B. Boser, I. Guyon, and V. Vapnik, *A training algorithm for optimal margin classifier*, in Proceedings of the 5th ACM Workshop on Computational Learning Theory, Pittsburgh, PA, 1992, pp. 144–152.

[11] C. Burges, *A tutorial on support vector machines for pattern recognition*, Data Min. Knowl. Disc., 2 (1998), pp. 121–167.

[12] C. Burges and B. Schölkopf, *Improving the accuracy and speed of support vector machines*, in Advances in Neural Information Processing Systems 9 M. Mozer, M. Jordan, and T. Petsche, eds., MIT Press, Cambridge, MA, 1997.

[13] C.-C. Chang, C.-W. Hsu, and C.-J. Lin, *The analysis of decomposition methods for support vector machines*, IEEE Trans. Neural Networks, 11 (2000), pp. 1003–1008.

[14] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[15] C. Cortes and V. Vapnik, *Support vector networks*, Mach. Learn., 20 (1995), pp. 1–25.

[16] Y. H. Dai, *Alternate stepsize gradient method*, Optimization, 52 (2003), pp. 395–415.

[17] Y. H. Dai and R. Fletcher, *New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds*, Math. Program., 106 (2006), pp. 403–421.

[18] Y. H. Dai, W. W. Hager, K. Schittkowski, and H. Zhang, *The cyclic Barzilai-Borwein method for unconstrained optimization*, IMA J. Numer. Anal., 26 (2006), pp. 604–627.

[19] Y. H. Dai and H. Zhang, *An adaptive two-point stepsize gradient algorithm*, Numer. Algorithms, 27 (2001), pp. 377–385.

[20] E. D. Dolan and J. J. Moré, *Benchmarking optimization software with performance profiles*, Math. Program., 91 (2002), pp. 201–213.

[21] J. X. Dong, A. Krzyzak, and C. Y. Suen, *Fast SVM training algorithm with decomposition on very large data sets*, IEEE Trans. Pattern Anal. Mach. Intell., 27 (2005), pp. 603–618.

[22] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, *LIBLINEAR: A library for large linear classification*, J. Mach. Learn. Res., 9 (2008), pp. 1871–1874.

[23] R.-E. Fan, P.-H. Chen, and C.-J. Lin, *Working set selection using second order information for training support vector machines*, J. Mach. Learn. Res., 6 (2005), pp. 1889–1918.

[24] M. C. Ferris and T. S. Munson, *Interior point methods for massive support vector machines*, SIAM J. Optim., 13 (2003), pp. 783–804.

[25] M. C. Ferris and T. S. Munson, *Semismooth support vector machines*, Math. Program., 101 (2004), pp. 185–204.

[26] S. Fine and K. Scheinberg, *Efficient SVM training using low-rank kernel representations*, J. Mach. Learn. Res., 2 (2001), pp. 243–264.

[27] A. Friedlander, J. M. Martínez, B. Molina, and M. Raydan, *Gradient method with retards and generalizations*, SIAM J. Numer. Anal., 36 (1999), pp. 275–289.

[28] E. M. Gertz and J. D. Griffin, *Using an iterative linear solver in an interior-point method for generating support vector machines*, Comput. Optim. Appl., (2008), DOI 10.1007/s10589-008-9228-z.

[29] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana, *Continuous characterizations of the maximum clique problem*, Math. Oper. Res., 22 (1997), pp. 754–768.

[30] T. Glasmachers and C. Igel, *Maximum-gain working set selection for SVMs*, J. Mach. Learn. Res., 7 (2006), pp. 1437–1466.

[31] M. D. Gonzalez-Lima, W. W. Hager, and H. Zhang, *An affine-scaling interior-point method for continuous knapsack constraints*, 2010. Software available at http://www.math.-ufl.edu/∼hager/papers/SVM.

[32] L. Grippo, F. Lampariello, and S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal., 23 (1986), pp. 707–716.

[33] W. W. Hager and Y. Krylyuk, *Graph partitioning and continuous quadratic programming*, SIAM J. Discrete Math., 12 (1999), pp. 500–523.

[34] W. W. Hager, B. A. Mair, and H. Zhang, *An affine-scaling interior-point CBB method for box-constrained optimization*, Math. Program., 119 (2009), pp. 1–32.

[35] W. W. Hager and H. Zhang, *A new conjugate gradient method with guaranteed descent and an efficient line search*, SIAM J. Optim., 16 (2005), pp. 170–192.

[36] W. W. Hager and H. Zhang, *A new active set algorithm for box constrained optimization*, SIAM J. Optim., 17 (2006), pp. 526–557.

[37] D. Hush, P. Kelly, C. Scovel, and I. Steinwart, *QP algorithms with guaranteed accuracy and run time for support vector machines*, J. Mach. Learn. Res., 7 (2006), pp. 733–769.

[38] J. Joachims, *Text Categorization with Support Vector Machine*, Technical report LS VIII Number 23, ftp://ftp-ai.informatik.uni-dortmund.de/pub/Reports/report23.ps.Z, University of Dortmund, 1997.

[39] T. Joachims, *Making large-scale support vector machine learning practical*, in Advances in Kernel Methods—Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, Cambridge, MA, 1998, pp. 169–184.

[40] T. Joachims, *Training linear SVMs in linear time*, in Proceedings of the ACM Conference on Knowledge Discovery and Data Mining, 2006.

[41] S. Keerthi, S. Shevade, C. Bhattacharyya, and K. Murthy, *Improvements to Platts SMO*, Neural Comput., 13 (2001), pp. 637–649.

[42] S. S. Keerthi and E. G. Gilbert, *Convergence of a generalized SMO algorithm for SVM classifier design*, Mach. Learn., 46 (2002), pp. 351–360.

[43] S. S. Keerthi and S. K. Shevade, *SMO algorithm for least-squares SVM formulations*, Neural Comput., 15 (2003), pp. 487–507.

[44] C.-J. Lin, *Linear Convergence of a Decomposition Method for Support Vector Machines*, Technical report, Department of Computater Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

[45] C.-J. Lin, *On the convergence of a decomposition method for support vector machines*, IEEE Trans. Neural Networks, 12 (2001), pp. 1288–1298.

[46] C.-J. Lin, *Asymptotic convergence of an SMO algorithm without any assumptions*, IEEE Trans. Neural Networks, 13 (2002), pp. 248–250.

[47] O. L. Mangasarian and D. R. Musicant, *Lagrangian support vector machines*, J. Mach. Learn. Res., 1 (2001), pp. 161–177.

[48] T. S. Motzkin and E. G. Strauss, *Maxima for graphs and a new proof of a theorem of Turan*, Canad. J. Math., 17 (1965), pp. 533–540.

[49] E. Osuna, R. Freund, and F. Girosi, *Improved training algorithm for support vector machines*, in Proceedings of the IEEE Workshop on Neural Networks for Signal Processing, 1997, pp. 276–285.

[50] E. Osuna, R. Freund, and F. Girosi, *Support Vector Machines: Training and Applications*, Technical report AIM-1602, C.B.C.L. Paper No. 144, MIT, Cambridge, MA, 1997.

[51] E. Osuna, R. Freund, and F. Girosi, *Training support vector machines: An application to face detection*, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1997, pp. 130–136.

[52] L. Palagi and M. Sciandrone, *On the Convergence of a Modified Version of SVM^{light} Algorithm*, Technical report, Istituto di Analisi dei Sistemi ed Informatica, Rome, 2002.

[53] J. Platt, *Fast training of support vector machines using sequential minimal optimization*, in Advances in Kernel Methods—Support Vector Learning, B. Schölkopf, C. Burges, and A. Smola, eds., MIT Press, Cambridge, MA, 1998, pp. 41–65.

[54] M. Raydan, *The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem*, SIAM J. Optim., 7 (1997), pp. 26–33.

[55] M. Raydan and B. F. Svaiter, *Relaxed steepest descent and Cauchy-Barzilai-Borwein method*, Comput. Optim. Appl., 21 (2002), pp. 155–167.

[56] K. Scheinberg, *An efficient implementation of an active set method for SVMs*, J. Mach. Learn. Res., 7 (2006), pp. 2237–2257.

[57] M. Schmidt, *Identifying speaker with support vector networks*, in Proceedings of the 28th Symposium on the Interface, Sydney, Australia, 1996.

[58] B. Schölkopf, C. Burges, and V. N. Vapnik, *Extracting support data for a given task*, in Proceedings of the First International Conference on Knowledge Discovery and Data Mining, U. M. Fayyad and R. Uthurusamy, eds., AAAI Press, Menlo Park, CA, 1995.

[59] B. Schölkopf, *Incorporating invariances in support vector learning machines*, in Artificial Neural Networks, J. V. C. von der Malsburg, W. von Seelen, and B. Sendhoff, eds., Springer, Berlin, 1996, pp. 47–52.

[60] T. Serafini, G. Zanghirati, and L. Zanni, *Gradient projection methods for quadratic programs and applications in training support vector machines*, Optim. Methods Softw., 20 (2005), pp. 353–378.

[61] T. Serafini and L. Zanni, *On the working set selection in gradient-based decomposition techniques for support vector machines*, Optim. Methods Softw., 20 (2005), pp. 583–596.

[62] T. Serafini, L. Zanni, and G. Zanghirati, *GPDT: Gradient projection-based decomposition technique*, 2004. Software available at http://dm.unife.it/gpdt.

[63] P. Tseng and S. Yun, *A coordinate gradient descent method for linearly constrained smooth optimization and support vector machines training*, Comput. Optim. Appl., (2008), DOI 10.1007/s10589-008-9215-4.

[64] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, 1995.

[65] J. Weston, A. Gammerman, M. Stitson, V. Vapnik, V. Vork, and C. Watkins, *Density estimation using support vector machines*, Technical report CSD-TR-97-23, Royal Holloway College, 1997.

[66] K. Woodsend and J. Gondzio, *Exploiting separability in large-scale linear support vector machine training*, Comput. Optim. Appl., (2009), DOI 10.1007/s10589-009-9296-8.

[67] L. Zanni, *An improved gradient projection-based decomposition technique for support vector machines*, Comput. Manag. Sci., 3 (2006), pp. 131–145.

[68] L. Zanni, T. Serafini, and G. Zanghirati, *Parallel software for training large scale support vector machines on multiprocessor systems*, J. Mach. Learn. Res., 7 (2006), pp. 1467–1492.