A FAMILY OF PROJECTED DESCENT METHODS FOR OPTIMIZATION PROBLEMS WITH SIMPLE BOUNDS

by

A. Schwartz and E. Polak

Memorandum No. UCB/ERL M95/44

9 June 1995

A FAMILY OF PROJECTED DESCENT METHODS FOR OPTIMIZATION PROBLEMS WITH SIMPLE BOUNDS

by

A. Schwartz and E. Polak

Memorandum No. UCB/ERL M95/44

9 June 1995

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A FAMILY OF PROJECTED DESCENT METHODS FOR OPTIMIZATION PROBLEMS WITH SIMPLE BOUNDS†

by

A. Schwartz and E. Polak
Department of Electrical Engineering
and Computer Sciences
University of California
Berkeley, CA 94720, U.S.A.

ABSTRACT

This paper presents a family of projected descent direction algorithms with inexact line search for solving large-scale minimization problem subject to simple bounds on the decision variables. The global convergence of algorithms in this family is ensured by conditions on the descent directions and line search. Whenever a sequence constructed by an algorithm in this family enters a sufficiently small neighborhood of a local minimizer $\hat{x}$ satisfying standard second order sufficiency conditions, it gets trapped and converges to this local minimizer. Furthermore, in this case, the active active constraint set at $\hat{x}$ is identified in a finite number of iterations. This fact is used to ensure that the rate of convergence to a local minimizer, satisfying standard second order sufficiency conditions, depends only on the behavior of the algorithm in the unconstrained subspace.

As a particular example, we present projected versions of the modified Polak-Ribière conjugate gradient method and of the limited-memory BFGS quasi-Newton method that retain the convergence properties associated with those algorithms applied to unconstrained problems.

Key words: optimization, conjugate gradient, L-BFGS, projection, simple bounds.

† The research reported herein was sponsored by the National Science Foundation grant ECS-93-02926.

1. INTRODUCTION.

Consider the problem

$$\mathbf{P} \qquad \min_{x \in \mathbb{R}^n} f(x) \quad \text{subject to } x^i \geq 0\,, \; i = 1, \ldots, n\,,$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and $x = (x^1, x^2, \ldots, x^n)$.

Algorithms for solving problem **P** based on the projection of a descent direction were first *proposed* by Goldstein [Gol.64] and Levitin and Polyak [LeP.65]. In [Ber.76], Bertsekas used the projection operator defined in [Gol.64, LeP.65] to construct a projected gradient descent algorithm with an Armijo step-size rule for solving **P**. Whenever a sequence constructed by this algorithm enters a sufficiently small neighborhood of a local minimizer $\hat{x}$ satisfying standard second order sufficiency conditions, it gets trapped and converges to this local minimizer. Furthermore, in this case, the active active constraint set at $\hat{x}$ is identified in a finite number of iterations. This fact was used in [Ber.82] to construct a modified projected Newton method, again using the projection operator defined in [Gol.64, LeP.65], with a modified Armijo step-size rule. The algorithm in [Ber.82] employs the Newton search direction only in the estimated subspace of non-binding (inactive) variables, and uses the gradient direction in the estimated subspace of binding (active) variables. Under reasonable assumptions, Bertsekas showed that his projected modified Newton method for solving **P** is globally convergent with Q-quadratic rate. The algorithm in [Ber.82] is easily extended to problems with simple bounds of the form $b_l^i \leq x^i \leq b_u^i$, $i = 1, \ldots, n$ where $b_l^i \leq b_u^i$. Bertsekas also provides an extension for handling general linear constraints of the form $b_l \leq Ax \leq b_u$. The Bertsekas projected Newton method was further extended to handle general convex constraints in [Dun.88]. In a similar vein, a globally convergence augmented Lagrangian algorithm based on a projection operator for handling simple bounds is developed in [CGT.91]. The efficiency of this family of algorithms derives from the fact that *(a)* the search direction computation is simple, *(b)* any number of constraints can be added to or removed from the active constraint set at each iteration, and *(c)* under standard second order sufficiency condition the algorithms identify the correct active constraint set after a finite number of iterations. This last fact implies that the rate of convergence depends only on the rate of convergence of the algorithm in the subspace of decision variables that are unconstrained at the solution.

Another example of a projection based algorithm for solving optimization problems with simple bounds can be found in [QuD.74] where Quintana and Davison present a *conceptual* algorithm with exact line search based upon a modified version of the Fletcher-Reeves conjugate gradient method in function space combined with a projection operator. This algorithm was intended for the solution of optimal control problems with bounded controls. The soundness of the algorithm in [QuD.74] is not clear because the proof of convergence assumes that there exists, at each iteration k, a step-size $\alpha_k > 0$ that causes a decrease in function value. However, the authors do not show that such a positive step-size exists. Furthermore, Quintana and Davison require an *a posteriori* assumption (their eqn. (26)) that is not directly related to the problem or the method under consideration.

In this paper, we extend the results in [Ber.82] by showing that the concept of Bertsekas' projection method can be used with any search direction and step size rules that satisfy general conditions similar to those in [PSS.74]. In particular, we show that our version of the projection method can be used with search directions that are determined by a conjugate gradient in the subspace of unconstrained decision variables. The extension to conjugate-gradient methods is particularly valuable for solving large-scale optimization problems with simple bound constraints, because conjugate-gradient methods do not require much additional storage or computation beyond that required by the steepest descent method but, in practice, perform considerably better than steepest descent.

The remainder of this paper consists of three sections. In Section 2, we define the projection operator and state an algorithm model for a family of projected descent methods whose search direction are required to satisfy certain conditions. We prove convergence of this algorithm model and the fact that it identifies the correct active constraint set in a finite number of iterations under second order sufficiency conditions. Several of our proofs are similar to those in [Ber.82]. As an example of the construction of admissible search directions for our algorithm, we use the Polak-Ribière conjugate gradient formula which numerical experience has shown to be more effective than the Fletcher-Reeves formulation (an explanation for this empirical result is given in [Pow.76]). We provide one example of the fact that standard rate of convergence results for the conjugate gradient method still hold for its projected version. To conclude Section 2, we describe an extension of the algorithm model that handles simple bounds of the form $b_l^i \le x^i \le b_u^i$, $i = 1, \ldots, n$. In Section 3, we present numerical results obtained in solving an optimal control problem with simple control bounds, using three implementations of our algorithm based on steepest descent, conjugate gradient, and an implementation of the limited-memory quasi-Newton method, L-BFGS (described in [Noc.80]) for the search direction computations. These numerical results indicate that the projected conjugate gradient method and the project L-BFGS method perform significantly better than the projected steepest descent method. Finally, in Section 4, we state our concluding remarks.

2. ALGORITHM MODEL FOR MINIMIZATION SUBJECT TO SIMPLE BOUNDS.

The algorithm to be presented is described with the help of the following notation: for any $z \in \mathbb{R}^n$, the projection operator $[\cdot]_+$ is given by

$$[z]_+ \triangleq \begin{bmatrix} \max\{0, z^1\} \\ \vdots \\ \max\{0, z^n\} \end{bmatrix}, \tag{2.1a}$$

and, for any search direction $d \in \mathbb{R}^n$, $x \in \mathbb{R}^n$ and step-size $\lambda \in \mathbb{R}_+$,

$$x(\lambda, d) \triangleq [x + \lambda d]_+ . \tag{2.1b}$$

For any index set $I \subset \{1, \ldots, n\}$ and $x, y \in \mathbb{R}^n$, we define $\langle x, y \rangle_I \triangleq \sum_{i \in I} x^i y^i$, and $\|x\|_I^2 \triangleq \langle x, x \rangle_I$. Without subscripts, $\langle \cdot, \cdot \rangle$ and $\|\cdot\|$ denote the Euclidean inner product and norm, respectively, on $\mathbb{R}^n$. Finally, we will let

$$\mathcal{F} \triangleq \{ x \in \mathbb{R}^n \mid x^i \geq 0, \ i = 1, \ldots, n \} \tag{2.1c}$$

denote the feasible set for problem **P**.

Definition 2.1. A point $\hat{x} \in \mathcal{F}$ is said to be a *stationary point* for the problem **P** if its directional derivative is non-decreasing in all feasible directions:

$$df(\hat{x}; x - \hat{x}) \geq 0, \ \forall x \in \mathcal{F}, \tag{2.2a}$$

or equivalently, for $i = 1, \ldots, n$,

$$\frac{\partial f(\hat{x})}{\partial x^i} \geq 0, \quad \text{and} \quad \frac{\partial f(\hat{x})}{\partial x^i} = 0 \ \text{if} \ \hat{x}^i > 0 . \tag{2.2b}$$

□

Active and almost active bounds. The projected descent algorithm model (Algorithm Model PD) which we will present requires, for each iterate x_k, the definition of sets $I_k = I(x_k) \subset \{1, 2, \ldots, n\}$ and $A_k = A(x_k) \subset \{1, 2, \ldots, n\}$. The set A_k contains the indices of the "active" or "almost active" bounds at iteration k and the set I_k is the complement of A_k in $\{1, \ldots, n\}$. With $g(x) = \nabla f(x)$, we define†

$$w(x) \triangleq \| x - [x - g(x)]_+ \| , \tag{2.3a}$$

and

$$\varepsilon(x) \triangleq \min\{ \varepsilon, w(x) \} , \tag{2.3b}$$

where $\varepsilon > 0$ is a parameter in Algorithm Model PD. We can see that $\varepsilon(x) = 0$ if and only if $x \in \mathcal{F}$ is a stationary point because the requirement that $x \in \mathcal{F}$ and that (2.2b) hold is equivalent to the requirement

† More generally, $w(x)$ can be defined as $w(x) \triangleq \| x - [x - Dg(x)]_+ \|$ where D is a positive definite diagonal matrix.

that

$$\max\{-g^i(x), -x^i\} = 0\,,\ \ i = 1, \ldots, n \tag{2.3c}$$

which, upon addition of x^i, $i = 1, \ldots, n$, to both sides, yields $[x - g(x)]_+ = x$, *i.e.* that $w(x) = 0$. Next, for $x \in \mathcal{F}$, we define

$$A(x) \triangleq \left\{ i \in 1, \ldots, n \mid 0 \le x^i \le \varepsilon(x)\,,\ g^i(x) > 0 \right\}\,, \tag{2.4a}$$

and

$$I(x) \triangleq \left\{ i \in 1, \ldots, n \mid i \notin A(x) \right\} = \left\{ i \in 1, \ldots, n \mid x^i > \varepsilon(x) \text{ or } g^i(x) \le 0 \right\}\,. \tag{2.4b}$$

To understand the logic behind the definition of the active constraint index set $A(x)$, first consider the situation corresponding to $\varepsilon = 0$. In this case, if $i \in A(x)$, $x^i = 0$ and $g^i(x) > 0$. Thus, x^i is at its bound and, moreover, any movement in $\mathcal{F}$ away from that bound will cause an increase in the objective function, hence our algorithm will leave x_i unchanged. When $\varepsilon > 0$, as we will set in Algorithm Model PD, the set $A(x)$ also includes indices of variables that are almost at their bounds and, because $g^i(x) > 0$, are likely to hit their bounds during the line search. Thus, given $x \in \mathcal{F}$, the set $A(x)$ tends to identify the active constraints at a "nearby" stationary point.

Note that in Algorithm Model PD, below, the search directions are specified only to the extent that they satisfy three conditions (stated in (2.5a,b,c)). It is clear that the direction of steepest descent, and more generally, any direction of the form $d_k = -D_k g_k$ where D_k is a symmetric, positive definite matrix that is diagonal with respect to indices $i \in A_k$ and has eigenvalues bounded from above and away from zero, satisfies these conditions. In the sequel we will show how d_k satisfying these conditions can be constructed using a conjugate gradient formula.

The most important property of Algorithm Model PD is that it identifies the correct active constraint set in a finite number of iterations. Once the correct active constraint set is identified, the active variables x_i remain at the value of zero while on the orthogonal "unconstrained" subspace Algorithm Model PD behaves as an unconstrained optimization algorithm. Because of this, the rate of convergence of Algorithm Model PD is that associated with whatever method is used to determine the components of the search direction d_k in the "unconstrained" subspace.

Algorithm Model PD:

Data: $\alpha, \beta \in (0,1), M \in \mathbb{N}, \sigma_1 \in (0,1), \sigma_2 \in (1,\infty), \varepsilon \in (0,\infty), x_0 \in \mathcal{F}$.

Step 0: Set $k = 0$.

Step 1: Compute $g_k = \nabla f(x_k)$ and set $A_k = A(x_k)$, $I_k = I(x_k)$. If $\|g_k\|_{I_k} = 0$ and $x_k^i = 0$ for all $i \in A_k$, stop.

Step 2: Select scalars m_k^i, $i \in A_k$, and a search direction d_k satisfying the following conditions:

$$d_k^i = -m_k^i g_k^i \ , \ \sigma_1 \le m_k^i \le \sigma_2 \ , \ \forall i \in A_k \ , \tag{2.5a}$$

$$\langle d_k, g_k \rangle_{I_k} \le -\sigma_1 \|g_k\|_{I_k}^2 \ , \tag{2.5b}$$

$$\|d_k\|_{I_k} \le \sigma_2 \|g_k\|_{I_k} \ . \tag{2.5c}$$

Step 3: Compute the step-size $\lambda_k = \beta^m$ where m is the smallest integer greater than $-M$ such that λ_k satisfies the Armijo-like rule:

$$f(x_k(\lambda_k, d_k)) - f(x_k) \le \alpha \left\{ \lambda_k \langle g_k, d_k \rangle_{I_k} - \langle g_k, x_k - x_k(\lambda_k, d_k) \rangle_{A_k} \right\} \ . \tag{2.6a}$$

Set

$$x_{k+1} = x_k(\lambda_k, d_k) = [x_k + \lambda_k d_k]_+ \ . \tag{2.6b}$$

Step 4: Replace k by $k+1$ and go to Step 1. □

Note that in (2.5a) one can choose $m_k^i = 1$, for all $i \in A_k$, in which case the search direction in the subspace of active constraints is the steepest descent direction.

Remark 2.2. It is easy to see that the the right-hand side of (2.6a) is non-positive. The first term of the bracketed expression is non-positive because $\langle g_k, d_k \rangle_{I_k} \le -\sigma_1 \|g_k\|_{I_k}^2$ by (2.5b). The second term is non-negative:

$$\langle g_k, x_k - x_k(\lambda_k, d_k) \rangle_{A_k} \ge 0, \tag{2.7}$$

because for all $i \in A_k$, $g_k^i > 0$ and $d_k^i = -m_k^i g_k^i \le -\sigma_1 g_k^i < 0$ and hence $x_k^i - x_k^i(\lambda, d_k) \ge 0$ for all $\lambda \ge 0$. □

Remark 2.3. The requirements in (2.5a,b,c) are the similar to those used in the Polak-Sargent-Sebastian Theorem of convergence for abstract, iterative minimization processes [PSS.74]; they ensure that d_k is bounded from above and below and that d_k does not become orthogonal to g_k. To wit, let θ_k be the angle between the vectors d_k and $-g_k$. From (2.5a,b) we have that

$$\langle d_k, g_k \rangle \le -\sigma_1 \|g_k\|_{A_k}^2 + \langle d_k, g_k \rangle_{I_k} \le -\sigma_1 \|g_k\|^2 , \tag{2.8a}$$

and from (2.5a,c)

$$\|d_k\|^2 \le \sigma_2^2 \|g_k\|_{A_k}^2 + \|d_k\|_{I_k}^2 \le \sigma_2^2 \|g_k\|^2 . \tag{2.8b}$$

Thus, $\|g_k\|^2 \ge (1/\sigma_2)\|d_k\|\,\|g_k\|$. Using this expression in (2.8a) we see that

$$\cos\theta_k = \frac{-\langle d_k, g_k \rangle}{\|d_k\|\|g_k\|} \ge \frac{\sigma_1}{\sigma_2} > 0 . \tag{2.8c}$$

□

Remark 2.4. The convergence result in Theorems 2.7 and 2.9, given below, hold for any bounded $\lambda'_k \ge 0$ such that $f(x_k(\lambda'_k)) \le f(x_k(\lambda_k, d_k))$ where λ_k satisfies the Armijo rule in (2.6a). □

Before proving convergence, we will show that the step-size rule is well defined and that the stopping criterion in Step 3 of Algorithm Model PD is satisfied by a point x_k if and only if x_k is a stationary point.

Proposition 2.5. Let x_k, d_k be any iterate and corresponding search direction constructed by Algorithm Model PD, *i.e.*, d_k satisfies the conditions in (2.5a,b,c). Then

(a) x_k is a stationary point for problem **P** if and only if $x_k(\lambda, d_k) = x_k$ for all $\lambda \ge 0$;

(b) x_k is a stationary point for problem **P** if and only if $\|g_k\|_{I_k} = 0$ and $x_k^i = 0$ for all $i \in A_k$;

(c) if x_k is not a stationary point for problem **P** then there exists $\bar{\lambda} > 0$ such that

$$f(x_k(\lambda, d_k)) - f(x_k) \le \alpha \left\{ \lambda \langle g_k, d_k \rangle_{I_k} - \langle g_k, x_k - x_k(\lambda, d_k) \rangle_{A_k} \right\} , \quad \forall \lambda \in [0, \bar{\lambda}) , \tag{2.9}$$

i.e., the step-size rule (2.6a) is well defined at x_k and will be satisfied with $\lambda_k \ge \min\{\beta^{-M}, \beta\bar{\lambda}\}$.

Proof. *(a)* Suppose that x_k is a stationary point. Then (2.2b) implies that $g_k^i = 0$ for all $i \in I_k$. Hence, $d_k^i = 0$ for all $i \in I_k$, since $\|d_k\|_{I_k} \le \sigma_2 \|g_k\|_{I_k}$. Hence $x_k^i(\lambda, d_k) = [x_k^i + \lambda d_k^i]_+ = x_k^i$ for all $\lambda \ge 0$, $i \in I_k$. Now, if $i \in A_k$, then $g_k^i > 0$ and, since x_k is stationary, it follows from (2.2b) that $x_k^i = 0$. Hence for all $i \in A_k$, $x_k^i(\lambda, d_k) = [-\lambda m_k^i g^i]_+ = 0 = x_k^i$ for all $\lambda \ge 0$. Thus, $x_k(\lambda, d_k) = [x_k]_+ = x_k$ for all $\lambda \ge 0$.

Next, suppose that $x_k(\lambda, d_k) = x_k$ for all $\lambda \ge 0$. Then $d_k^i = 0$ if $x_k^i > 0$ and $d_k^i \le 0$ if $x_k^i = 0$. Let the index sets $I_1(x_k)$, $I_2(x_k)$ be defined by

$$I_1(x_k) \triangleq \{ i \in I_k \mid x_k^i > 0 \} , \quad I_2(x_k) \triangleq \{ i \in I(x_k) \mid x_k^i = 0 \} , \tag{2.10a}$$

so that $I_k = I_1(x_k) \cup I_2(x_k)$. It follows from the above that if $i \in I_1(x_k)$, then $d_k^i = 0$, and if $i \in I_2(x_k)$, then $d_k^i \le 0$, and also $g_k^i \le 0$ (by definition of I_k since $x_k^i = 0$). Thus,

$$\langle g_k, d_k \rangle_{I_k} = \langle g_k, d_k \rangle_{I_1(x_k)} + \langle g_k, d \rangle_{I_2(x_k)} = \langle g_k, d_k \rangle_{I_2(x_k)} \ge 0 . \tag{2.10b}$$

But, from (2.5b), $\langle g_k, d_k \rangle_{I_k} \le -\sigma_1 \| g_k \|_{I_k}^2$. Therefore, $\| g_k \|_{I_k} = 0$ and hence $g_k^i = 0$ for all $i \in I_k$. For $i \in A_k$, $d_k^i = -m_k^i g_k^i < 0$. Since $x_k^i(\lambda, d_k) = x_k^i$, for all $\lambda \ge 0$, this implies that $x_k^i = 0$. Thus we have that for all $i \in I_k$, $g_k^i = 0$ and for all $i \in A_k$, $g_k^i > 0$ and $x_k^i = 0$. Consequently, x_k is a stationary point.

(b) Suppose that x_k is a stationary point. If $i \in A_k$ then, because $g_k^i > 0$ for all $i \in A_k$, it follows from (2.2b), that $x_k^i = 0$. If $i \in I_k$, then $g_k^i = 0$ since $g_k^i \le 0$ for all $i \in I_k$ and $g_k^i \ge 0$ by (2.2b) for all i such that $x_k^i = 0$. To complete the proof, suppose that $\| g_k \|_{I_k} = 0$ and $x_k^i = 0$ for all $i \in A_k$. Hence, since $x_k^i \ge 0$ for all $i \in I_k$ and, since $g_k^i > 0$ for all $i \in A_k$, it follows that (2.2b) holds for all i.

(c) Let $g \triangleq g_k$. Suppose that x_k is not a stationary point. Define the index sets $I_3(x_k)$, $I_4(x_k)$, $A_1(x_k)$, $A_2(x_k)$ as follows:

$$I_3(x_k) \triangleq \{ i \in I_k \mid x_k^i > 0 \text{, or } (x_k^i = 0 \text{ and } d_k^i > 0) \} , \quad I_4(x_k) \triangleq \{ i \in I_k \mid x_k^i = 0 , \, d_k^i \le 0 \}, \tag{2.11a}$$

$$A_1(x_k) \triangleq \{ i \in A_k \mid x_k^i > 0 \} , \quad A_2(x_k) \triangleq \{ i \in A_k \mid x_k^i = 0 \} . \tag{2.11b}$$

First note that $I_3(x_k) \cup A_1(x_k) \ne \emptyset$. To see this, suppose that $i \in I_4(x_k) \cup A_2(x_k)$ for all $i = 1, \ldots, n$. Then $x_k^i = 0$ and $d_k^i \le 0$ (since $d_k^i = -m_k^i g_k^i < 0$ for $i \in A_2(x_k)$) which implies that $x_k^i(\lambda, d_k) = [x_k^i + \lambda d_k^i]_+ = 0 = x_k^i$ for all i. Consequently, by part *(a)* of this proposition, x_k must be a stationary point, This is a contradiction. Now, let

$$\lambda_1 = \sup \{ \lambda \mid x_k^i + \lambda d_k^i \ge 0 , \; i \in I_3(x_k) \} , \tag{2.12a}$$

$$\lambda_2 = \sup \{ \lambda \mid x_k^i + \lambda d_k^i \ge 0 , \; i \in A_1(x_k) \} . \tag{2.12b}$$

Clearly $\lambda_1 > 0$ (possibly infinite) and $\lambda_2 > 0$. If $I_3(x_k)$ is empty let $\lambda_1 = \infty$ or, if $A_1(x_k)$ is empty, let $\lambda_2 = \infty$. Now, if $i \in I_4(x_k)$, $x_k^i(\lambda, d_k) = [\lambda d_k^i]_+ = 0 = x_k^i$ for all $\lambda \ge 0$. Similarly, if $i \in A_2(x_k)$, $x_k^i(\lambda, d_k) = [-\lambda m_k^i g_k^i]_+ = 0 = x_k^i$ for all $\lambda \ge 0$. On the other hand, if $i \in I_3(x_k)$ and $\lambda \in [0, \lambda_1)$, then $x_k^i(\lambda, d_k) = x_k^i + \lambda d_k^i$ and if $i \in A_1(x_k)$ and $\lambda \in [0, \lambda_2]$, then $x_k^i(\lambda, d_k) = x_k^i - \lambda m_k^i g_k^i$. Therefore, with

$$\bar{d}_k^i = \begin{cases} d_k^i & \text{if } i \in I_3(x_k) \cup A_1(x_k) \\ 0 & \text{otherwise} \end{cases} , \quad i = 1, \ldots, n , \tag{2.13a}$$

it follows that

$$x_k(\lambda, d_k) = [x_k + \lambda d_k]_+ = x_k + \lambda \bar{d}_k , \quad \forall \lambda \in [0, \min \{ \lambda_1, \lambda_2 \}) . \tag{2.13b}$$

Next, from (2.13a), we obtain

$$\langle \bar{d}_k, g_k \rangle = \langle d_k, g_k \rangle_{I_3(x_k)} + \langle d_k, g_k \rangle_{A_1(x_k)} . \tag{2.14a}$$

Now, from (2.5b), $\langle d_k, g_k \rangle_{I_k} = \langle d_k, g_k \rangle_{I_3(x_k)} + \langle d_k, g_k \rangle_{I_4(x_k)} \le -\sigma_1 \| g_k \|_{I_k}^2$. But, for $i \in I_4(x_k)$, $d_k^i \le 0$ and $g_k^i \le 0$, so $\langle d_k, g_k \rangle_{I_4(x_k)} \ge 0$. Thus, $\langle d_k, g_k \rangle_{I_3(x_k)} \le -\sigma_1 \| g_k \|_{I_k}^2$. This, together with (2.5a) and (2.14a), implies that

$$\langle \bar{d}_k , g_k \rangle \le -\sigma_1 \| g_k \|_{I_k}^2 - \sigma_1 \| g_k \|_{A_1(x_k)}^2 . \tag{2.14b}$$

Since x_k is not a stationary point, there exists at least one $i \in I_k \cup A_1$ such that $g_k^i \ne 0$. Hence $\langle \bar{d}_k , g_k \rangle < 0$, *i.e.* $\bar{d}_k$ is a feasible descent direction. Next, it follows from (2.13b) that, for all $\lambda \in [0, \min \{ \lambda_1, \lambda_2 \})$, the Armijo-like step-size rule in (2.6a) is equivalent to the following requirement on λ,

$$f(x_k + \lambda \bar{d}_k) - f(x_k) \le \alpha\lambda \langle g_k , \bar{d}_k \rangle_{A_k \cup I_3(x_k)} + \alpha\lambda \langle g_k , d_k \rangle_{I_4(x_k)} . \tag{2.14c}$$

But for all $i \in I_4(x_k)$, $g_k^i \le 0$ and $d_k^i \le 0$. Therefore the last term in the (2.14c) is non-negative. Hence (2.14c) is satisfied if the following, harder, condition is satisfied,

$$f(x_k + \lambda \bar{d}_k) - f_k \le \alpha\lambda \langle g_k , \bar{d}_k \rangle . \tag{2.14d}$$

But this is the usual Armijo rule applied to an unconstrained problem which can always be satisfied with a positive step-size when $\langle g_k , \bar{d}_k \rangle < 0$. Hence, there exists $0 < \bar{\lambda} \le \min \{ \lambda_1, \lambda_2 \}$ such that (2.9) holds. □

We will now show that Algorithm Model PD produces a sequence of iterates whose accumulation points are stationary points. The following assumption will be used:

Assumption 2.6

The gradient $\nabla f(\cdot)$ is Lipschitz continuous on bounded subsets of $\mathcal{F}$; *i.e.*, given any bounded set $S \subset \mathcal{F}$, there exist a scalar $L < \infty$ such that

$$\| \nabla f(x) - \nabla f(y) \| \le L \| x - y \| , \quad \forall x, y \in S . \tag{2.15}$$

□

Theorem 2.7. Suppose that Assumption 2.6 is satisfied. Then every accumulation point of a sequence $\{ x_k \}$, generated by Algorithm Model PD, is a stationary point of problem **P**.

Proof. First suppose $\{ x_k \}$ is a finite sequence with final index $k = k_f$. Then by Lemma 2.5*(b)*, x_{k_f} is a stationary point. If $\{ x_k \}$ is an infinite sequence then by Lemma 2.5*(b)*, it contains no stationary points. We will show by contradiction that, in this case, every accumulation of $\{ x_k \}$ is a stationary point.

For each k, $f(x_{k+1}) \le f(x_k)$ since the right hand side of (2.6a) is non-positive (*cf.* Remark 2.2). Additionally, $f(\cdot)$ is continuous and $x_k \xrightarrow{K} \hat{x}$. Thus,

$$f(x_k) - f(x_{k+1}) \to 0 . \tag{2.16a}$$

Since the bracketed expression on the right-hand side of (2.6a) is the difference of a non-negative term and a non-positive term (*cf.* Remark 2.2), (2.16a) implies that

$$\lambda_k \langle g_k, d_k \rangle_{I_k} \to 0 , \tag{2.16b}$$

$$\langle g_k, x_k - x_k(\lambda_k, d_k) \rangle_{A_k} \to 0 . \tag{2.16c}$$

We will use this to show that

$$\liminf_{k \in K} \lambda_k = 0 . \tag{2.17}$$

Since $\hat{x}$ is not a stationary point, there must exist an index j such that either

$$\hat{x}^j > 0 \quad \text{and} \quad \frac{\partial f(\hat{x})}{\partial x^j} \neq 0 \tag{2.18a}$$

or

$$\hat{x}^j = 0 \quad \text{and} \quad \frac{\partial f(\hat{x})}{\partial x^j} < 0 . \tag{2.18b}$$

If $j \in I_k$ for an infinite number of $k \in K$, then (2.17) follows immediately from (2.5b), (2.16b) and (2.18a,b). On the other hand, if $j \in A_k$ for an infinite number of $k \in K$, then we must have that $g_k^j > 0$ for all of these k, so (2.18b) cannot hold. Hence, from (2.18a),

$$\hat{x}^j > 0 \quad \text{and} \quad \frac{\partial f(\hat{x})}{\partial x^j} > 0 , \tag{2.18c}$$

and since $\partial f(\cdot) / \partial x^j$ is continuous, there exists $k_2 \in K$ such that for all $k \in K$ and $k \geq k_2$, $g_k^j > 0$. Since we also have that, for all $k \in K$ for which $j \in A_k$,

$$\langle g_k, x_k - x_k(\lambda_k, d_k) \rangle_{A_k} \geq g_k^j [x_k^j - x_k^j(\lambda_k, d_k)] \geq 0 , \tag{2.19}$$

it follows from (2.16c) and the fact that $g_k^j > 0$ for $k \geq k_2$ that $\lim_{k \in K} [x_k^j - x_k^j(\lambda_k, d_k)] = 0$. Finally, because $x_k^j > 0$ and $d_k^j = -m_k^j g_k^j < 0$ for all $k \geq k_2$ such that $j \in A_k$, we see that (2.17) holds.

The proof will now be completed by showing the $\{ \lambda_k \}$ is bounded away from zero, thereby contradicting (2.17). Since $x_k \xrightarrow{K} \hat{x}$, the subsequence $\{ x_k \}_{k \in K}$ is bounded and, by (2.5a,c) and continuity of $\nabla f(\cdot)$, $\{ d_k \}_{k \in K}$ is also bounded. This implies that $\{ x_k(\lambda_k, d_k) \}_{k \in K}$, $\lambda_k \in [0, \beta^{-M}]$, is bounded. Thus, by Assumption 2.6 we have that for $s \in [0, 1]$,

$$\| g_k - \nabla f(x_k - s[x_k - x_k(\lambda, d_k)]) \|_2 \leq sL \| x_k - x_k(\lambda, d_k) \| , \quad k \in K . \tag{2.20}$$

Expanding the left-hand side of (2.6a), we have for $k \in K$ and $\lambda \in [0, \beta^{-M}]$,

$$f(x_k(\lambda, d_k)) - f(x_k) = \langle g_k, x_k(\lambda, d_k) - x_k \rangle + \int_0^1 \langle \nabla f(x_k - s[x_k - x_k(\lambda, d_k)]) - g_k, x_k(\lambda, d_k) - x_k \rangle \, ds$$

$$\leq \langle g_k, x_k(\lambda, d_k) - x_k \rangle + \| x_k(\lambda, d_k) - x_k \| \int_0^1 sL \| x_k(\lambda, d_k) - x_k \| \, ds$$

$$= \langle g_k, x_k(\lambda, d_k) - x_k \rangle + \frac{L}{2} \| x_k(\lambda, d_k) - x_k \|^2 . \tag{2.21}$$

Now, for $i \in A_k$, $x_k^i(\lambda, d_k) = [x_k^i - \lambda m_k^i g_k^i]_+ \geq x_k^i - \lambda m_k^i g_k^i$ so that $x_k^i - x_k^i(\lambda, d_k) \leq \lambda m_k^i g_k^i$. Thus,

$$\lambda \sum_{i \in A_k} m_k^i g_k^i [x_k^i - x_k^i(\lambda, d_k)] \geq \| x_k - x_k(\lambda, d_k) \|_{A_k}^2 . \tag{2.22}$$

Now consider the sets $I_{5,k} \triangleq \{ i \in I_k \mid g_k^i > 0 \}$ and $I_{6,k} \triangleq \{ i \in I_k \mid g_k^i \leq 0 \}$. If $i \in I_{5,k}$ then $x_k^i > \varepsilon_k$ (for otherwise $i \in A_k$). Since $x_k \xrightarrow{K} \hat{x}$ and $\| \hat{x} - [\hat{x} - \nabla f(\hat{x})]_+ \| > 0$, we must have that $\liminf_{k \in K} w(x_k) > 0$. This implies that there exists $\bar{\varepsilon} > 0$ such that $\varepsilon(x_k) \geq \bar{\varepsilon}$ for all $k \in K$. Let $\Delta < \infty$ be such that $\| g_k \|_{I_k} \leq \Delta$ for all $k \in K$. Then, from (2.5c), $\| d_k \|_{I_k} \leq \sigma_2 \Delta$. So, for $\lambda \in [0, \bar{\varepsilon} / \sigma_2 \Delta]$ and $i \in I_{5,k}$, $x_k^i(\lambda, d_k) = x_k^i + \lambda d_k^i$. Thus,

$$\sum_{i \in I_{5,k}} g_k^i [x_k^i - x_k^i(\lambda, d_k)] = -\lambda \langle g_k, d_k \rangle_{I_{5,k}} , \quad \forall \lambda \in [0, \bar{\varepsilon} / \sigma_2 \Delta] . \tag{2.23a}$$

Next, for all $\lambda \geq 0$, $x_k^i - x_k^i(\lambda, d_k) \leq -\lambda d_k^i$, and since $g_k^i \leq 0$ for $i \in I_{6,k}$, we have that

$$\sum_{i \in I_{6,k}} g_k^i [x_k^i - x_k^i(\lambda, d_k)] \geq -\lambda \langle g_k, d_k \rangle_{I_{6,k}} , \quad \forall \lambda \geq 0 . \tag{2.23b}$$

Combining these last two expressions gives us

$$\langle g_k, x_k - x_k(\lambda, d_k) \rangle_{I_k} \geq -\lambda \langle g_k, d_k \rangle_{i \in I_k} , \quad \forall \lambda \in [0, \bar{\varepsilon} / \sigma_2 \Delta] . \tag{2.23c}$$

Finally, from (2.5b,c) we have that $-\langle d_k, g_k \rangle_{I_k} \geq \sigma_1 \| g_k \|_{I_k}^2 \geq \sigma_1 / \sigma_2^2 \| d_k \|_{I_k}^2$, and since, for all i and $\lambda \geq 0$, $| x_k^i - x_k^i(\lambda) | \leq \lambda | d_k^i |$, we have that

$$\| x_k - x_k(\lambda, d_k) \|_{I_k}^2 \leq \lambda^2 \| d_k \|_{I_k}^2 \leq -\lambda^2 \frac{\sigma_2^2}{\sigma_1} \langle d_k, g_k \rangle_{I_k} . \tag{2.24}$$

Thus, from (2.23c), we see that

$$\langle g_k, x_k(\lambda, d_k) - x_k \rangle = -\langle g_k, x_k - x_k(\lambda, d_k) \rangle_{A_k} - \langle g_k, x_k - x_k(\lambda, d_k) \rangle_{I_k}$$

$$\leq -\langle g_k, x_k - x_k(\lambda, d_k) \rangle_{A_k} + \lambda \langle g_k, d_k \rangle_{I_k} , \tag{2.25a}$$

and, from (2.22) and (2.24),

$$\frac{L}{2} \| x_k(\lambda, d_k) - x_k \| = \frac{L}{2} \| x_k(\lambda, d_k) - x_k \|_{A_k}^2 + \frac{L}{2} \| x_k(\lambda) - x_k \|_{I_k}^2$$

$$\leq \frac{\lambda \sigma_2 L}{2} \langle g_k, x_k - x_k(\lambda, d_k) \rangle_{A_k} - \lambda^2 \frac{\sigma_2^2 L}{2 \sigma_1} \langle g_k, d_k \rangle_{I_k} , \quad \forall \lambda \in [0, \bar{\varepsilon} / \sigma_2 \Delta] . \tag{2.25b}$$

Substituting the expressions (2.25a,b) into (2.21), we obtain that for all $\lambda \in [0, \bar{\varepsilon} / \sigma_2 \Delta]$,

$$f(x_k(\lambda, d_k)) - f(x_k) \le \lambda\left(1 - \lambda\frac{\sigma_2^2 L}{2\sigma_1}\right)\langle g_k, d_k\rangle_{I_k} + \left(\frac{\lambda\sigma_2 L}{2} - 1\right)\langle g_k, x_k(\lambda, d_k) - x_k\rangle_{A_k} . \qquad (2.26)$$

Comparing this with (2.6a), and noting from (2.5b) and (2.7) that $\langle g_k, d_k\rangle_{I_k} \le 0$ and $\langle g_k, x_k(\lambda) - x_k\rangle_{A_k} \ge 0$, we see that the Armijo-like rule is satisfied for any $\lambda \ge 0$ such that $\lambda \le \bar{\varepsilon}/\sigma_2\Delta$, $\lambda - \lambda^2 L\sigma_2^2/2\sigma_1 \ge \alpha\lambda$ and $\lambda\sigma_2 L/2 - 1 \le -\alpha$. Since $\beta \in (0,1)$ and the step-size rule requires the smallest m such that $\lambda = \beta^m$ satisfies (2.6a), we see that for all $k \in K$,

$$\lambda_k \ge \min\left\{ \frac{\bar{\varepsilon}}{\sigma_2\Delta}, \frac{2\sigma_1(1-\alpha)}{\sigma_2^2 L}, \beta^{-M} \right\} > 0 . \qquad (2.27)$$

This contradicts (2.17) and proves the result. □

Next, we proceed towards a proof that under suitable conditions, after a finite number of iterations, Algorithm Model PD reverts to an unconstrained optimization algorithm on the subspace defined by the non-binding variables at a strict local minimizer limit point. Let $B(x)$ denote the set of all binding constraints at x, *i.e.*,

$$B(x) \triangleq \{ i \mid x^i = 0 \} , \ \forall x \ge 0 . \qquad (2.28)$$

We will use the following alternative statement of the standard second order sufficiency condition with strict complementary slackness for a point $\hat{x}$ to be a strict local minimizer for problem **P**†

$$z^T \nabla^2 f(\hat{x}) z > 0 , \ \forall z \in \{ z \in \mathbb{R}^n \mid z^i = 0 , \ \forall i \in B(\hat{x}) \} , \qquad (2.29a)$$

and

$$\frac{\partial f(\hat{x})}{\partial x^i} > 0 , \ \forall i \in B(\hat{x}) . \qquad (2.29b)$$

□

In Lemma 2.8, below, and in the proof of Theorem 2.9, to follow, let $\mathcal{B}(\hat{x}, \rho) \triangleq \{ x \in \mathbb{R}^n \mid \| x - \hat{x} \| \le \rho \}$ denote the closed ball of radius ρ around $\hat{x}$. The script $\mathcal{B}$ used here should not be confused with the B used in (2.28) to define the binding constraint set.

Lemma 2.8. Suppose that $f(\cdot)$ is twice continuously differentiable and that $\{ x_k \}_{k=0}^{\infty}$, $x_k \in \mathcal{F}$, is a sequence with an accumulation point $\hat{x}$ which satisfies the sufficient condition (2.29a,b). Let $\hat{\rho}$ be the radius of attraction for $\hat{x}$, *i.e.* $f(x) < f(\hat{x})$ for all $x \in \mathcal{B}(\hat{x}, \hat{\rho})$, $x \ne \hat{x}$. If there exists a $\hat{\rho} > 0$ such that $\| x_{k+1} - x_k \| < \hat{\rho}$ for any k such that $x_k \in \mathcal{B}(\hat{x}, \hat{\rho})$, then $x_k \to \hat{x}$ as $k \to \infty$.

† See, for example, [Lue.84], pp. 316-317: *viz.*, equation (2.2b) holds and the Hessian of the Lagrangian $L(x) \triangleq f(x) - \mu^T x$ is positive definite on the subspace $\{ z \in \mathbb{R}^n \mid z^i = 0 , \ i \in B(\hat{x}) \}$ and the multipliers $\mu^i = \partial f(\hat{x})/\partial x^i$ are positive for all $i \in B(\hat{x})$ and zero for all $i \notin B(\hat{x})$.

Proof. Let $\rho = \min\{\hat{\rho}, \hat{\rho}\}$. Since $\hat{x}$ is a strict local minimizer and $\rho \leq \hat{\rho}$, there exists an $c > f(\hat{x})$ such that the level set $L_c \triangleq \{x \in \mathcal{F} \mid f(x) \leq c\}$ satisfies $L_c \cap \mathcal{B}(\hat{x}, 2\rho) \subset \mathcal{S}(\hat{x}, \rho)$. Since $\hat{x}$ is an accumulation point of $\{x_k\}$, there exists $N < \infty$ such that $x_N \in L_c \cap \mathcal{B}(\hat{x}, \rho)$. Since $\rho < \hat{\rho}$, $\|x_{N+1} - x_N\| < \hat{\rho}$, thus, $x_{N+1} \in \mathcal{B}(\hat{x}, 2\rho)$. But since $f(x_{N+1}) < f(x_N)$, we see that $x_{N+1} \in L_c$. Hence, $x_{N+1} \in L_c \cap \mathcal{B}(\hat{x}, \rho)$. This argument can be repeated indefinitely to show that $x_k \in \mathcal{B}(\hat{x}, \hat{\rho})$ for all $k \geq N$. Finally, since $\{f(x_k)\}$ is monotone decreasing and has an accumulation point $f(\hat{x})$, $f(x_k) \to f(\hat{x})$ as $k \to \infty$. Now, since $x_k \in \mathcal{B}(\hat{x}, \hat{\rho})$ for all $k \geq N$, all of the accumulation points of $\{x_k\}$ must be in $\mathcal{B}(\hat{x}, \hat{\rho})$, and furthermore, if $\hat{\hat{x}} \neq \hat{x}$ is an accumulation point of $\{x_k\}$ then we must have that $f(\hat{\hat{x}}) = f(\hat{x})$. But this is a contradiction since $f(\hat{x}) < f(x)$ for all $x \in \mathcal{B}(\hat{x}, \hat{\rho})$ such that $x \neq \hat{x}$. Thus, $x_k \to \hat{x}$ as $k \to \infty$. □

Theorem 2.9. Suppose Assumptions 2.6 holds and $f(\cdot)$ is twice continuously differentiable. Consider a sequence $\{x_k\}$ produced by Algorithm Model PD. If $\{x_k\}$ has an accumulation points $\hat{x}$ such that (2.29a,b) hold, then $x_k \to \hat{x}$ and

$$A_k = B(x_k) = B(\hat{x}), \quad \forall k \geq N+1. \tag{2.30}$$

Proof. Because $\hat{x}$ is a local minimizer, it follows from (2.2b) and the definition of $w(x)$ in (2.3a), that $w(\hat{x}) = 0$. It is also clear that $w(\cdot)$ is continuous. Thus, there exists a $\rho_1 > 0$ such that $\varepsilon(x) \triangleq \min\{\varepsilon, w(x)\} = w(x)$ for all $x \in \mathcal{B}(\hat{x}, \rho_1)$. Since $g(\cdot) \triangleq \nabla f(\cdot)$ is continuous, we have by (2.29b) that for x sufficiently close to $\hat{x}$, $g^i(x) \geq x^i$ for all $i \in B(\hat{x})$. Thus, there exists $\rho_2 \in (0, \rho_1]$ such that for all $x \in \mathcal{B}(\hat{x}, \rho_2)$

$$[x^i - g^i(x)]_+ = 0, \quad \forall i \in B(\hat{x}) \tag{2.31a}$$

$$g^i(x) > 0, \quad \forall i \in B(\hat{x}). \tag{2.31b}$$

From (2.31a) it follows that for $x \in \mathcal{B}(\hat{x}, \rho_2)$,

$$\varepsilon^2(x) = w^2(x) = \|x - [x - g(x)]_+\|^2 = \sum_{i \in B(\hat{x})} (x^i)^2 + \sum_{i \notin B(\hat{x})} (x^i - [x^i - g]_+)^2 \geq \sum_{i \in B(\hat{x})} (x^i)^2. \tag{2.32a}$$

Hence,

$$x^i \leq \varepsilon(x), \quad \forall i \in B(\hat{x}) \text{ and } x \in \mathcal{B}(\hat{x}, \rho_2). \tag{2.32b}$$

Also, since $(\hat{x})^i > 0$ for all $i \notin B(\hat{x})$ and $\varepsilon(\hat{x}) = w(\hat{x}) = 0$, there exist scalars $\bar{\varepsilon} > 0$ and $\rho_3 \in (0, \rho_2]$ such that

$$x^i > \bar{\varepsilon} > \varepsilon(x)\,,\ \ \forall i \notin B(\hat{x}) \text{ and } x \in \mathcal{B}(\hat{x}, \rho_3)\,. \tag{2.33}$$

Thus, we see from (2.31b), (2.32b), (2.33) and the definition of $A(x)$ given in (2.4a), that

$$A(x) = B(\hat{x})\,,\ \ \forall x \in \mathcal{B}(\hat{x}, \rho_3)\,. \tag{2.34}$$

Now for any k, i such that $x_k \in \mathcal{B}(\hat{x}, \rho_3)$ and $i \in A_k$ we have by, definition of A_k, that $d_k^i = -m_k^i g_k^i < 0$ and $x_k^i \le \varepsilon(x_k)$ and by (2.34), $i \in B(\hat{x})$. Thus, from (2.29b), the continuity of $\nabla f(\cdot)$ and the fact that $\lambda_k \ge \bar{\lambda}$ for all k, where $\bar{\lambda}$ is given by (2.27), there exists a $\rho_4 \in (0, \rho_3]$ such that for any $i \in A_k$ and $x_k \in \mathcal{B}(\hat{x}, \rho_4)$, $0 \le x_{k+1}^i = [x_k^i - \lambda_k m_k^i g_k^i]_+ \le [x_k^i - \bar{\lambda} m_k^i g_k^i]_+ = 0$, which implies that $i \in B(x_{k+1})$. On the other hand, for any k, i such that $x_k \in \mathcal{B}(\hat{x}, \rho_4)$ and $i \notin A_k$, we have from (2.34) that $i \notin B(\hat{x})$ and hence, by (2.33), $x_k^i > \bar{\varepsilon}$. Since $\hat{x}$ is a local minimum, we see from (2.2b) that $\partial f(\hat{x}) / \partial x^i = 0$ for all $i \in B(\hat{x})$. Also, λ_k is bounded, $\varepsilon(\cdot)$ is continuous, and by (2.5c), $\|d_k\|_{I_k} < \sigma_2 \|g_k\|_{I_k}$. Therefore, since $\|x_{k+1} - x_k\| \le \|x_{k+1} - x_k\|_{A_k} + \|x_{k+1} - x_k\|_{I_k} \le \varepsilon(x_k) + \sigma_2 \|g_k\|_{I_k}$ is arbitrarily small for x_k sufficiently close to $\hat{x}$, there exists $\rho_5 \in (0, \rho_4]$ such that if $x_k \in \mathcal{B}(\hat{x}, \rho_5)$, then *(i)* $x_{k+1} \in \mathcal{B}(\hat{x}, \rho_4)$, *(ii)* $x_{k+1}^i > \varepsilon(x_{k+1})$ for $i \notin A_k$ which implies that $i \notin A_{k+1}$, and *(iii)* $\|x_{k+1} - x_k\| \le \hat{\rho}$. It follows from these arguments that

$$B(x_{k+1}) = A_{k+1} = A_k = B(\hat{x})\,,\ \ \forall k \text{ such that } x_k \in \mathcal{B}(\hat{x}, \rho_5)\,, \tag{2.35}$$

and from Lemma 2.8, $x_k \to \hat{x}$. Thus, there exists $N < \infty$ such that $x_k \in \mathcal{B}(\hat{x}, \rho_5)$ for all $k \ge N$ and therefore (2.30) holds for all $k \ge N + 1$. □

Remark 2.10. In [Ber.82], the search directions are given by $d_k = -D^k g_k$ where the D^k are symmetric, positive definite matrices, with elements D_{ij}^k, that are diagonal with respect to the indices $i \in A_k$, *i.e.*,

$$D_{ji}^k = D_{ij}^k = 0\ ,\ \forall i \in A_k\ ,\ j = 1, 2, \ldots, n\ ,\ j \ne i\ , \tag{2.36a}$$

and are required to satisfy

$$\gamma_1 w_k^{q_1} \|z\|^2 \le z^T D^k z \le \gamma_2 w_k^{q_2} \|z\|^2\,,\ \ \forall z \in \mathbb{R}^n\ , \tag{2.36b}$$

where γ_1 and γ_2 are positive scalars and q_1 and q_2 are non-negative integers. It is easy to see that in the case $q_1 = q_2 = 0$, with $\gamma_1 \in (0, 1)$ and $\gamma_2 \in (1, \infty)$, $d_k = -D_k g_k$ will satisfy the conditions required by (2.5a,b,c). If we replace the constants σ_1 and σ_2 by $\sigma_1 w_k^{q_1}$ and $\sigma_2 w_k^{q_2}$, respectively, in (2.5a,b,c), the Bertsekas search directions satisfy these tests for all non-negative, integer q_1 and q_2. □

It follows from Theorem 2.9 that, under the conditions stated, Algorithm Model PD will identify the constrained components of the solution $\hat{x}$ after a finite number of iterations N. Hence, for all $k \ge N + 1$, $x_k^i = 0$ if $i \in B(\hat{x})$ and $x_k^i > 0$ if $i \notin B(\hat{x})$. Consequently, for all $k \ge N + 1$, Algorithm Model PD reduces to an unconstrained optimization algorithm on the subspace $\{ x \in \mathbb{R}^n \mid x^i = 0 , \forall i \in B(\hat{x}) \}$ and its rate of convergence is governed entirely by the rules used in the construction of the components d_k^i, $i \in I_k$, of the search direction.

This fact is reflected in Corollary 2.11 which deals with the rate of convergence of an implementation of Algorithm Model PD that uses the Polak-Ribière conjugate gradient rule for constructing the components d_k^i, $i \in I_k$, of the search direction d_k. The corollary states that Algorithm Model PD, with search directions d_k given by (2.37a,b,c), exact line searches and restarts imposed every $m+1$ iterations, has iterates that converge $m+1$-step linearly with a root rate constant that depends on only the smallest $n-r-m$ eigenvalues of the Hessian at the solution restricted to the unconstrained subspace where $r = |B(\hat{x})|$ is the number of constraints binding at the solution. Since it follows from the interlacing eigenvalue property of symmetric matrices [GoL.89, Cor. 8.1.4] that the condition of the restricted Hessian is no worse than that of the Hessian itself, the presence of bounds on the decision variables can only serve to reduce the convergence rate constant. For problems that include penalty functions, if m is taken to be the number of penalized constraints then Corollary 2.11 shows that the $m+1$-step convergence root rate constant is independent of the size of the penalty constant (see [Lue.71]).

Corollary 2.11. Suppose that

(a) in problem **P**, $f(\cdot)$ is three times continuously differentiable with positive definite Hessian $H(x)$ and that $\hat{x}$, the unique global minimizer of **P**, satisfies the sufficient conditions (2.29a,b)†, and that $B(\hat{x}) \triangleq \{ n-r+1, \ldots, n \}$, with $1 \le r \le n$ (achieved by renumbering the variables, if necessary).

(b) $\{ x_k \}$ is a sequence produced by Algorithm Model PD with search directions d_k determined as follows (with $d_{-1} = 0 \in \mathbb{R}^n$):

$$\tilde{d}_k^i = -g_k + \mu_k d_{k-1} , \quad i \in I_k , \quad \text{where } \mu_k = \frac{\langle g_k , g_k - g_{k-1} \rangle_{I_k}}{\| g_k \|_{I_k}^2} , \tag{2.37a}$$

$$d_k^i = \begin{cases} \tilde{d}_k^i & \text{if } \langle \tilde{d}_k , g_k \rangle_{I_k} \le -\sigma_1 \| g_k \|_{I_k}^2 \text{ and } \| \tilde{d}_k \|_{I_k} \le \sigma_2 \| g_k \|_{I_k} \\ -g_k^i & \text{otherwise} \end{cases} , \quad \forall i \in I_k , \tag{2.37b}$$

$$d_k^i = -m_k^i g_k^i , \quad \forall i \in A_k , \tag{2.37c}$$

and with the step size λ_k determined by an exact line search, and with restarts imposed every $m+1 \le n-r$ iterations.

† The convexity of the constraint set and the strict convexity of the objective function guarantees that **P** has a unique global minimizer.

Let $H_{1,1}(\hat{x})$ denote the upper-left $(n-r)\times(n-r)$ diagonal block of $H(\hat{x})$, and let a denote its minimum eigenvalue and b its $(m+1)$-th largest ($(n-m-r)$-th smallest) eigenvalue. If, in Algorithm Model PD, $\sigma_2 \geq 1 + b/a$, then for any $\delta > 0$ there exists $N < \infty$ such that for all $k \geq N/(m+1)$,

$$\| x_{(k+n)(m+1)} - \hat{x} \| \leq c_k \left[\frac{b-a}{b+a} + \delta \right]^n , \quad n = 0,1,2,\ldots \tag{2.38}$$

where c_k is a bounded constant.

Proof. Clearly $f(\cdot)$ satisfies Assumption 2.6. Hence, by Theorem 2.9, there exists $N_1 < \infty$ such that for all $k \geq N_1+1$, $B(x_k) = B(\hat{x})$, *i.e.*, for all $k \geq N_1+1$, $x_k^i = 0$ for $i \in B(\hat{x})$ and for all $i \notin B(\hat{x})$, d_k^i is determined by equations (2.36), (2.37a,b). Furthermore, it can be shown that with the choice for σ_2 given in the Corollary statement, the tests in (2.37a) will not fail for $k \geq N_1+1$. Thus, the search direction $\tilde{d}_k = (d_k^1 \cdots d_k^r)$, $k \geq N+1$, is determined by the unconstrained, partial conjugate gradient method, with restarts every $m+1$ iterations, applied to the unconstrained subspace $\{ x \in \mathbb{R}^n \mid x^i = 0, \; i \in B(\hat{x}) \}$. It follows from Corollary 5.1 in [Lue.71] that there exists a finite $N > N_1$ such that (2.38) holds. □

Remark 2.12. In general, when exact line searches are not used, the tests in (2.37a), needed to ensure global convergence, provide an automatic resetting mechanism for the conjugate gradient method. It is possible to prove global convergence for a modified version of the PR conjugate gradient method without requiring any restarts [GiN.92]. However, the step-size λ_k must then satisfy stronger conditions than the Armijo-like rule. The accuracy of the line search also affects the conjugacy of the directions obtained by conjugate gradient algorithms. It is therefore common practice to apply a criterion more stringent than the Armijo rule for accepting a step-size (for instance, the step-size might also be required to satisfy the strong Wolfe condition, see [GiN.92]). The convergence results of Theorems 2.7 and 2.9 will still hold for more accurate line searches (*cf.* Remark 2.4). □

Extension to upper and lower bounds. Algorithm Model PD can easily be extended to deal with upper and lower bounds of the form $b_l^i \leq x^i \leq b_u^i$, $i = 1,\ldots,n$. Merely replace the projection operator $[\cdot]_+$ with the projection operator $[\cdot]_\#$, defined for $z \in \mathbb{R}^n$ and $i = 1,\ldots,n$, by

$$[z]_\#^i \triangleq \begin{cases} b_l^i & \text{if } z^i \leq b_l^i , \\ z^i & \text{if } b_l^i < z^i < b_u^i , \\ b_u^i & \text{if } z^i \geq b_u^i , \end{cases} \tag{2.40}$$

and replace the set $A_k = A(x_k)$ with

$$A_k \triangleq \{ i \mid b_l^i \leq x_k^i \leq b_u^i + \varepsilon(x_k) \text{ and } g_k^i > 0 , \text{ or } b_l^i - \varepsilon(x_k) \leq x_k^i \leq b_u^i \text{ and } g_k^i < 0 \} . \tag{2.41}$$

The set I_k is defined, as before, as the complement of A_k in $\{1,2,\ldots,n\}$.

3. COMPUTATIONAL RESULTS.

One source of large-scale optimization problems is discretizations of optimal control problems. An optimal control problem can be discretized by replacing the differential equations describing the system dynamics with a system of difference equations that describes some integration algorithm applied to the differential equations, and by replacing the infinite dimensional function space of controls with a finite dimensional subspace of parameterized controls. The result is a standard nonlinear programming problem whose decision variables are the control parameters. The number of decision variables in the nonlinear program is equal to the dimension of the approximating control subspace.

We used Algorithm Model PD to solve a discretization of the following optimal control problem:

OCP:
$$\underset{u \in L_2[0,2.5]}{\text{minimize}} \left[Cx_1^2(2.5) + \int_0^{2.5} x_1^2(t) + u^2(t)\, dt \right]$$

subject to

$$\dot{x}_1(t) = x_2(t) \;\; ; \;\; x_1(0) = -5 \,,$$

$$\dot{x}_2(t) = -x_1(t) + [1.4 - 0.14x_2^2(t)]x_2(t) + 4u(t) \;\; ; \;\; x_2(0) = -5 \,.$$

$$u(t) \geq -4|t - 1.5| \,, \;\; \forall t \in [0, 2.5] \,,$$

with C a parameter.

The discretization was carried out using a second order Runge-Kutta method (the explicit trapezoidal rule) with $u(\cdot)$ restricted to the subspace of continuous, piecewise linear functions. Specifically, the decision variable for the discretized problem are $u = (u^0, u^1, \ldots, u^n) \in \mathbb{R}^n$ where $u^i = u(t_i)$ are the values of $u(\cdot)$ at the breakpoints $t_i = i(2.5/1000)$, $i = 0, \ldots, 1000$. Thus $n = 1001$. The use of Runge-Kutta integration for discretization of optimal control problems is described in detail in [ScP.95] where the discretized problems are shown to be "consistent approximations" to the original optimal control problem. We utilized the natural coordinate transformation described in [ScP.95] associated with this discretization in order to prevent unnecessary ill-conditioning. For this case, the transformation is given by $\tilde{u} = \mathbf{M}_N^{1/2} u$ where, due to continuity imposed at the breakpoints t_k, $\mathbf{M}_N$ is an $n \times n$ diagonal matrix with diagonal$(M) = [\tfrac{1}{2}\ 1\ 1\ \cdots\ 1\ 1\ \tfrac{1}{2}]/N$†.

In addition to the coordinate transformation required by the theory of consistent approximations, we also pre-scale the problem by multiplying $f(\cdot)$ by the factor γ, where γ is defined by

† Other coordinate transformations are also sometimes useful. For example, a coordinate transformation for use with the conjugate gradient method applied to optimal control problems with a special structure is discussed, and shown to be extremely effective, in [Ber.74]. Another possibility is to use the inverse of the diagonal of the Hessian. This matrix can be efficiently computed using a recursive algorithm similar to the one described in [DuB.89]. However, our experience indicates that this is not effective for optimal control problems discretized as described above.

$$S = (1 + \|u_0\|_\infty) / (100 \|g_0\|_\infty) \tag{3.1a}$$

$$\delta u = [u_0 - S g_0]_\# , \tag{3.1b}$$

$$\gamma = \frac{1}{2} \left| \frac{\langle \delta u , \delta u \rangle_{I_0}}{f(u_0 + \delta u) - f(u_0) - \langle g_0, \delta u \rangle_{I_0}} \right| , \tag{3.1c}$$

with $[\cdot]_\#$ is given by (2.40). This pre-scaling makes it likely that a step-size of one is accepted in the first iteration of the algorithm (γ is the distance along the projected steepest descent direction, δu, to the minimum of a quadratic fit to $f(\cdot)$) and it acts as a normalization on the problem so that the tests in (2.5a,b,c) and the numerical termination criterion are less scale sensitive.

For purposes of comparison, we solved the discretized optimal control problem with a projected steepest descent algorithm, a projected conjugate gradient method and a projected version of the limited memory quasi-Newton algorithm (L-BFGS) presented in [Noc.80,LiN.89].

The projected steepest descent algorithm uses the search directions $d_k = -g_k$.

For the projected conjugate gradient method we used the search directions given in equations (2.37a,b,c) with $m_k^i = 1$ for all $i \in A_k$. After computing a step-size λ_k that satisfies the Armijo-like rule in (2.6a), we construct a quadratic approximation $q(\lambda)$ to $f(u_k(\lambda, d_k))$ such that $q(0) = f(u_k)$, $q'(0) = g_k$ and $q(\lambda_k) = f(u_k(\lambda_k, d_k))$ and set λ'_k equal to the minimizer of this quadratic. If $f(u_k(\lambda'_k) < f(u_k(\lambda_k))$, then λ'_k is chosen as the next step-size. Otherwise λ_k is used. This procedure requires one extra function evaluation per iteration and does not affect the convergence proofs (*cf.* Remark 2.12).

The L-BFGS algorithm computes an approximation G_k to the inverse of the Hessian based on a limited number of applications of the BFGS quasi-Newton update formula. At each iteration, the algorithm uses vectors $s_k \triangleq u_k - u_{k-1}$ and $y_k \triangleq g_k - g_{k-1}$ stored over a fixed number of previous iterations. The procedure is as follows: Let $\hat{m} = \min\{k, m-1\}$. Then, at iteration k, G_k is computed from $\hat{m}+1$ BFGS updates applied to a starting estimate G_k^0 of the inverse Hessian (which can be different at each iteration) according to

$$\begin{aligned} G_{k+1} = {} & (V_k^T \cdots V_{k-\hat{m}}^T) G_k^0 (V_{k-\hat{m}} \cdots V_k) \\ & + \rho_{k-\hat{m}} (V_k^T \cdots V_{k-\hat{m}+1}^T) s_{k-\hat{m}} s_{k-\hat{m}}^T (V_{k-\hat{m}+1} \cdots V_k) \\ & + \rho_{k-\hat{m}+1} (V_k^T \cdots V_{k-\hat{m}+2}^T) s_{k-\hat{m}} s_{k-\hat{m}}^T (V_{k-\hat{m}+2} \cdots V_k) \\ & \vdots \\ & + \rho_k s_k^T , \end{aligned} \tag{3.2}$$

where $\rho_k = 1/(y_k^T s_k)$, $V_k = I - \rho_k y_k s_k^T$. Here we use I (without any subscript) to denote the $n \times n$ identity matrix. As in [LiN.89], we let $G_k^0 = I$ in the first iteration and during restarts, and on other iterations we let $G_k^0 = \gamma_k I$ where $\gamma_k = \langle y_k, s_k \rangle_{I_k} / \| y_k \|_{I_k}^2$ is a self-scaling term demonstrated in [ShP.78] to markedly improve the performance of quasi-Newton algorithms. An efficient, recursive procedure for computing $G_k g_k$ without explicitly forming any matrices is given in [Noc.80].

The L-BFGS algorithm has proven to be quite effective [ZNB.93]. We used this algorithm, with $\hat{m} = 12$, restricted to the unconstrained subspaces $\{ x \in \mathbb{R}^n \mid x^i = 0 \,,\, i \in A_k \}$, to compute $d_k^i, i \in I_k$. In (2.5a), we chose $m_k^i = \gamma_k$ for all $i \in A_k$. We also added the following tests:

(i) Do not use current information y_k and s_k for Hessian updates if $\langle y_k, s_k \rangle_{I_k} < -0.001 \| g_k \|_{I_k}^2$.

(ii) Restart if $\| d_k \|_{I_k}^2 > 1000 \gamma_k \| g_k \|_{I_k}^2$,

(iii) Restart if $\langle d_k, g_k \rangle_{I_k} > -0.2 \gamma_k \| g_k \|_{I_k}^2$,

With the tests *(ii)* and *(iii)*, the search direction d_k satisfies the conditions in (2.5a,b,c) and therefore this method is convergent.

The data required by Algorithm Model PD were chosen as follows: $\alpha = 1/2$, $\beta = 3/5$, $M = 20$, $u_0^i = 0$ for $i = 1, \ldots, n$, and $\varepsilon = 0.2$ except, for the projected L-BFGS method, $\alpha = 1/3$ was used to ensure that a step-size of 1 would be excepted. The values for σ_1 and σ_2 are as given in the reseting tests for the conjugate gradient method and the L-BFGS method. The minimization routine was terminated when

(i) $x_k^i = 0$ for all $i \in A_k$,

(ii) $\| g_k \|_{I_k} / | I_k | < \varepsilon_{\text{mach}}^{2/3} (1 + | f(u_k) |)$,

(iii) $f(u_k) - f(u_{k-1}) < 10 \varepsilon_{\text{mach}} (1 + | f(u_k) |)$,

(iv) $\| u_k - u_{k-1} \|_\infty < \varepsilon_{\text{mach}}^{1/2} (1 + \| u_k \|_\infty)$,

where the machine precision, $\varepsilon_{\text{mach}} \approx 2.2204e-16$. Note that this is a very demanding termination criterion.

With $C = 0$, there were 171 binding constraints at the solution and these were identified after 8 iterations for projected conjugate gradient, after 7 iterations for the projected L-BFGS methods and after 18 iterations for the projected steepest descent method. With $C = 100$, there were 436 binding constraints at the solution and these were identified after 24, 33 and 241 iterations, respectively, for the three methods.

The number of iterations, function evaluations and gradient evaluations required to reach termination for problem **OCP** with $C = 0$ are given in Table 1. The same information is provided in Table 2 for problem **OCP** with $C = 100$. It is clear that the projected conjugate gradient and the projected L-BFGS methods perform significantly better than the project steepest descent method.

Method	Function Evaluations	Gradient evaluations	Iterations
Conjugate Gradient	89	19	18
L-BFGS	45	14	13
Steepest Descent	143	30	30

Table 1: Work done to solve problem **OCP** with $C = 0$.

Method	Function Evaluations	Gradient evaluations	Iterations
Conjugate Gradient	290	41	40
L-BFGS	247	46	45
Steepest Descent	1891	356	355

Table 2: Work done to solve problem **OCP** with $C = 100$.

The optimal solution, $\hat{u}$, of the discretized problem with $C = 100$ is shown in figure 1.

4. CONCLUSION

We have presented an implementable projected descent algorithm model. Algorithm Model PD, and proved its convergence for any search directions satisfying the conditions in equations (2.5a,b,c). This algorithm model solves a common class of problems involving simple bounds on the decision variables. Furthermore, many problems with simple bounds on the decision variables as well as some additional general constraints can be converted into the form of problem **P** using quadratic penalty functions or augmented Lagrangians. The Algorithm Model PD, when used in conjunction with a conjugate gradient method or the limited-memory BFGS method for determining the unconstrained portion of the search directions, has the advantage of requiring very little storage and work per iteration. Yet, the rate of convergence can be expected to be the same as that of the unconstrained conjugate gradient or limited-memory BFGS methods after a finite number of iterations.

REFERENCES

[Ber.74] D.P. Bertsekas, *Partial Conjugate Gradient Methods for a Class of Optimal Control Problems*, IEEE Trans. Autom. Cntrl, 19 (1974), pp. 209-217.

[Ber.76] D.P. Bertsekas, *On the Goldstein-Levitin-Poljak gradient projection method*, IEEE Trans. Autom. Cntrl. 21 (1976), pp. 174-184.

[Ber.82] D.P. Bertsekas, *Projected Newton Methods for Optimization Problems with Simple Constraints*, SIAM J. Control and Optimization 20 (1982), pp. 221-246.

[CGT.91] A. R. Conn, N. Gould, and P. L. Toint, *A Globally Convergent Augmented Lagrangian Algorithm for Optimization with General Constraints and Simple Bounds*,, SIAM J. Numer. Anal. 28 (1991), pp. 545-572.

[DuB.89] J.C. Dunn and D.P. Bertsekas, *Efficient Dynamic Programming Implementations of Newton's Method for Unconstrained Optimal Control Problems*, J. Optim. Theory and Appl., 63 (1989), pp. 23-38.

[Dun.88] J.C. Dunn, *A Projected Newton Method for Minimization Problems with Nonlinear Inequality Constraints*, Numer. Math. 53 (1988), pp. 377-409.

[GiN.92] J.C. Gilbert and J. Nocedal, *Global Convergence Properties of Conjugate Gradient Methods for Optimization*, SIAM J. Optimization 2 (1992), pp. 21-42.

[Gol.64] A.A. Goldstein, *Convex programming in Hilbert space*, Bull. Amer. Math. Soc. 70 (1964), pp. 709-710.

[GoL.89] G.H. Golub and C.F. Loan, *Matrix Computations*, second edition, Johns Hopkins University Press, 1989.

[LeP.65] E.S. Levitin and B.T. Polyak, *Constrained minimization problems*, USSR Comput. Math. Math. Phys., 6 (1966), pp.1-50.

[LiN.89] D.C. Liu and J. Nocedal, *On the Limited Memory BFGS Method for Large Scale Optimization*, Math. Prog. 45 (1989), pp. 503-528.

[Lue.84] D.G. Luenberger, *Linear and Nonlinear Programming*, second edition, Addison-Wesley Pub. Co., 1984.

[Noc.80] J. Nocedal, *Updating quasi-Newton matrices with limited storage*, Mathematics of Computation 35 (1980), pp. 773-782.

[Pow.76] M.J.D. Powell, *Restart Procedures for the Conjugate Gradient Method*, Math. Prog. 12 (1977), pp. 241-254.

[PSS.74] E. Polak, R. W. H.Sargent and D. J. Sebastian, *On the convergence of sequential minimization algorithms*, J. Optimization Theory and Applications 14 (1974), pp. 439-442.

[QuD.74] V.H. Quintana and E.J. Davison, *Clipping-off Gradient Algorithms to Compute Optimal Controls with Constrained Magnitude*, Int. J. Control, 20 (1974), pp. 243-255.

[ScP.95] A. Schwartz and E. Polak, *Consistent approximations for optimal control problems based on Runge-Kutta integration*, to appear in SIAM J. Optim. Control.

[ShP.78] D.F. Shanno and K.H. Phua, *Matrix conditioning and nonlinear optimization,* Math. Prog. 14 (1978), pp. 149-160.

[ZNB.93] X. Zou, I.M. Navon, M. Berger, K.H, Phua, T. Schlick, and F.X. Dimet, *Numerical experience with limited-memory quasi-Newton and Truncated Newton Methods*, SIAM J. Optim. 3 (1993), pp. 582-608.

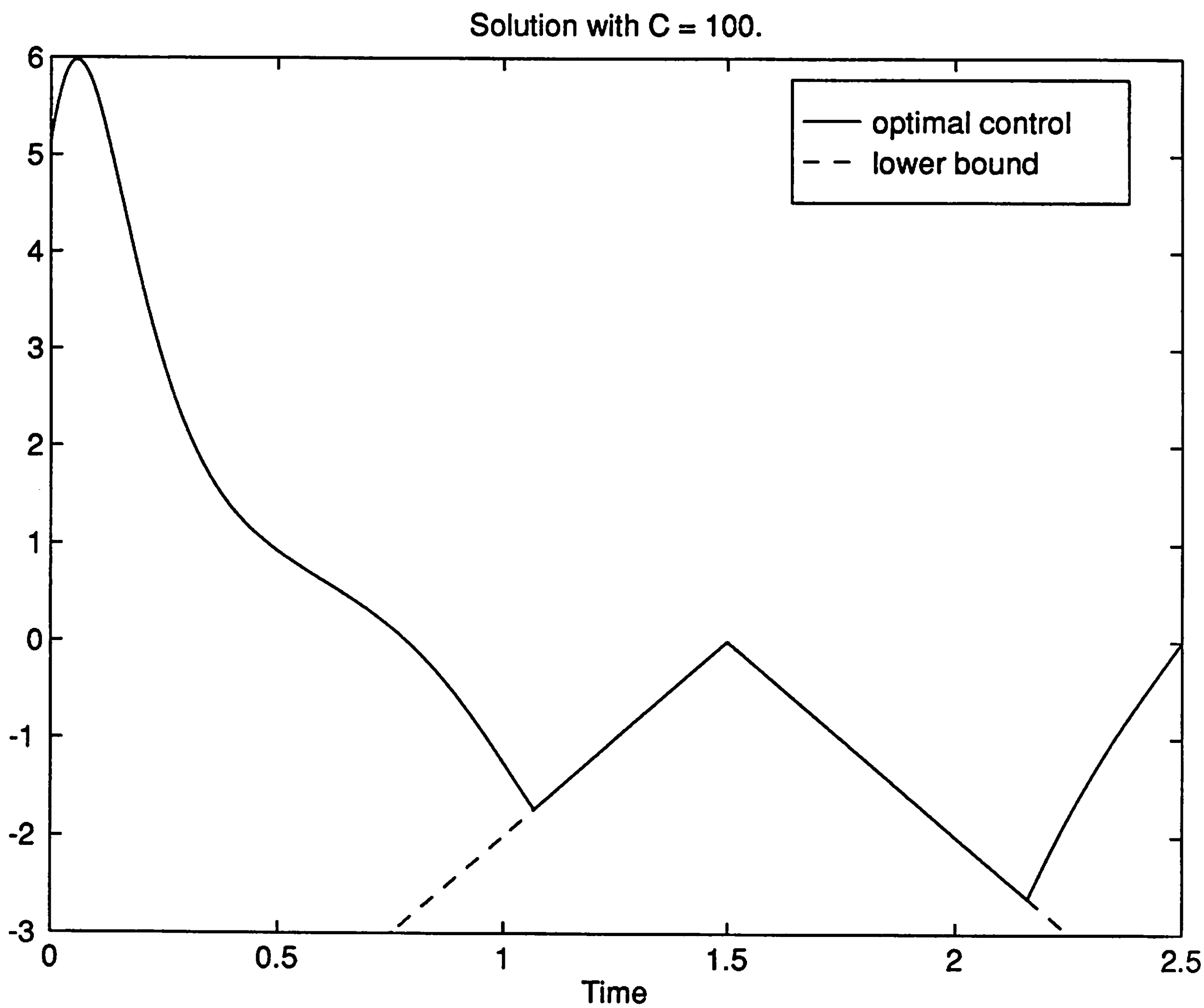
Solution with C = 100.
optimal control
lower bound
6
5
4
3
2
1
0
-1
-2
-3
0
0.5
1
1.5
2
2.5
Time

FIGURE 1