A Class of Projected-Search Methods for Bound-Constrained Optimization

Michael W. Ferry* Philip E. Gill† Elizabeth Wong† Minxin Zhang†

UCSD Center for Computational Mathematics
Technical Report CCoM-20-07
August 10, 2020

Abstract

Projected-search methods for bound-constrained optimization are based on performing a search along a piecewise-linear continuous path obtained by projecting a search direction onto the feasible region. A potential benefit of a projected-search method is that many changes to the active set can be made at the cost of computing a single search direction.

As the objective function is not differentiable along the search path, it is not possible to use a projected-search method with a step that satisfies the Wolfe conditions, which require the directional derivative of the objective function at a point on the path. For this reason, methods based in full or in part on a simple backtracking procedure must be used to give a step that satisfies an "Armijo-like" sufficient decrease condition. As a consequence, conventional projected-search methods are unable to exploit sophisticated safeguarded polynomial interpolation techniques that have been shown to be effective for the unconstrained case.

This paper describes a new framework for the development of a general class of projected-search methods for bound-constrained optimization. At each iteration, a descent direction is computed with respect to a certain *extended* active set. This direction is used to specify a search direction that is used in conjunction with a step length computed by a *quasi*-Wolfe search. The quasi-Wolfe search is designed specifically for use with a piecewise-linear search path and is similar to a conventional Wolfe line search, except that a step is accepted under a wider range of conditions. These conditions take into consideration steps at which the restriction of the objective function on the search path is not differentiable. Standard existence and convergence results associated with a conventional Wolfe line search are extended to the quasi-Wolfe case. In addition, it is shown that under a standard nondegeneracy assumption, any method within the framework will identify the optimal active set in a finite number of iterations.

Computational results are given for a specific projected-search method that uses a limited-memory quasi-Newton approximation of the Hessian. The results show that, in this context, a quasi-Wolfe search is substantially more efficient and reliable than an Armijo-like search based on simple backtracking. Comparisons with a state-of-the-art bound-constrained optimization package are also presented.

Key words. Bound-constrained optimization, projected-search methods, line search methods, projected gradient methods, quasi-Newton methods.

AMS subject classifications. 49M37, 65K05, 90C30, 90C53

*NVIDIA Corporation, Hillsboro, Oregon (`michael@mwferry.com`).

†Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0112 (`pgill@ucsd.edu`, `elwong@ucsd.edu`, `miz151@ucsd.edu`) Research supported in part by National Science Foundation grants DMS-0915220 and DMS-1318480. The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

1. Introduction

This paper describes a new framework for the development of a general class of projected-search methods for the bound-constrained problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\;\; f(x) \quad \text{subject to} \quad x \in \Omega, \tag{BC}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a twice-continuously differentiable function and $\Omega = \{ x \in \mathbb{R}^n : \ell \le x \le u \}$ for vectors of lower and upper bounds such that $\ell \le u$ (with all inequalities defined componentwise). The gradient of f at x is denoted by $\nabla f(x)$. The active set of variables on their bounds at $x \in \Omega$ is denoted by $\mathcal{A}(x)$, i.e., $\mathcal{A}(x) = \{i : x_i = \ell_i \text{ or } x_i = u_i\}$.

Projected-search methods for problem (BC) generate a sequence of feasible iterates $\{x_k\}_{k=0}^{\infty}$ such that $x_{k+1} = \mathbf{proj}_{\Omega}(x_k + \alpha_k p_k)$, where p_k is a descent direction for f at x_k, α_k is a scalar step length, and $\mathbf{proj}_{\Omega}(x)$ is the projection of x onto the feasible region, i.e.,

$$[\,\mathbf{proj}_{\Omega}(x)\,]_i = \begin{cases} \ell_i & \text{if } x_i < \ell_i, \\ u_i & \text{if } x_i > u_i, \\ x_i & \text{otherwise.} \end{cases}$$

The new iterate may be written as $x_{k+1} = x_k(\alpha_k)$, where $x_k(\alpha)$ denotes the vector $x_k(\alpha) = \mathbf{proj}_{\Omega}(x_k + \alpha p_k)$. A potential benefit of a projected-search method is that many changes to the active set can be made at the cost of computing a single search direction. The projected-search methods of Goldstein [17], Levitin and Polyak [22], and Bertsekas [2] are based on using the steepest-descent direction $p_k = -\nabla f(x_k)$. Bertsekas [4] proposes a method based on computing p_k using a Newton-like method. Calamai and Moré [8] consider methods that identify the optimal active set using a projected-search method and then switch to Newton's method. Projected-search methods based on computing p_k using a quasi-Newton method are proposed by Ni and Yuan [26], Kim, Sra and Dhillon [20], Ferry [11], and Ferry et al. [12].

Many methods for unconstrained minimization generate a sequence of iterates $\{x_k\}_{k=0}^{\infty}$ such that x_{k+1} is chosen to give a decrease in f that is at least as good as a fixed fraction η_A ($0 < \eta_A < \frac{1}{2}$) of the decrease in the local affine model $f(x_k) + \nabla f(x_k)^T(x - x_k)$. If x_{k+1} is computed as $x_{k+1} = x_k + \alpha_k p_k$, where p_k is a descent direction for f at x_k and α_k is a positive scalar, then the decrease condition may be written as

$$f(x_k + \alpha_k p_k) \le f(x_k) + \alpha_k \eta_A \nabla f(x_k)^T p_k, \tag{1.1}$$

which is known as the Armijo condition (see, e.g., Armijo [1], Ortega and Rheinboldt [28]). Most Armijo line searches are implemented as a simple backtracking procedure in which an initial step is reduced by a constant factor until the Armijo condition (1.1) is satisfied. Alternatively, backtracking may be used in conjunction with a simple quadratic interpolation scheme using $f(x_k)$, $\nabla f(x_k)^T p_k$ and $f(x_k + \alpha p_k)$ at each trial α (see Dennis and Schnabel [9]).

Many practical methods use an α_k that satisfies an additional condition on the directional derivative $\nabla f(x_k + \alpha_k p_k)^T p_k$. In particular, the strong Wolfe conditions require that α_k satisfies both the Armijo condition (1.1) and

$$|\nabla f(x_k + \alpha_k p_k)^T p_k| \le \eta_W |\nabla f(x_k)^T p_k|, \tag{1.2}$$

where η_W is a preassigned scalar such that $\eta_W \in (\eta_A, 1)$ (see, e.g., Wolfe [30], Moré and Thuente [25], and Gill et al. [16]). The strong Wolfe conditions allow η_W to be chosen to vary the accuracy of the step. If η_A is fixed at a value close to zero (e.g., 10^{-4}), then a value of η_W close to η_A gives a "tighter" or more accurate step with respect to closeness to a critical point of $\nabla f(x_k + \alpha p_k)^T p_k$. A value of η_W close to one results in a "looser" or more approximate step. A Wolfe line search is able to exploit sophisticated safeguarded polynomial interpolation techniques to provide methods that are more reliable and efficient than those based on backtracking (see, e.g., Hager [19] and Morè and Thuente [25]).

In a projected-search method, the function $x_k(\alpha)$ defines a piecewise-linear continuous path, and the function $f\big(x_k(\alpha)\big)$ is not necessarily differentiable along $x_k(\alpha)$. In particular, $f\big(x_k(\alpha)\big)$ may have a "kink" at any $\alpha > 0$ at which $[\, p_k\,]_i \ne 0$ and either $[\, x_k + \alpha p_k\,]_i = \ell_i$ or $[\, x_k + \alpha p_k\,]_i = u_i$. This implies that it is not possible to use a line search based on the conventional Wolfe conditions. Thus, existing projected-search methods are restricted to using a search based on satisfying an Armijo-like condition along the path $x_k(\alpha)$. For the case where $p_k = -\nabla f(x_k)$, a commonly used Armijo-like condition is

$$f\big(x_k(\alpha_k)\big) \le f(x_k) + \eta_A \nabla f(x_k)^T (x_k(\alpha) - x_k), \tag{1.3}$$

proposed by Bertsekas [2] (see also, Calamai and Moré [8]). However, for a general p_k, this may not be a sufficient-decrease condition for a backtracking search as there is no guarantee that the second term on the right-hand side of (1.3) is negative if the path $x_k(\alpha)$ changes direction. An Armijo-like condition that is appropriate for a general descent direction p_k is

$$f\big(x_k(\alpha_k)\big) \le f(x_k) + \alpha_k \eta_A \nabla f(x_k)^T p_k \tag{1.4}$$

(see, e.g., Ni and Yuan [26] and Kim, Sra and Dhillon [20]). Throughout the following discussion, (1.4) is referred to as the *quasi-Armijo* condition. If γ and σ denote fixed parameters such that $\gamma > 0$ and $\sigma \in (0, 1)$, then a quasi-Armijo step has the form $\alpha_k = \gamma \sigma^{t_k}$, where t_k is the smallest nonnegative integer such that the quasi-Armijo condition (1.4) is satisfied. Other sufficient decrease conditions have been proposed. For example, Bertsekas [4] considers an Armijo-like condition based on a combination of (1.3) and (1.4), with the term (1.3) defined with components of a scaled steepest-descent direction.

1.1. Contributions and organization of the paper

Several contributions are made concerning the design and analysis of algorithms for bound-constrained optimization. (i) A new framework is presented for the development of a general class of projected-search methods for the solution of problem (BC). For any method within the proposed framework, a descent direction d_k is computed with respect to a perturbed or extended active set (a similar set is used by Bertsekas [4]). The vector d_k may be computed in many ways, e.g., using an exact or modified Newton-like method or a quasi-Newton method. This direction is used as the basis for the computation of a search direction p_k, and an associated step length α_k such that $f\big(\mathbf{proj}_{\Omega}(x_k + \alpha_k p_k)\big) < f(x_k)$. (ii) Methods within the proposed framework use a *quasi-Wolfe* search, which is specifically designed for use with a piecewise-linear continuous search path. The behavior of the search is similar to that of a conventional Wolfe line search, except that a step is accepted under a wider range of conditions that take into account steps at which f is not differentiable. As in the unconstrained case, the quasi-Wolfe step can be computed using safeguarded polynomial interpolation and the accuracy of the step can be adjusted. (iii) The convergence of any method within the framework is established under assumptions that are typical in the analysis of projected-search methods. In addition, it is shown that if a method converges to a nondegenerate stationary point, then the optimal active set is identified in a finite number of iterations. It follows that once the optimal active set has been identified, any method within the framework will have the same convergence rate as its unconstrained counterpart.

The paper is organized in seven sections. The standard results associated with a conventional Wolfe line search are reviewed in Section 2. Analogous results are established for the quasi-Wolfe search in Section 3. Details of the proposed framework are formulated in Section 4. The convergence properties of a sequence generated by any method within the framework are established in Section 5. Section 6 concerns the numerical performance of the proposed projected-search method when the descent direction is computed using a limited-memory quasi-Newton method. Comparisons with the state-of-the-art package `LBFGS-B` are also presented. The paper concludes with a summary and conclusions.

1.2. Notation

The vectors e and e_j denote, respectively, the column vector of ones and the jth column of the identity matrix I. The dimensions of e, e_j and I are defined by the context. The subscript i is appended to vectors to denote the ith component of that vector, whereas the subscript k is appended to a vector to denote its value during the kth iteration of an algorithm, e.g., x_k represents the value for x during the kth iteration, whereas $[\,x_k\,]_i$ denotes the ith component of the vector x_k. The ith component of the gradient of the scalar-valued function f is denoted by $\nabla_i f(x)$. The vector two-norm or its induced matrix norm are denoted by $\|\cdot\|$.

2. The Wolfe Line Search

A typical Wolfe line search may be viewed as a two-stage process. The first stage involves the determination of an interval containing a Wolfe step, if one exists. The second stage locates a Wolfe step in this interval using safeguarded polynomial interpolation. If the first stage fails, then the objective function is necessarily unbounded below. The key principle that drives the first stage is that certain conditions may be formulated that determine if an interval contains a Wolfe step. Much of the discussion in this section is based on the work of Moré and Sorensen [24], Morè and Thuente [25]. More information may be found in Wolfe [31]. The schematic description of the line-search algorithm given in Algorithm 1 below follows that of Nocedal and Wright [27]. In order to simplify the notation we omit the suffix k and consider the univariate function $\phi(\alpha) = f(x + \alpha p)$ for fixed vectors x and p. With this notation the Wolfe conditions (1.1) and (1.2) may be written in the form

$$\phi(\alpha) \leq \phi(0) + \alpha\eta_A\phi'(0), \quad \text{and} \quad |\phi'(\alpha)| \leq \eta_W|\phi'(0)|.$$

Much of the theory associated with a Wolfe line search is based on the properties of the auxiliary function

$$\omega(\alpha) = \phi(\alpha) - \big(\phi(0) + \alpha\eta_A\phi'(0)\big), \quad \text{with} \quad \omega'(\alpha) = \phi'(\alpha) - \eta_A\phi'(0).$$

Moré and Sorensen [24] show that a minimizer of this function at which ω is negative satisfies the Wolfe conditions. An example of a function ϕ and its associated auxiliary function ω are depicted in Figure 1.

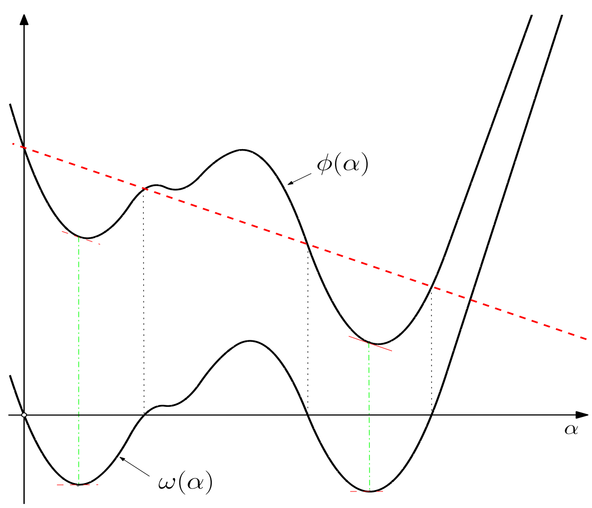

Figure 1: The graph depicts $\phi(\alpha) = f(x + \alpha p)$ as a function of positive α, with the shifted function $\omega(\alpha) = \phi(\alpha) - \big(\phi(0) + \alpha\eta_A\phi'(0)\big)$ superimposed. The dashed line represents the affine function $\phi(0) + \alpha\eta_A\phi'(0)$.

The first stage of a Wolfe line search is motivated by the following proposition.

Proposition 2.1. *Let $\{\alpha_i\}_{i=0}^{\infty}$ be a strictly monotonically increasing sequence with $\alpha_0 = 0$. Let ϕ and ω be continuously differentiable univariate functions such that $\phi'(0) < 0$ and $\omega(\alpha) = \phi(\alpha) - \big(\phi(0) + \alpha\eta_A\phi'(0)\big)$ with $0 < \eta_A < 1$. If there exists a least bounded index j such that at least one of the following conditions is true:*

(a) *α_j is a Wolfe step;*

(b) *$\omega(\alpha_j) \ge \omega(\alpha_{j-1})$; or*

(c) *$\omega'(\alpha_j) \ge 0$,*

then there exists a Wolfe step $\alpha^* \in [\alpha_{j-1}, \alpha_j]$. Collectively, (a)–(c) are called the stage-one conditions*.*

Proof. Observe that α_{j-1} must satisfy none of the conditions (a)–(c), otherwise j would not be the least index. This implies that $\omega(\alpha_{j-1}) < \omega(\alpha_{j-2}) < \cdots < \omega(\alpha_0) = 0$ from (b), and $\omega'(\alpha_{j-1}) < 0$ from (c).

Case 1. If (a) is true, the proposition is true trivially.

Case 2. If (b) is true, let $\bar{\alpha} = \sup\{\alpha \in [\alpha_{j-1},\, \alpha_j] : \omega(\beta) \le 0 \text{ for all } \beta \in [\alpha_{j-1}, \alpha]\}$. If $\bar{\alpha} = \alpha_j$, then $\omega(\bar{\alpha}) = \omega(\alpha_j) \ge \omega(\alpha_{j-1})$; if $\bar{\alpha} < \alpha_j$, then by the continuity of ω, $\omega(\bar{\alpha}) = 0 > \omega(\alpha_{j-1})$. From the mean-value theorem there must exist an $\widehat{\alpha} \in (\alpha_{j-1}, \bar{\alpha})$ such that $\omega'(\widehat{\alpha}) = \big(\omega(\bar{\alpha}) - \omega(\alpha_{j-1})\big)/(\bar{\alpha} - \alpha_{j-1}) > 0$. The function $\omega(\alpha)$ is continuously differentiable with $\omega'(\alpha_{j-1}) < 0$ and $\omega'(\widehat{\alpha}) > 0$. The intermediate-value theorem then implies that there must exist a step $\alpha^* \in [\alpha_{j-1}, \widehat{\alpha}]$ such that $\omega'(\alpha^*) = 0$. As $\omega(\alpha^*) \le 0$, α^* is a Wolfe step.

Case 3. Finally, consider the case where (c) is true. If $\omega(\alpha) < 0$ for all $[\alpha_{j-1}, \alpha_j]$, then, as $\omega'(\alpha_{j-1}) < 0$ and $\omega'(\alpha_j) \ge 0$, the continuity of ω' and the intermediate-value theorem imply that there exists a step $\alpha^* \in [\alpha_{j-1}, \alpha_j]$ such that $\omega'(\alpha^*) = 0$. As $\omega(\alpha^*) < 0$, α^* is a Wolfe step. Otherwise, if there exists some $\alpha \in [\alpha_{j-1}, \alpha_j]$ such that $\omega(\alpha) \ge 0$, let $\bar{\alpha} = \sup\{\alpha \in [\alpha_{j-1},\, \alpha_j] : \omega(\beta) \le 0 \text{ for all } \beta \in [\alpha_{j-1}, \alpha]\}$. The continuity of ω implies that $\omega(\bar{\alpha}) = 0$. The same argument used in Case 2 may be used to show that there must exist an $\widehat{\alpha} \in (\alpha_{j-1}, \bar{\alpha})$ such that $\omega'(\widehat{\alpha}) > 0$ and an $\alpha^* \in [\alpha_{j-1}, \widehat{\alpha}]$ such that $\omega'(\alpha^*) = 0$ with $\omega(\alpha^*) \le 0$. ∎

Note that the converse result is not true, e.g., there may be a Wolfe step in the interval $[0, \alpha_1]$ even though none of the stage-one conditions are satisfied for $j = 1$. The behavior of $\omega(\alpha)$ is unknown at any $\alpha \in (0, \alpha_1)$.

If the first step α_1 is not a Wolfe step, successively larger steps are computed until either one of the stage-one conditions is satisfied or j is such that $\alpha_j = \alpha_{\max}$. In practice, $\alpha_{\max}$ is an upper bound imposed on the step and the search is terminated if the bound is exceeded during the stage-one iterations. If a given α_j does not satisfy the stage-one conditions then $\omega(\alpha_j) < \omega(\alpha_{j-1}) < \cdots < \omega(\alpha_0) = 0$. If the algorithm reaches $\alpha_{j_{\max}} = \alpha_{\max}$ and none of the stage-one conditions have been satisfied, it terminates with $\alpha_{j_{\max}}$, which is an Armijo step with the least computed function value.

Proposition 2.1 implies that if one of the stage-one conditions is satisfied at iteration j, then the interval $[\alpha_{j-1}, \alpha_j]$ must contain a Wolfe step. At this point the line search terminates successfully if the stage-one condition (a) is satisfied, or moves on to the second stage. The computations associated with the second stage are based on the following result.

Proposition 2.2. *Let ϕ and ω be defined as in* Proposition 2.1. *Assume there exist distinct points α_{low} and α_{high} such that*

(a) $\omega(\alpha_{\text{low}}) \le 0$;

(b) $\omega(\alpha_{\text{low}}) \le \omega(\alpha_{\text{high}})$; *and*

(c) $\omega'(\alpha_{\text{low}})(\alpha_{\text{high}} - \alpha_{\text{low}}) < 0$.

Then there exists a Wolfe step $\alpha^* \in \mathcal{I}$, where $\mathcal{I}$ is the interval defined with endpoints α_{low} and α_{high}.

Proof. The proof is similar to that of Proposition 2.1, and is a special case of the proof of Proposition 3.3. ∎

The conditions (a)–(c) of Proposition 2.2 are referred to collectively as the *stage-two conditions*. The subscripts associated with the points α_{low} and α_{high} serve to emphasize the fact that $\omega(\alpha_{\text{low}}) \le \omega(\alpha_{\text{high}})$. It is not necessarily the case that $\alpha_{\text{low}} < \alpha_{\text{high}}$.

Algorithm 1 gives a schematic outline of a Wolfe line search. The calculations required for a Wolfe line search may be organized into two "functions" associated with the stage-one and stage-two conditions. If the first stage finds an interval that contains a Wolfe step, the first-stage function labels the endpoints α_{low} and α_{high} based on relative magnitudes of $\omega(\alpha_{j-1})$ and $\omega(\alpha_j)$, and calls the stage-two function `Stage_Two`$(\alpha_{\text{low}}, \alpha_{\text{high}})$. The second-stage function interpolates the endpoints to calculate a best-guess step, α_{new}, in the interval. The second-stage function is called recursively using α_{new} and an existing endpoint, labeling them so that the stage-two conditions hold for each call. This is repeated until α_{new} is a Wolfe step. In practice, it rarely takes more than one or two interpolations to find a Wolfe step.

A practical implementation of a Wolfe line search is very complex. There are many ways to interpolate to obtain a new point in the second stage. The use of finite precision imposes the need for some sort of safeguarding during interpolation and gives rise to a whole host of issues, including how to handle cases when the function or step length are changing by a value near or less than machine precision. See, e.g., Brent [6], Hager [19], Ghosh and Hager [15], and Moré and Thuente [25] for more details.

3. The Quasi-Wolfe Search

As projected-search methods perform a search on the piecewise continuously differentiable function $f\big(\mathbf{proj}_{\Omega}(x_k + \alpha p_k)\big)$, it is not possible for such methods to use a conventional Wolfe line search. In this section we consider a new step type, called a *quasi-Wolfe step*, that is designed to extend the benefits of a Wolfe line search to projected-search methods.

Algorithm 1 Schematic outline of a Wolfe line search.

```
function WOLFE_LINE_SEARCH(α)
    restriction: α > 0;
    constants: η_A ∈ (0, 1/2), η_W ∈ (η_A, 1), γ_e > 1, α_max ∈ (0, +∞);
    α ← min{α, α_max};   α_old ← 0;
    while α is not a Wolfe step and α ≠ α_max do
        if ω(α) ≥ ω(α_old) then
            α ← Stage_Two(α_old, α); break;
        else if ω′(α) ≥ 0 then
            α ← Stage_Two(α, α_old); break;
        else
            α_old ← α;   α ← min{ γ_e α, α_max };          [Increase α towards α_max]
        end if
    end while
    return α;
end function
function STAGE_TWO(α_low, α_high)
    restriction: ω(α_low) ≤ ω(α_high);
    Choose α_new in the interior of the interval defined by α_low and α_high;
    if α_new is a Wolfe step then
        return α_new;
    else if ω(α_new) ≥ ω(α_low) then
        return Stage_Two(α_low, α_new);
    else if ω′(α_new)(α_high − α_low) < 0 then
        return Stage_Two(α_new, α_high);
    else
        return Stage_Two(α_new, α_low);
    end if
end function
```

3.1. The quasi-Wolfe step

Performing a search on the univariate function

$$\psi_k(\alpha) = f\big(x_k(\alpha)\big) = f\big(\mathbf{proj}_{\Omega}(x_k + \alpha p_k)\big),$$

instead of $\phi_k(\alpha) = f(x_k + \alpha p_k)$, is a substantially more difficult task because ψ_k is only piecewise continuously differentiable, with a finite number of jump discontinuities in the derivative (see Section 3.2 below). Propositions 2.1 and 2.2, established in the preceding section, cannot be used to guarantee a Wolfe step in the nondifferentiable case because they use the mean-value theorem and require the line-search function to be differentiable.

In the following discussion, the suffix k is omitted if the iteration index is not relevant to the discussion. The definition of a quasi-Wolfe step involves the left and right derivatives $\psi'_-(\alpha)$ and $\psi'_+(\alpha)$ of ψ at α, which are defined as

$$\psi'_-(\alpha) = \lim_{\beta\to\alpha^-} \psi'(\beta) \quad \text{and} \quad \psi'_+(\alpha) = \lim_{\beta\to\alpha^+} \psi'(\beta).$$

Definition 3.1. *Let η_A and η_W be constant scalars such that $0 < \eta_A < \eta_W < 1$. A step $\alpha > 0$ is called a* quasi-Wolfe step *if it satisfies the quasi-Armijo condition*

(C$_1$) $\psi(\alpha) \le \psi(0) + \alpha\eta_A\psi'_+(0)$,

and at least one of the following conditions:

(C$_2$) $|\psi'_-(\alpha)| \le \eta_W|\psi'_+(0)|$;

(C$_3$) $|\psi'_+(\alpha)| \le \eta_W|\psi'_+(0)|$;

(C$_4$) *ψ is not differentiable at α and $\psi'_-(\alpha) \le 0 \le \psi'_+(\alpha)$.*

Figure 2 depicts three examples of a kink point satisfying the quasi-Wolfe conditions.

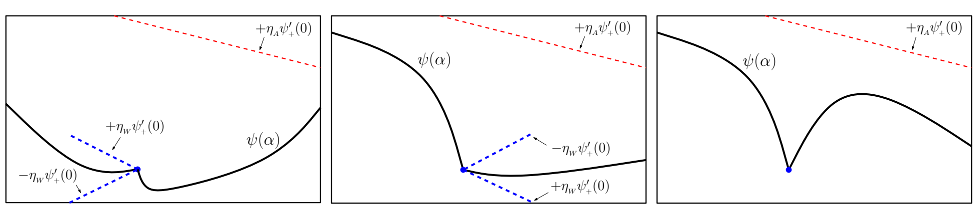

Figure 2: Three examples of a kink point satisfying the quasi-Wolfe conditions. The left, center and right figures depict kink points satisfying conditions **(C$_2$)**, **(C$_3$)** and **(C$_4$)** respectively. The slope of each dashed line is marked.

The properties of the new search are characterized by extending the framework for the differentiable case. In particular, the discussion makes extensive use of the auxiliary function

$$\omega(\alpha) = \psi(\alpha) - \big(\psi(0) + \alpha\eta_A\psi'_+(0)\big), \quad \text{with} \quad \omega'_\pm(\alpha) = \psi'_\pm(\alpha) - \eta_A\psi'_+(0). \tag{3.1}$$

The following lemma is used to establish the propositions below.

Lemma 3.1. *Let* a, $b \in \mathbb{R}$ *be such that* $0 \le a < b$, *and assume that* θ *is a univariate, continuous, piecewise continuously differentiable function with a finite number of jump discontinuities in the derivative.*

(a) *If* $\theta'_+(a) \le 0$ *and* $\theta(a) \le \theta(b)$, *then there exists an* $x \in (a, b)$ *such that*

$$\theta'_-(x) \le 0 \le \theta'_+(x).$$

(b) *If* $\theta'_+(a) < 0$ *and* $\theta'_-(b) > 0$ *then there exists an* $x \in (a, b)$ *such that*

$$\theta'_-(x) \le 0 \le \theta'_+(x).$$

If θ *is differentiable at* x *then the inequalities in the conclusions of parts* (a) *and* (b) *hold as equalities.*

Proof. For part (a), let $a = s_0 < s_1 < s_2 < \cdots < s_t < s_{t+1} = b$, where s_1, s_2, …, s_t represent all the points in (a, b) at which θ is nondifferentiable. First, suppose that $\theta'_+(y) \le 0$ for all $y \in (a, b)$. Then θ is continuously differentiable and nonincreasing within each subinterval $[s_j, s_{j+1}]$ for $j = 0, 1, \ldots, t$. It follows that $\theta(a) \ge \theta(s_1) \ge \cdots \ge \theta(s_t) \ge \theta(b)$. By assumption, this is true only when $\theta(a) = \theta(b)$, which implies that $\theta(a) = \theta(s_1)$. Thus, by Rolle's Theorem, there exists an $x \in (a, s_1) \subset [a, b]$ such that $\theta'(x) = \theta'_\pm(x) = 0$. Now suppose there is a $y \in (a, b)$ such that $\theta'_+(y) > 0$, and let $x = \inf \{ y \in (a, b) : \theta'_+(y) > 0 \}$. Then $x \in (a, b)$, $\theta'_+(x) \ge 0$, and $\theta'_-(x) = \lim_{y \to x^-} \theta'_+(y) \le 0$. For part (b), let $x = \inf \{ y \in (a, b) : \theta'_+(y) > 0 \}$. Then $x \in (a, b)$, $\theta'_+(x) \ge 0$, and $\theta'_-(x) = \lim_{y \to x^-} \theta'_+(x) \le 0$. ∎

The next result establishes conditions on f and Ω that guarantee the existence of a quasi-Wolfe step at each iteration.

Proposition 3.1. *Let* f *be a scalar-valued continuously differentiable function defined on* $\Omega = \{ x \in \mathbb{R}^n : \ell \le x \le u \}$. *Assume that* $x_0 \in \Omega$ *is chosen such that the level set* $\mathcal{L}\big(f(x_0)\big)$ *is bounded, and assume that* $\{p_k\}$ *is a sequence of descent directions. If* η_A *and* η_W *are fixed scalars such that* $0 < \eta_A < \eta_W < 1$, *then at every iteration* k *either there exists an* $\alpha_L^{(k)} > 0$ *and an interval* $(\alpha_L^{(k)}, \alpha_U^{(k)})$ *such that every* $\alpha \in (\alpha_L^{(k)}, \alpha_U^{(k)})$ *is a quasi-Wolfe step, or there exists a quasi-Wolfe step that satisfies the condition* $(\mathbf{C}_4)$.

Proof. We omit the suffix k and write $\psi(\alpha) = f\big(\mathbf{proj}_\Omega(x + \alpha p)\big)$. First, it will be shown that there exists a positive scalar σ such that the function ω of (3.1) satisfies $\omega(\alpha) < 0$ for all $\alpha \in (0, \sigma)$. As $\psi'_+(0) = \nabla f(x)^T p < 0$ and $\eta_A < 1$, it must hold that

$$\omega'_+(0) = (1 - \eta_A)\psi'_+(0) < 0,$$

in which case there must be a scalar σ ($\sigma > 0$) such that $\omega(\alpha) < 0$ for all $\alpha \in (0, \sigma)$. It follows that there exists a $\sigma_1 \in (0, \sigma)$ such that $\omega(\sigma_1) < 0$.

From the compactness of the level set $\mathcal{L}\big(f(x_0)\big)$, $\psi(\alpha)$ is bounded below by some constant ψ_{low}, i.e., $\psi(\alpha) \ge \psi_{\text{low}}$ for all $\alpha \in [0, \infty)$. As $\psi(0) + \alpha\eta_A\psi'_+(0) \to -\infty$ as

$\alpha \to \infty$, there must exist a positive σ_2 such that $\psi(0) + \sigma_2 \eta_A \psi'_+(0) = \psi_{\text{low}}$, and we have

$$\omega(\sigma_2) = \psi(\sigma_2) - \psi(0) - \sigma_2 \eta_A \psi'_+(0) = \psi(\sigma_2) - \psi_{\text{low}} \geq 0.$$

Given scalars σ_1 and σ_2 ($0 \leq \sigma_1 < \sigma_2$) such that $\omega(\sigma_1) < 0$ and $\omega(\sigma_2) \geq 0$, the intermediate-value theorem states that there must exist at least one positive α such that $\omega(\alpha) = 0$. Let β denote the least positive root of $\omega(\alpha) = 0$, then $\omega(\alpha) < 0$ for all $\alpha \in (0, \beta)$. As $\omega(0) = 0$, $\omega(\beta) = 0$, and $\omega'_+(0) < 0$, by Lemma 3.1 (a), there exists an $\xi \in (0, \beta)$ such that

$$\omega'_-(\xi) \leq 0 \leq \omega'_+(\xi), \ \text{ or, equivalently, } \ \psi'_-(\xi) \leq \eta_A \psi'_+(0) \leq \psi'_+(\xi).$$

By construction, $\xi \in (0, \beta)$, which implies that $\omega(\xi) \leq 0$, or equivalently, ξ satisfies the quasi-Armijo condition ($\mathbf{C}_1$). If $\psi'_+(\xi) \leq 0$, then the inequality $\eta_A < \eta_W$ implies that ξ is a quasi-Wolfe step that satisfies the derivative condition ($\mathbf{C}_3$). By the piecewise continuity of $\psi'_+(\alpha)$, there exists an $\alpha_L > 0$ and an interval (α_L, α_U) such that every $\alpha \in (\alpha_L, \alpha_U)$ is a quasi-Wolfe step. Otherwise, if $\psi'_+(\xi) > 0$, then ξ is a quasi-Wolfe step that satisfies the condition ($\mathbf{C}_4$). ∎

The following result is analogous to Proposition 2.1 and motivates the first stage of a quasi-Wolfe search.

Proposition 3.2. *Let $\{\alpha_i\}_{i=0}^{\infty}$ be a strictly monotonically increasing sequence with $\alpha_0 = 0$. Let ψ be a continuous piecewise-differentiable univariate function whose derivative has a finite number of jump discontinuities. Assume that $\psi'_+(0) < 0$ and define $\omega(\alpha) = \psi(\alpha) - \big(\psi(0) + \alpha \eta_A \psi'_+(0)\big)$ with $0 < \eta_A < 1$. If there exists a least bounded index j such that at least one of the following "stage-one" conditions is true:*

- **(a)** *α_j is a quasi-Wolfe step;*
- **(b)** *$\omega(\alpha_j) \geq \omega(\alpha_{j-1})$; or*
- **(c)** *$\omega'_-(\alpha_j) \geq 0$,*

then there exists a quasi-Wolfe step $\alpha^* \in [\alpha_{j-1}, \alpha_j]$.

Proof. Observe that α_{j-1} must satisfy none of the conditions (a)–(c), otherwise j would not be the least index. This implies that $\omega(\alpha_{j-1}) < \omega(\alpha_{j-2}) < \cdots < \omega(\alpha_0) = 0$ from (b), and $\omega'_-(\alpha_{j-1}) < 0$ from (c).

The first step is to show that

$$\omega'_+(\alpha_{j-1}) < 0. \tag{3.2}$$

If $\omega'(\alpha_{j-1})$ exists, then $\omega'_+(\alpha_{j-1}) = \omega'_-(\alpha_{j-1}) < 0$. If $\omega'(\alpha_{j-1})$ does not exist, then (c) implies that $\omega'_-(\alpha_{j-1}) = \psi'_-(\alpha_{j-1}) - \eta_A \psi'_+(0) < 0$, in which case $\psi'_-(\alpha_{j-1}) < 0$ because $\psi'_+(0) < 0$ by assumption. As ($\mathbf{C}_4$) cannot hold at α_{j-1}, it follows that $\psi'_+(\alpha_{j-1}) < 0$. Now, if ($\mathbf{C}_3$) does not hold at α_{j-1} then $\psi'_+(\alpha_{j-1}) < \eta_W \psi'_+(0) < \eta_A \psi'_+(0)$. Thus, $\omega'_+(\alpha_{j-1}) = \psi'_+(\alpha_{j-1}) - \eta_A \psi'_+(0) < 0$. The inequality (3.2) is used in the proofs that follow.

Case 1. If (a) is true, the proposition holds trivially.

Case 2. If (b) is true, let $\bar{\alpha} = \sup\{\alpha \in [\alpha_{j-1},\, \alpha_j] : \omega(\beta) \le 0 \text{ for all } \beta \in [\alpha_{j-1}, \alpha]\}$. If $\bar{\alpha} = \alpha_j$, then $\omega(\bar{\alpha}) = \omega(\alpha_j) \ge \omega(\alpha_{j-1})$; if $\bar{\alpha} < \alpha_j$, then by the continuity of ω, $\omega(\bar{\alpha}) = 0 > \omega(\alpha_{j-1})$. In either case, as $\omega'_+(\alpha_{j-1}) < 0$ by (3.2), part (a) of Lemma 3.1 implies that there exists an $\alpha^* \in [\alpha_{j-1}, \bar{\alpha}]$ such that

$$\omega'_-(\alpha^*) \le 0 \le \omega'_+(\alpha^*).$$

This implies that

$$\psi'_-(\alpha^*) \le \eta_A \psi'_+(0) \le \psi'_+(\alpha^*).$$

From the definition of $\bar{\alpha}$, α^* satisfies the quasi-Armijo condition ($\mathbf{C}_1$). As $\psi'_-(\alpha^*) < 0$, if $\psi'_+(\alpha^*) \ge 0$, then α^* is a quasi-Wolfe step by ($\mathbf{C}_4$). Alternatively, if $\psi'_+(\alpha^*) < 0$, then

$$\eta_W \psi'_+(0) < \eta_A \psi'_+(0) \le \psi'_+(\alpha^*) < 0,$$

and again, α^* is a quasi-Wolfe step by ($\mathbf{C}_3$).

Case 3. Finally, consider the case where (c) is true, i.e., $\omega'_-(\alpha_j) \ge 0$. By (3.2), $\omega'_+(\alpha_{j-1}) < 0$. If $\omega(\alpha) \le 0$ for all $\alpha \in [\alpha_{j-1}, \alpha_j]$, then either $\omega'_-(\alpha_j) = 0$ such that α_j is a quasi-Wolfe step, or part (b) of Lemma 3.1 establishes the existence of a step $\alpha^* \in (\alpha_{j-1}, \alpha_j)$ such that

$$\omega'_-(\alpha^*) \le 0 \le \omega'_+(\alpha^*),$$

and α^* satisfies the quasi-Armijo condition ($\mathbf{C}_1$). Otherwise, let $\bar{\alpha} = \sup\{\alpha \in [\alpha_{j-1}, \alpha_j] : \omega(\beta) \le 0 \text{ for all } \beta \in [\alpha_{j-1}, \alpha]\}$. By the continuity of ω, $\omega(\bar{\alpha}) = 0 > \omega(\alpha_{j-1})$. It follows from part (a) of Lemma 3.1 that there exists a step $\alpha^* \in [\alpha_{j-1}, \bar{\alpha}]$ such that

$$\omega'_-(\alpha^*) \le 0 \le \omega'_+(\alpha^*),$$

and α^* satisfies the quasi-Armijo condition ($\mathbf{C}_1$). The same argument used for the preceding case shows that α^* is a quasi-Wolfe step. ∎

The second stage of a quasi-Wolfe search is based on the following proposition.

Proposition 3.3. *Let ψ and ω be defined as in* Proposition 3.2. *Assume there exist distinct points α_{low} and α_{high} such that*

(a) $\omega(\alpha_{\text{low}}) \le 0$;

(b) $\omega(\alpha_{\text{low}}) \le \omega(\alpha_{\text{high}})$; *and*

(c) $\omega'_+(\alpha_{\text{low}}) < 0$ *if* $\alpha_{\text{low}} < \alpha_{\text{high}}$ *or* $\omega'_-(\alpha_{\text{low}}) > 0$ *if* $\alpha_{\text{low}} > \alpha_{\text{high}}$,

then there exists a quasi-Wolfe step $\alpha^* \in \mathcal{I}$, where $\mathcal{I}$ is the interval defined with endpoints α_{low} and α_{high}.

Proof. First, consider the case where $\alpha_{\text{low}} < \alpha_{\text{high}}$. Let $\bar{\alpha} = \sup\big\{\, \alpha \in \mathcal{I} : \omega(\beta) \le 0$ for all $\beta \in [\alpha_{\text{low}}, \alpha] \,\big\}$. By the continuity of ω, $\omega(\bar{\alpha}) = 0 \ge \omega(\alpha_{\text{low}})$. It follows from part (a) of Lemma 3.1 that there exists a step $\alpha^* \in [\alpha_{\text{low}}, \bar{\alpha}]$ such that $\omega(\alpha^*) \le 0$ and

$$\omega'_-(\alpha^*) \le 0 \le \omega'_+(\alpha^*).$$

The same argument used in Proposition 3.2 shows that α^* is a quasi-Wolfe step.

For the case $\alpha_{\text{low}} > \alpha_{\text{high}}$, let $\widetilde{\omega}(\alpha) = \omega(\alpha_{\text{low}} + \alpha_{\text{high}} - \alpha)$. Then $\widetilde{\omega}(\alpha_{\text{high}}) = \omega(\alpha_{\text{low}}) \le 0$, and $\widetilde{\omega}'_+(\alpha_{\text{high}}) = -\omega'_-(\alpha_{\text{low}}) < 0$. Let $\bar{\alpha} = \sup\big\{\, \alpha \in \mathcal{I} : \widetilde{\omega}(\beta) \le 0$ for all $\beta \in [\alpha_{\text{high}}, \alpha] \,\big\}$. The continuity of $\widetilde{\omega}$ implies that $\widetilde{\omega}(\bar{\alpha}) = 0 \ge \widetilde{\omega}(\alpha_{\text{high}})$. It follows from part (a) of Lemma 3.1 that there exists a step $\beta^* \in [\alpha_{\text{high}}, \bar{\alpha}]$ such that $\widetilde{\omega}(\beta^*) \le 0$ and

$$\widetilde{\omega}'_-(\beta^*) \le 0 \le \widetilde{\omega}'_+(\beta^*).$$

Let $\alpha^* = \alpha_{\text{low}} + \alpha_{\text{high}} - \beta^*$, then $\alpha^* \in \mathcal{I}$, $\omega(\alpha^*) \le 0$ and

$$\omega'_-(\alpha^*) = -\widetilde{\omega}'_+(\beta^*) \le 0 \le -\widetilde{\omega}'_-(\beta^*) = \omega'_+(\alpha^*).$$

It follows that α^* is a quasi-Wolfe step. ∎

Although the implementation of a quasi-Wolfe search is similar to that of a Wolfe line search, there are a number of crucial practical issues associated with the potential nondifferentiability of the line-search function. These issues include the definition of the derivatives of the line-search function and the computation of a new estimate of a quasi-Wolfe step.

3.2. Derivatives of the search function

The purpose of this section is to establish expressions for the left- and right-derivatives of the search function $\psi(\alpha) = f\big(x(\alpha)\big)$, where $x(\alpha)$ is the vector $\mathbf{proj}_{\Omega}(x + \alpha p)$ with components

$$x_i(\alpha) = \begin{cases} \ell_i & \text{if } \; x_i + \alpha p_i < \ell_i, \\ u_i & \text{if } \; x_i + \alpha p_i > u_i, \\ x_i + \alpha p_i & \text{if } \; \ell_i \le x_i + \alpha p_i \le u_i. \end{cases}$$

First, we consider the derivatives of $x(\alpha)$. Under the assumptions that x is feasible and α is positive, it must hold that if $x_i + \alpha p_i < \ell_i$ then $p_i < 0$, and if $x_i + \alpha p_i > u_i$, then $p_i > 0$. This implies that the right derivative of $x(\alpha)$ with respect to α is given by

$$[x'_+(\alpha)]_i = \begin{cases} 0 & \text{if } \; x_i(\alpha) = \ell_i \; \text{ and } \; p_i < 0, \\ 0 & \text{if } \; x_i(\alpha) = u_i \; \text{ and } \; p_i > 0, \\ p_i & \text{otherwise.} \end{cases}$$

The vector $x'_+(\alpha)$ may be expressed in terms of $P_x(p)$, the *projected direction of p at x*, which is defined as

$$[\,P_x(p)\,]_i = \begin{cases} 0 & \text{if } \; x_i = \ell_i \; \text{ and } \; p_i < 0, \\ 0 & \text{if } \; x_i = u_i \; \text{ and } \; p_i > 0, \\ p_i & \text{otherwise.} \end{cases}$$

The vector $P_x(p)$ represents the projection of p onto the closure of the set of feasible directions at $x(\alpha)$. If $x(\alpha)$ is differentiable at a point α, then

$$x'(\alpha) = x'_+(\alpha) = P_{x(\alpha)}(p). \tag{3.3}$$

If $x(\alpha)$ is not differentiable at α then there must be at least one index i such that

$$(x_i + \alpha p_i = \ell_i \;\text{ and }\; p_i < 0) \quad \text{or} \quad (x_i + \alpha p_i = u_i \;\text{ and }\; p_i > 0).$$

An α satisfying one of these conditions is called a *kink step with respect to* i and it also must hold that $x'_+(\alpha) \neq x'_-(\alpha)$. In order to compute the left derivative $x'_-(\alpha)$, consider the values of $x'(\beta)$ as β approaches α from below. If α is a kink step with respect to i then $x_i + \beta p_i$ is feasible for all β sufficiently close to α and it follows from (3.3) that $x'_i(\beta) = p_i$. If this value is combined with the components of $x'_i(\beta)$ associated with the differentiable case, we obtain

$$x'_-(\alpha) = P^-_{x(\alpha)}(p),$$

where

$$[\, P^-_{x(\alpha)}(p)\,]_i = \begin{cases} p_i & \text{if } \alpha \text{ is a kink step with respect to } i, \\ [\, P_{x(\alpha)}(p)\,]_i & \text{otherwise.} \end{cases}$$

Next we consider the derivatives of the search function $\psi(\alpha)$. If $\psi(\alpha)$ is differentiable at α, then the chain rule gives

$$\psi'(\alpha) = \frac{d}{d\alpha} f\big(x(\alpha)\big) = \nabla f\big(x(\alpha)\big)^T \frac{d}{d\alpha} x(\alpha) = \nabla f\big(x(\alpha)\big)^T x'(\alpha).$$

Using this expression with the expression (3.3) for $x'(\alpha)$ gives

$$\psi'(\alpha) = \nabla f\big(x(\alpha)\big)^T P_{x(\alpha)}(p).$$

If $\psi(\alpha)$ is not differentiable at α, then α is a kink step and $\psi'_-(\alpha) \neq \psi'_+(\alpha)$. For any α, $\lim_{\beta\to\alpha^+} x'(\beta) = x'_+(\alpha)$, and $\lim_{\beta\to\alpha^-} x'(\beta) = x_-(\alpha)$. It follows that the right- and left-derivatives of $\psi_+(\alpha)$ with respect to α are given by

$$\psi'_+(\alpha) = \nabla f\big(x(\alpha)\big)^T x'_+(\alpha) = \nabla f\big(x(\alpha)\big)^T P_{x(\alpha)}(p),$$

and

$$\psi'_-(\alpha) = \nabla f\big(x(\alpha)\big)^T x_-(\alpha) = \nabla f\big(x(\alpha)\big)^T P^-_{x(\alpha)}(p).$$

These expressions imply that there is a jump of magnitude $\big|p_i \nabla_i f\big(x(\alpha)\big)\big|$ in the derivative of ψ at a kink step with respect to i.

3.3. Computing a quasi-Wolfe step

As in the Wolfe line search discussed in Section 2, a quasi-Wolfe search may be regarded as having two stages. The first stage begins with an initial step length α_0 and continues with steps of increasing magnitude until one of three things happens: an acceptable step length is found; an interval that contains a quasi-Wolfe step is found; or the step is considered to be unbounded. In practice, the search is terminated if the computed step length exceeds a preassigned upper bound $\alpha_{\max}$ during the first-stage iterations. If the search terminates at $\alpha_{\max}$ without finding an interval containing a quasi-Wolfe step, then every step computed up to that point satisfies the quasi-Armijo condition.

If the first stage terminates with a bounded step, the second stage repeatedly calls a function `Stage_Two`$(\alpha_{\text{low}}, \alpha_{\text{high}})$, where

(a) the interval bounded by α_{low} and α_{high} contains a quasi-Wolfe step;

(b) among all the step lengths generated so far, α_{low} gives the least value of ω;

(c) α_{high} is chosen so that $\omega'_{+}(\alpha_{\text{low}}) < 0$ if $\alpha_{\text{low}} < \alpha_{\text{high}}$, or $\omega'_{-}(\alpha_{\text{low}}) > 0$ if $\alpha_{\text{low}} > \alpha_{\text{high}}$.

It must be emphasized that in practice, the stage-two calculations are not implemented as a recursive procedure. The recursive structure depicted in Algorithm 1 is illustrative and reflects the fact that the intervals defined by α_{low} and α_{high} form a *nested sequence*. If $\mathcal{I}_0$ is the interval resulting from stage-one, the computations of stage-two generate a sequence of intervals $\{\mathcal{I}_j\}$ and a sequence of points $\{\alpha_{\text{low}}^{(j)}\}$ such that $\alpha^{(j)} \in \mathcal{I}_j$, each $\mathcal{I}_j$ contains a quasi-Wolfe step, and $\mathcal{I}_j \subset \mathcal{I}_{j-1}$. The intervals $\mathcal{I}_j$ form a nested sequence of "intervals of uncertainty". Algorithm 2 gives a schematic outline of a quasi-Wolfe search.

A major difference between a Wolfe and a quasi-Wolfe search concerns how interpolation is used to find new steps in the second stage. Each time `Stage_Two`$(\alpha_{\text{low}}, \alpha_{\text{high}})$ is invoked, a new trial step α_{new} is generated. In the differentiable case, α_{new} is usually obtained by polynomial interpolation using the value of ϕ and its derivatives at α_{low} and α_{high}. If the univariate search function is only piecewise differentiable, there may be kink points between α_{low} and α_{high}, in which case a conventional interpolation approach may not provide a good estimate of a quasi-Wolfe step. One strategy to speed convergence in this situation is to search for the kink step (if it exists) between α_{low} and α_{high} that is closest to α_{low}. This approach is justified by the following argument. If a new point α_{new} is not a quasi-Wolfe step, then based on Proposition 3.3, the end points α_{low} and α_{high} are updated to α_{low} and α_{new} in two cases:

Case (1). $\omega(\alpha_{\text{new}}) \geq \omega(\alpha_{\text{low}})$;

Case (2). $\omega'_{+}(\alpha_{\text{new}}) < 0$ if $\alpha_{\text{high}} < \alpha_{\text{low}}$, or $\omega'_{-}(\alpha_{\text{new}}) > 0$ if $\alpha_{\text{high}} > \alpha_{\text{low}}$.

In these cases, the new interval bounded by α_{low} and α_{new} will not contain a kink step. In the remaining case:

Algorithm 2 Schematic outline of a quasi-Wolfe search.

function QUASI_WOLFE_SEARCH(α)
    restriction: $\alpha > 0$;
    constants: $\eta_A \in (0, \frac{1}{2})$, $\eta_W \in (\eta_A, 1)$, $\gamma_e > 1$, $\alpha_{\max} \in (0, +\infty)$;
    $\alpha \leftarrow \min\{\alpha, \alpha_{\max}\}$; $\quad \alpha_{\text{old}} \leftarrow 0$;
    while α is not a quasi-Wolfe step `and` $\alpha \neq \alpha_{\max}$ **do**
        if $\omega(\alpha) \geq \omega(\alpha_{\text{old}})$ **then**
            $\alpha \leftarrow$ `Stage_Two`$(\alpha_{\text{old}}, \alpha)$; break;
        else if $\omega'_-(\alpha) \geq 0$ **then**
            $\alpha \leftarrow$ `Stage_Two`$(\alpha, \alpha_{\text{old}})$; break;
        else
            $\alpha_{\text{old}} \leftarrow \alpha$; $\quad \alpha \leftarrow \min\left\{ \gamma_e \alpha, \alpha_{\max} \right\}$;   [Increase α towards $\alpha_{\max}$]
        end if
    end while
    return α;
end function

function STAGE_TWO($\alpha_{\text{low}}, \alpha_{\text{high}}$)
    restriction: $\omega(\alpha_{\text{low}}) \leq \omega(\alpha_{\text{high}})$;
    Choose α_{new} in the interior of the interval defined by α_{low} and α_{high};
    if α_{new} is a quasi-Wolfe step **then**
        return α_{new};
    else if $\omega(\alpha_{\text{new}}) \geq \omega(\alpha_{\text{low}})$ **then**
        return `Stage_Two`$(\alpha_{\text{low}}, \alpha_{\text{new}})$;
    else if $\omega'_+(\alpha_{\text{new}}) < 0$ `and` $\alpha_{\text{low}} < \alpha_{\text{high}}$ **then**
        return `Stage_Two`$(\alpha_{\text{new}}, \alpha_{\text{high}})$;
    else if $\omega'_-(\alpha_{\text{new}}) > 0$ `and` $\alpha_{\text{low}} > \alpha_{\text{high}}$ **then**
        return `Stage_Two`$(\alpha_{\text{new}}, \alpha_{\text{high}})$;
    else
        return `Stage_Two`$(\alpha_{\text{new}}, \alpha_{\text{low}})$;
    end if
end function

Case (3). $\omega'_+(\alpha_{\rm new}) \ge 0$ if $\alpha_{\rm high} < \alpha_{\rm low}$, or $\omega'_-(\alpha_{\rm new}) \le 0$ if $\alpha_{\rm high} > \alpha_{\rm low}$,

the new interval will be bounded by $\alpha_{\rm high}$ and $\alpha_{\rm new}$, but may contain kink points. However, the new interval must contain at least one fewer kink point.

The search for the kink points proceeds as follows. The first time the function `Stage_Two`$(\alpha_{\rm low},\ \alpha_{\rm high})$ is invoked, the kink steps are computed in $O(n)$ floating-point operations (flops) from

$$\kappa_i = \begin{cases} (u_i - x_i)/p_i & \text{if } \ p_i > 0, \\ (\ell_i - x_i)/p_i & \text{if } \ p_i < 0, \\ \infty & \text{if } \ p_i = 0. \end{cases}$$

As the interval bounded by $\alpha_{\rm low}$ and $\alpha_{\rm high}$ contains a quasi-Wolfe step, only the kink steps within that interval need be stored. These steps are then sorted in decreasing order within $O(n \log n)$ flops using a heapsort algorithm (see, e.g., Williams [29], Knuth [21, Section 5.2.3]). The kink step closest to $\alpha_{\rm low}$, say κ_1^*, is either the smallest or the largest kink step within the interval of uncertainty, depending on whether $\alpha_{\rm low}$ is smaller or greater than $\alpha_{\rm high}$. Once κ_1^* has been found, the search for κ_l^* $(l > 1)$ is made towards $\alpha_{\rm low}$ starting at the kink step κ_{l-1}^* from the preceding iteration. To prevent the iterations from lingering at **Case (3)** for too long, an upper limit is imposed on the number of consecutive kink steps as trial steps. If this limit is reached, a new trial step is generated by bisection.

Once all the kinks in the interval of uncertainty have been eliminated, conventional polynomial interpolation may be used to generate a new step length. However, some care is necessary to choose the appropriate left or right derivative for use in the interpolation (see Section 3.2).

If there is just one kink step in the interval of uncertainty, $\alpha_{\rm new}$ is set to be that kink step. As the number of kink steps in an interval increases, it becomes more difficult to strike a balance between making effective use of the knowledge they exist and efficiency; for example, if an interval contains 10^6 kink steps, it is not practical to jump to the middle one and repeat on each subinterval.

4. Framework for a Class of Projected-Search Methods

This section concerns the formulation of a framework for the development of a general class of projected-search methods for problem (BC). Given an initial $x_0 \in \mathit{\Omega}$, the sequence of iterates $\{x_k\}$ satisfies $x_{k+1} = x_k(\alpha_k) = \mathbf{proj}_{\mathit{\Omega}}(x_k + \alpha_k p_k)$, where α_k is a quasi-Wolfe step, and p_k is a descent direction for f at x_k. The search direction p_k is based on the components of a feasible descent direction d_k computed in terms of a *working set* of indices at x_k such that

$$\begin{aligned} \mathcal{W}_k = \big\{\, i : [x_k]_i \le \ell_i + \epsilon_k \ \text{ and } \ \nabla_{\!i} f(x) > 0 \ \text{ or} \\ [x_k]_i \ge u_i - \epsilon_k \ \text{ and } \ \nabla_{\!i} f(x) < 0 \,\big\}, \end{aligned} \tag{4.1}$$

where ϵ_0 is a fixed positive parameter ϵ, and $\epsilon_k = \min\big\{\epsilon, \big\|\mathit{\Pi}_{k-1}^T \nabla f(x_{k-1})\big\|\big\}$ for $k \ge 1$, with $\mathit{\Pi}_{k-1}$ the matrix of columns of the identity matrix of order n associated with the indices in the complement of $\mathcal{W}_{k-1}$ in $\{1, 2, \dots, n\}$. The matrix $\mathit{\Pi}_{k-1}\mathit{\Pi}_{k-1}^T$ represents the projection $P_{\mathcal{W}_{k-1}}$ with respect to the set $\mathcal{W}_{k-1}$, i.e., for any $d \in \mathbb{R}^n$ it holds that $\mathit{\Pi}_{k-1}\mathit{\Pi}_{k-1}^T d = P_{\mathcal{W}_{k-1}}(d)$, with

$$[\, P_{\mathcal{W}_{k-1}}(d)\,]_i = \begin{cases} 0 & \text{if } \ i \in \mathcal{W}_{k-1}, \\ d_i & \text{if } \ i \notin \mathcal{W}_{k-1}. \end{cases}$$

The search direction p_k is defined in terms of any direction d_k such that $d_k = \mathit{\Pi}_k \mathit{\Pi}_k^T d_k$, and $\nabla f(x_k)^T d_k < 0$. Once d_k is determined, the components of d_k are modified if necessary to give a search direction p_k such that $[\, p_k\,]_i = \max\{[\, d_k\,]_i, 0\}$ if $[\, x_k\,]_i \le \ell_i + \epsilon_k$ and $[\, p_k\,]_i = \min\{[\, d_k\,]_i, 0\}$ if $[\, x_k\,]_i \ge u_i - \epsilon_k$. This additional step guarantees convergence in the situation where iterates approach a boundary point from the interior of the feasible region—a phenomenon known as zigzagging or jamming (see Bertsekas [3]). The vector p_k satisfies $p_k = \mathit{\Pi}_k \mathit{\Pi}_k^T p_k$, and retains the descent property of d_k. For example, if $[\, d_k\,]_i \ne 0$ and $[\, x_k\,]_i \le \ell_i + \epsilon_k$, then the definition of $\mathcal{W}_k$ implies that $\nabla_i f(x_k) \le 0$. If $[\, p_k\,]_i > 0$ then $[\, p_k\,]_i = [\, d_k\,]_i$. Otherwise, $[\, d_k\,]_i < 0$ with $\nabla_i f(x_k)[\, d_k\,]_i \ge 0$, and setting $[\, p_k\,]_i = 0$ makes the directional derivative more negative.

The working set at x_k is a subset of the *extended active set*, which is defined as

$$\mathcal{A}_{\epsilon_k}(x_k) = \{\, i : [\, x_k\,]_i \le \ell_i + \epsilon_k \ \text{ or } \ [\, x_k\,]_i \ge u_i - \epsilon_k \,\}.$$

It is shown in Section 5 that, under certain conditions, $\{\epsilon_k\} \to 0$, and $\mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x_k)$ for k sufficiently large, which would imply that $p_k = d_k$ for k sufficiently large.

A general projected-search method based on the proposed framework is summarized in Algorithm 3. There are various choices for the direction d_k. For example,

Algorithm 3 Framework for a class of projected-search methods

constant: $\epsilon > 0$;
Choose $x_0 \in \mathit{\Omega}$;
Let $\epsilon_0 = \epsilon$; $\quad k = 0$;
while not converged **do**
  Determine the working set $\mathcal{W}_k$ (4.1);
  Compute a feasible descent direction d_k at x_k such that $[\, d_k\,]_i = 0$ if $i \in \mathcal{W}_k$;
  Modify d_k to give a search direction p_k:

$$[\, p_k\,]_i = \begin{cases} \max\{[\, d_k\,]_i, 0\} & \text{if } [\, x_k\,]_i \le \ell_i + \epsilon_k \\ \min\{[\, d_k\,]_i, 0\} & \text{if } [\, x_k\,]_i \ge u_i - \epsilon_k \\ [\, d_k\,]_i & \text{otherwise;} \end{cases}$$

  Compute a quasi-Wolfe step α_k; $\quad x_{k+1} = \mathbf{proj}_{\mathit{\Omega}}(x_k + \alpha_k p_k)$;
  $\epsilon_{k+1} = \min\{\epsilon, \|\mathit{\Pi}_k^T \nabla f(x_k)\|\}$;
  $k \leftarrow k + 1$;
end while

if $d_k = -\mathit{\Pi}_k \mathit{\Pi}_k^T \nabla f(x_k)$, then the method is a variant of projected gradient. Other choices include computing d_k as the unconstrained minimizer of a primal-dual barrier function (see Ferry et al. [13]), and as the solution of the subproblem

$$\underset{d}{\text{minimize}}\ \nabla f(x_k)^T d + \tfrac{1}{2} d^T H_k d \quad \text{subject to } d_i = 0 \text{ for all } i \in \mathcal{W}_k, \tag{4.2}$$

where H_k is a positive-definite approximation of the Hessian $\nabla^2 f(x_k)$. For the numerical experiments presented in Section 6, d_k was the solution of (4.2) with H_k chosen as a positive-definite limited-memory BFGS approximation of $\nabla^2 f(x_k)$ (see Ferry et al. [12]).

5. Convergence Analysis

In this section we consider the convergence properties of the class of projected-search methods described in Section 4. As an introduction, we first consider the convergence of a method with a quasi-Armijo search, which gives a step satisfying the condition (1.4).

Theorem 5.1. (Projected search using a quasi-Armijo search) *Let f be a scalar-valued continuously differentiable function defined on $\Omega = \{ x \in \mathbb{R}^n : \ell \le x \le u \}$. Assume that $x_0 \in \Omega$ is chosen such that the level set $\mathcal{L}\big(f(x_0)\big)$ is bounded, and $\{x_k\}$ is defined by $x_{k+1} = x_k(\alpha_k)$, where α_k is a quasi-Armijo step. For an arbitrarily fixed $\epsilon > 0$, define $\epsilon_0 = \epsilon$, and*

$$\epsilon_k = \min\left\{\epsilon, \left\|\mathit{\Pi}_{k-1}^T \nabla f(x_{k-1})\right\|\right\}$$

for $k \ge 1$, where each $\mathit{\Pi}_k$ is a matrix with orthonormal columns that spans the set of projected directions with respect to the working set $\mathcal{W}_k$. If $\{p_k\}$ is a sequence of descent directions with $\|p_k\| \le \theta$ for some constant θ independent of k, $\mathit{\Pi}_k \mathit{\Pi}_k^T p_k = p_k$ for all k, and the components of p_k satisfy $[\,p_k\,]_i \ge 0$ if $[\,x_k\,]_i \le \ell_i + \epsilon_k$, and $[\,p_k\,]_i \le 0$ if $[\,x_k\,]_i \ge u_i - \epsilon_k$, then

$$\lim_{k\to\infty} \left|\nabla f(x_k)^T p_k\right| = 0.$$

Proof. First, we show that $\lim_{k\to\infty} \left|\nabla f(x_k)^T p_k\right| = 0$ if $\liminf_{k\to\infty} \left\|\mathit{\Pi}_k^T \nabla f(x_k)\right\| \ne 0$. Observe that the quasi-Armijo condition (1.4) implies that $\{f(x_k)\}$ is a strictly decreasing sequence. As the set $\mathcal{L}\big(f(x_0)\big)$ is closed and bounded, it follows that $\{f(x_k)\}$ converges, with

$$0 = \lim_{k\to\infty} f(x_k) - f(x_{k+1}) \ge \lim_{k\to\infty} \alpha_k \eta_A |\nabla f(x_k)^T p_k| = 0.$$

The proof is by contradiction. Suppose that $|\nabla f(x_k)^T p_k| \not\to 0$ as $k \to \infty$, then there must exist some $\bar\epsilon > 0$ such that $|\nabla f(x_k)^T p_k| > \bar\epsilon$ infinitely often. Let $\mathcal{G} = \{ k : |\nabla f(x_k)^T p_k| > \bar\epsilon \}$, then it must be that $\alpha_k \to 0$ for $k \in \mathcal{G}$. For all $k \in \mathcal{G}$, define the step $\beta_k = \alpha_k/\sigma$. The hypothesis that $\liminf_{k\to\infty} \left\|\mathit{\Pi}_k^T \nabla f(x_k)\right\| \ne 0$ implies $\liminf_{k\to\infty} \epsilon_k > 0$. As $\{\|p_k\|\}$ is uniformly bounded by θ and $\liminf_{k\to\infty} \epsilon_k > 0$, there exists $\bar k$ such that each component of $\beta_k p_k$ satisfies $|[\,\beta_k p_k\,]_i| < \epsilon_k$ for all $k \ge \bar k$ in $\mathcal{G}$. The assumptions on components of p_k imply that $[\,p_k\,]_i > 0$ only if

$u_i - [\,x_k\,]_i > \epsilon_k$, and $[\,p_k\,]_i < 0$ only if $[\,x_k\,]_i - \ell_i > \epsilon_k$. It follows that for all $k \geq \bar{k}$ in $\mathcal{G}$, $\ell_i \leq [\,x_k + \beta_k p_k\,]_i \leq u_i$ and $\mathbf{proj}_{\Omega}(x_k + \beta_k p_k) = x_k + \beta_k p_k$.

Let $\bar{\mathcal{G}}$ denote the indices $k \geq \bar{k}$ of iterations at which a reduction in the initial step length was necessary, i.e., $\bar{\mathcal{G}} = \{\, k : t_k > 0,\ k \in \mathcal{G},\ k \geq \bar{k}\,\}$. Since α_k converges to zero, $\bar{\mathcal{G}}$ must be an infinite set. By definition,

$$f(x_k + \beta_k p_k) = f(\mathbf{proj}_{\Omega}(x_k + \beta_k p_k)) > f(x_k) + \beta_k \eta_A \nabla f(x_k)^T p_k, \ \text{ for all } \ k \in \bar{\mathcal{G}}.$$

Adding $-\beta_k \nabla f(x_k)^T p_k$ to both sides and rearranging gives

$$\begin{aligned} f(x_k + \beta_k p_k) - f(x_k) - \beta_k \nabla f(x_k)^T p_k &> -\beta_k (1 - \eta_A) \nabla f(x_k)^T p_k \\ &> \beta_k (1 - \eta_A) \bar{\epsilon}, \ \text{ for all } \ k \in \bar{\mathcal{G}}. \end{aligned} \tag{5.1}$$

The Taylor expansion of $f(x_k + \beta_k p_k)$ gives

$$f(x_k+\beta_k p_k) - f(x_k) - \beta_k \nabla f(x_k)^T p_k = \beta_k \int_0^1 \big(\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\big)^T p_k \, d\tau. \tag{5.2}$$

If $\|\cdot\|_D$ denotes the norm dual to $\|\cdot\|$, i.e., $\|x\|_D = \max_{v \neq 0} |x^T v| / \|v\|$, then

$$\big|\big(\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\big)^T p_k\big| \leq \|\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\|_D \|p_k\|.$$

If this inequality is substituted in (5.2), it then follows from (5.1) that

$$\begin{aligned} (1 - \eta_A)\bar{\epsilon} &< \int_0^1 \big(\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\big)^T p_k \, d\tau \\ &\leq \max_{0 \leq \tau \leq 1} \|\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\|_D \|p_k\|, \ \text{ for all } \ k \in \bar{\mathcal{G}}. \end{aligned}$$

The continuity of ∇f implies that there exists some $\tau_k \in [0, \beta_k]$ such that

$$\max_{0 \leq \tau \leq 1} \|\nabla f(x_k + \tau \beta_k p_k) - \nabla f(x_k)\|_D = \|\nabla f(x_k + \tau_k p_k) - \nabla f(x_k)\|_D.$$

Then

$$(1 - \eta_A)\bar{\epsilon} < \|\nabla f(x_k + \tau_k p_k) - \nabla f(x_k)\|_D \|p_k\|. \tag{5.3}$$

However, $\alpha_k p_k \to 0$ implies $\tau_k p_k \to 0$ for $k \in \mathcal{G}$, and the continuity of ∇f gives

$$\|\nabla f(x_k + \tau_k p_k) - \nabla f(x_k)\|_D \to 0.$$

As $\{\|p_k\|\}$ is uniformly bounded above by θ, the right-hand side of (5.3) converges to zero, which gives the required contradiction.

Next it will be shown by contradiction that every convergent subsequence of $\{|\nabla f(x_k)^T p_k|\}$ converges to zero regardless of the value of $\liminf_{k \to \infty} \big\|\mathit{\Pi}_k^T \nabla f(x_k)\big\|$. As $\mathit{\Pi}_k \mathit{\Pi}_k^T p_k = p_k$ for all k,

$$|\nabla f(x_k)^T p_k| = |\nabla f(x_k)^T \mathit{\Pi}_k \mathit{\Pi}_k^T p_k| \tag{5.4}$$

for all k. Suppose that there exists a convergent subsequence of $\{|\nabla f(x_k)^T p_k|\}$, say $\{|\nabla f(x_{k_j})^T p_{k_j}|\}$, that converges to a positive value. Then by (5.4), the sequence

$\{|\nabla f(x_{k_j})^T \Pi_{k_j} \Pi_{k_j}^T p_{k_j}|\}$ converges to a positive value. As $\{\|p_k\|\}$ is bounded by a constant θ,

$$\liminf_{j\to\infty} \left\|\Pi_{k_j}^T \nabla f(x_{k_j})\right\| > 0.$$

Applying the previous arguments to the subsequence $\{|\nabla f(x_{k_j})^T p_{k_j}|\}$ gives

$$\lim_{j\to\infty} |\nabla f(x_{k_j})^T p_{k_j}| = 0,$$

which is a contradiction.

As the level set $\mathcal{L}\big(f(x_0)\big)$ is bounded, $\{|\nabla f(x_k)^T p_k|\}$ is a bounded sequence. It follows that

$$\liminf_{k\to\infty} |\nabla f(x_k)^T p_k| = \limsup_{k\to\infty} |\nabla f(x_k)^T p_k| = 0.$$

Therefore, $\lim_{k\to\infty} |\nabla f(x_k)^T p_k| = 0.$ ∎

Theorem 5.2. (Projected search using quasi-Wolfe search) *Let f be a scalar-valued continuously differentiable function defined on $\Omega = \{\, x \in \mathbb{R}^n : \ell \le x \le u \,\}$. Assume that $x_0 \in \Omega$ is chosen such that the level set $\mathcal{L}\big(f(x_0)\big)$ is bounded, and $\{x_k\}$ is given by $x_{k+1} = x_k(\alpha_k)$, where α_k is a quasi-Wolfe step. For an arbitrarily fixed $\epsilon > 0$, define $\epsilon_0 = \epsilon$, and*

$$\epsilon_k = \min\big\{\epsilon, \left\|\Pi_{k-1}^T \nabla f(x_{k-1})\right\|\big\}.$$

for $k \ge 1$, where each Π_k is a matrix with orthonormal columns that spans the set of projected directions with respect to the working set $\mathcal{W}_k$. If $\{p_k\}$ is a sequence of descent directions with $\|p_k\| \le \theta$ for some constant θ independent of k, $\Pi_k \Pi_k^T p_k = p_k$ for all k, and the components of p_k satisfy $[\,p_k\,]_i \ge 0$ if $[\,x_k\,]_i \le \ell_i + \epsilon_k$, and $[\,p_k\,]_i \le 0$ if $[\,x_k\,]_i \ge u_i - \epsilon_k$, then

$$\lim_{k\to\infty} |\nabla f(x_k)^T p_k| = 0.$$

Proof. First, we show that $\lim_{k\to\infty} |\nabla f(x_k)^T p_k| = 0$ if $\liminf_{k\to\infty} \left\|\Pi_k^T \nabla f(x_k)\right\| \ne 0$. The first quasi-Wolfe condition ($\mathbf{C}_1$) is equivalent to the quasi-Armijo condition, and the arguments in the proof of Theorem 5.1 may be used to show that $\{f(x_k)\}$ is a convergent sequence. This implies that

$$\lim_{k\to\infty} \alpha_k \nabla f(x_k)^T p_k = 0.$$

The proof is by contradiction. Suppose that $|\nabla f(x_k)^T p_k| \not\to 0$ as $k \to \infty$, then there exists some $\bar\epsilon > 0$ such that $|\nabla f(x_k)^T p_k| > \bar\epsilon$ infinitely often. Let $\mathcal{G} = \{\, k : |\nabla f(x_k)^T p_k| > \bar\epsilon \,\}$, then it must be that $\alpha_k \to 0$ for $k \in \mathcal{G}$. As $\{\|p_k\|\}$ is uniformly bounded above by θ, $\alpha_k p_k \to 0$ for $k \in \mathcal{G}$.

If the quasi-Wolfe condition ($\mathbf{C}_2$) is satisfied, then

$$\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) \ge -\eta_W |\nabla f(x_k)^T p_k|.$$

Similarly, if the quasi-Wolfe condition $(\mathbf{C}_4)$ is satisfied, then

$$\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) \geq 0 \geq -\eta_W |\nabla f(x_k)^T p_k|.$$

In either case, as $\nabla f(x_k)^T p_k < 0$, it must hold that

$$\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k \geq (1-\eta_W)|\nabla f(x_k)^T p_k| > (1-\eta_W)\bar{\epsilon}, \quad \text{for } k \in \mathcal{G}.$$

The application of the triangle inequality yields

$$\begin{aligned} 0 < (1-\eta_W)\bar{\epsilon} &< \big|\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\big| \\ &\leq \big|\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k)\big| \\ &\qquad + \big|\nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\big|. \end{aligned} \tag{5.5}$$

Let $\|\cdot\|_D$ denote the norm dual to $\|\cdot\|$, then

$$\begin{aligned} &\big|\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k)\big| \\ &\quad \leq \|\nabla f\big(x_k(\alpha_k)\big) - \nabla f(x_k)\|_D \|P_{x_k(\alpha_k)}(p_k)\| \leq \|\nabla f\big(x_k(\alpha_k)\big) - \nabla f(x_k)\|_D \|p_k\|. \end{aligned}$$

As ∇f is continuous and $\|p_k\|$ is uniformly bounded, the right-hand side of this inequality must converge to zero for $k \in \mathcal{G}$, which implies that

$$\Big|\nabla f\big(x_k(\alpha_k)\big)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k)\Big| \to 0, \quad \text{for } k \in \mathcal{G}.$$

Basic norm inequalities give

$$\begin{aligned} \big|\nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\big| &\leq \|\nabla f(x_k)\|_D \|P_{x_k(\alpha_k)}(p_k) - p_k\| \\ &= \|\nabla f(x_k)\|_D \|P_{x_k(\alpha_k)}(p_k) - P_{x_k}(p_k)\|. \end{aligned}$$

As the level set $\mathcal{L}\big(f(x_0)\big)$ is bounded, and the gradient ∇f is continuous, the sequence of dual norms $\{\|\nabla f(x_k)\|_D\}$ is uniformly bounded. The hypothesis that $\liminf_{k\to\infty} \big\|\Pi_k^T \nabla f(x_k)\big\| \neq 0$ implies $\liminf_{k\to\infty} \epsilon_k > 0$. Also, because

$$\|x_k(\alpha_k) - x_k\| \leq \|\alpha_k p_k\| \to 0, \quad \text{for } k \in \mathcal{G},$$

there must exist an $\bar{k}$ such that for all $k \geq \bar{k}$ in $\mathcal{G}$,

$$[\, x_k(\alpha_k) - x_k \,]_i < \epsilon_k.$$

From the assumptions on the components of p_k, it must hold that for all $k \geq \bar{k}$ in $\mathcal{G}$, $[\, p_k \,]_i < 0$ only if $[\, x_k \,]_i > \ell_i + \epsilon_k$, in which case $[\, x_k(\alpha_k) \,]_i > \ell_i$; and $[\, p_k \,]_i > 0$ only if $[\, x_k \,]_i < u_i - \epsilon_k$, in which case $[\, x_k(\alpha_k) \,]_i < u_i$. It follows that, for $k \in \mathcal{G}$ sufficiently large,

$$P_{x_k(\alpha_k)}(p_k) = P_{x_k}(p_k) = p_k.$$

Therefore,

$$\|\nabla f(x_k)\|_D \|P_{x_k(\alpha_k)}(p_k) - P_{x_k}(p_k)\| \to 0, \quad \text{for } k \in \mathcal{G},$$

and consequently

$$\left|\nabla f(x_k)^T P_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\right| \to 0, \text{ for } k \in \mathcal{G}.$$

It follows that the right-hand side of (5.5) converges to zero for $k \in \mathcal{G}$, which gives the required contradiction.

It remains to consider the case where the quasi-Wolfe condition ($\mathbf{C}_3$) is satisfied, i.e.,

$$\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) \ge -\eta_W |\nabla f(x_k)^T p_k|.$$

The assumption that $\nabla f(x_k)^T p_k < 0$ gives

$$\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k \ge (1-\eta_W)|\nabla f(x_k)^T p_k| > (1-\eta_W)\bar{\epsilon}, \text{ for } k \in \mathcal{G},$$

which implies that

$$\begin{aligned} 0 < (1-\eta_W)\bar{\epsilon} &< \left|\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\right| \\ &\le \left|\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P^-_{x_k(\alpha_k)}(p_k)\right| \\ &\quad + \left|\nabla f(x_k)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\right|. \end{aligned} \tag{5.6}$$

The definition of the dual norm yields

$$\begin{aligned} &\left|\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P^-_{x_k(\alpha_k)}(p_k)\right| \\ &\qquad \le \|\nabla f\big(x_k(\alpha_k)\big) - \nabla f(x_k)\|_D \|P^-_{x_k(\alpha_k)}(p_k)\| \le \|\nabla f\big(x_k(\alpha_k)\big) - \nabla f(x_k)\|_D \|p_k\|. \end{aligned}$$

From the continuity of ∇f and uniform boundedness of $\|p_k\|$, the right-hand side of the above inequality converges to zero for $k \in \mathcal{G}$, which means that

$$\left|\nabla f\big(x_k(\alpha_k)\big)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T P^-_{x_k(\alpha_k)}(p_k)\right| \to 0, \text{ for } k \in \mathcal{G}.$$

Also,

$$\begin{aligned} \left|\nabla f(x_k)^T P^-_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\right| &\le \|\nabla f(x_k)\|_D \|P^-_{x_k(\alpha_k)}(p_k) - p_k\| \\ &= \|\nabla f(x_k)\|_D \|P^-_{x_k(\alpha_k)}(p_k) - P_{x_k}(p_k)\|. \end{aligned}$$

As the level set $\mathcal{L}\big(f(x_0)\big)$ is bounded, and ∇f is continuous, it must hold that the sequence of dual norms $\{\|\nabla f(x_k)\|_D\}$ is uniformly bounded. Also, as

$$\|x_k(\alpha_k) - x_k\| \le \|\alpha_k p_k\| \to 0, \text{ for } k \in \mathcal{G},$$

arguments analogous to those used to establish convergence in cases ($\mathbf{C}_2$) and ($\mathbf{C}_4$) give

$$P^-_{x_k(\alpha_k)}(p_k) = P_{x_k}(p_k) = p_k \text{ for } k \in \mathcal{G} \text{ sufficiently large,}$$

in which case

$$\|\nabla f(x_k)\|_D \|P^-_{x_k(\alpha_k)}(p_k) - P_{x_k}(p_k)\| \to 0, \text{ for } k \in \mathcal{G}.$$

This implies that

$$\left|\nabla f(x_k)^T P^{-}_{x_k(\alpha_k)}(p_k) - \nabla f(x_k)^T p_k\right| \to 0, \text{ for } k \in \mathcal{G}.$$

It follows that the right-hand side of (5.6) converges to zero for $k \in \mathcal{G}$, which gives the required contradiction.

Finally, the same arguments from the proof of Theorem 5.1 imply that

$$\lim_{k\to\infty} |\nabla f(x_k)^T p_k| = 0$$

regardless of the value of $\liminf_{k\to\infty} \left\|\Pi_k^T \nabla f(x_k)\right\|$. ∎

Based on the framework described in Section 4, the limit $\lim_{k\to\infty} |\nabla f(x_k)^T p_k| = 0$ implies that

$$\lim_{k\to\infty} |\nabla f(x_k)^T d_k| = 0, \tag{5.7}$$

which would further imply that the projected gradient, $\Pi_k \Pi_k^T \nabla f(x_k)$, converges to zero for an appropriate choice of d_k. For example, if $d_k = -\Pi_k \Pi_k^T \nabla f(x_k)$, or d_k is the solution of the subproblem (4.2) with the two-norm of the projected approximate Hessian, $\|\Pi_k^T H_k \Pi_k\|$, uniformly bounded, then it may be verified that (5.7) implies that $\|\Pi_k^T \nabla f(x_k)\| \to 0$.

Under the nondegeneracy assumption defined below, any algorithm based on the proposed framework for which $\|\Pi_k^T \nabla f(x_k)\| \to 0$ will identify the optimal active set in a finite number of iterations.

Definition 5.1. *A point $x^* \in \Omega$ is a* stationary point *of* (BC) *if $\nabla_i f(x^*) = 0$ for $\ell_i < x_i^* < u_i$, $\nabla_i f(x^*) \geq 0$ for $x_i^* = \ell_i$ and $\ell_i < u_i$, and $\nabla_i f(x^*) \leq 0$ for $x_i^* = u_i$ and $\ell_i < u_i$. A stationary point x^* is* nondegenerate *if $\nabla_i f(x^*) > 0$ for $x_i^* = \ell_i$ and $\ell_i < u_i$, and $\nabla_i f(x^*) < 0$ for $x_i^* = u_i$ and $\ell_i < u_i$.*

The next result shows that a projected-search method with either a quasi-Armijo or quasi-Wolfe search will identify the optimal active set in a finite number of iterations.

Theorem 5.3.
In addition to the assumptions of Theorem 5.1 *or* Theorem 5.2, *assume that $\{x_k\}$ converges to a nondegenerate stationary point x^*. Consider the extended active set*

$$\mathcal{A}_{\epsilon_k}(x_k) = \left\{\, i : [\, x_k\,]_i \leq \ell_i + \epsilon_k \;\text{ or }\; [\, x_k\,]_i \geq u_i - \epsilon_k \,\right\}.$$

If $\|\Pi_k^T \nabla f(x_k)\| \to 0$, then $\mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x_k) = \mathcal{A}(x^*)$ for all k sufficiently large.

Proof. First, we show that $\mathcal{A}(x^*) \subset \mathcal{A}_{\epsilon_k}(x_k)$ for k sufficiently large by contradiction. Assume the opposite is true, then there exists $i \in \mathcal{A}(x^*)$ such that $i \notin \mathcal{A}_{\epsilon_k}(x_k)$ for an infinite subsequence $\mathcal{K}$, which implies that $i \notin \mathcal{W}_k$ for all $k \in \mathcal{K}$. It follows that

$$|\nabla_i f(x_k)| \leq \|\Pi_k^T \nabla f(x_k)\| \text{ for } k \in \mathcal{K}.$$

As f is continuously differentiable and $\|\Pi_k^T \nabla f(x_k)\| \to 0$, letting $k \to \infty$ in $\mathcal{K}$ gives

$$|\nabla_i f(x^*)| = \lim_{k\to\infty, k\in\mathcal{K}} |\nabla_i f(x_k)| = 0.$$

This contradicts the nondegeneracy of x^*.

Now we show that $\mathcal{A}_{\epsilon_k}(x_k) \subset \mathcal{A}(x^*)$ for k sufficiently large. If $\ell_i = u_i$, a simple argument gives $i \in \mathcal{A}_{\epsilon_k}(x_k)$ and $i \in \mathcal{A}(x^*)$. Consider an index i such that $\ell_i < u_i$. From the definition of ϵ_k, the assumption $\|\Pi_k^T \nabla f(x_k)\| \to 0$ implies that $\epsilon_k \to 0$. Hence, for k sufficiently large, $\ell_i + \epsilon_k < u_i - \epsilon_k$. If $i \notin \mathcal{A}(x^*)$, then $\ell_i < [\,x^*\,]_i < u_i$. As $\{x_k\} \to x^*$ and $\epsilon_k \to 0$, $\ell_i + \epsilon_k < [\,x_k\,]_i < u_i - \epsilon_k$ for k sufficiently large, which implies that $i \notin \mathcal{A}_{\epsilon_k}(x_k)$. Therefore, if $i \notin \mathcal{A}(x^*)$, then $i \notin \mathcal{A}_k(x_k)$, i.e. $\mathcal{A}_{\epsilon_k}(x_k) \subset \mathcal{A}(x^*)$ for k sufficiently large. We conclude that $\mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x^*)$ for all k sufficiently large.

It remains to show that $\mathcal{A}(x_k) = \mathcal{A}_{\epsilon_k}(x_k)$ for k sufficiently large. Obviously $\mathcal{A}(x_k) \subset \mathcal{A}_{\epsilon_k}(x_k)$ for all k. It is trivial if $\ell_i = u_i$. Now consider the case where $\ell_i < u_i$. Note that $\{x_k\} \to x^*$ implies $\lim_{k\to\infty} \|x_{k+1} - x_k\| = 0$. As $\lim_{k\to\infty}(u_i - \epsilon_{k+1}) - (\ell_i + \epsilon_k) = u_i - \ell_i > 0$, $|[\,x_{k+1} - x_k\,]_i| < (u_i - \epsilon_{k+1}) - (\ell_i + \epsilon_k)$ for k sufficiently large. Suppose k_0 is such that, for all $k \ge k_0$, $\mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x^*)$ and $|[\,x_{k+1} - x_k\,]_i| < (u_i - \epsilon_{k+1}) - (\ell_i + \epsilon_k)$. The inclusion $\mathcal{A}_{\epsilon_k}(x_k) \subset \mathcal{A}(x_k)$ for all $k \ge k_0$ is established using a contradiction argument. Assume that there exists $i \in \mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x^*)$ for all $k \ge k_0$, but $i \notin \mathcal{A}(x_k)$ for some $\bar{k} \ge k_0$. Then either $\ell_i < [\,x_{\bar{k}}\,]_i \le \ell_i + \epsilon_{\bar{k}}$ or $u_i - \epsilon_{\bar{k}} \le [\,x_{\bar{k}}\,]_i < u_i$. If the inequality $\ell_i < [\,x_{\bar{k}}\,]_i \le \ell_i + \epsilon_{\bar{k}}$ holds, the definition of p_k in Algorithm 3 implies that $[\,p_{\bar{k}}\,]_i \ge 0$, and it must be the case that $\ell_i < [\,x_{\bar{k}}\,]_i \le [\,x_{\bar{k}+1}\,]_i$. In addition, $\big|[\,x_{\bar{k}+1} - x_{\bar{k}}\,]_i\big| < (u_i - \epsilon_{\bar{k}+1}) - (\ell_i + \epsilon_{\bar{k}})$ implies that $[\,x_{\bar{k}+1}\,]_i < u_i - \epsilon_{\bar{k}+1}$. As $i \in \mathcal{A}_{\epsilon_{\bar{k}+1}}(x_{\bar{k}+1})$, it must hold that $\ell_i < [\,x_{\bar{k}}\,]_i \le [\,x_{\bar{k}+1}\,]_i \le \ell_i + \epsilon_{\bar{k}+1}$. Inductively, for all $k \ge \bar{k}$, $\ell_i < [\,x_{\bar{k}}\,]_i \le [\,x_k\,]_i \le \ell_i + \epsilon_k$, which implies that $[\,x^*\,]_i \ge [\,x_{\bar{k}}\,]_i > \ell_i$. A similar argument shows that if $u_i - \epsilon_{\bar{k}} \le [\,x_{\bar{k}}\,]_i < u_i$, then $[\,x^*\,]_i \le [\,x_{\bar{k}}\,]_i < u_i$. It follows that $i \notin \mathcal{A}(x^*)$, which contradicts the assumption that $i \in \mathcal{A}_{\epsilon_k}(x_k) = \mathcal{A}(x^*)$ for all $k \ge k_0$. Therefore, $\mathcal{A}_{\epsilon_k}(x_k) \subset \mathcal{A}(x_k)$ for all $k \ge k_0$, which completes the proof. ∎

A simple example shows that the nondegeneracy of a stationary point is necessary for identifying the optimal active set in a finite number of iterations. Let $f : \mathbb{R}^2 \to \mathbb{R}$ be given by $f(x) = \frac{1}{5}\|x\|^2$, and let $\Omega = \{\, x \in \mathbb{R}^2 : x \ge 0 \,\}$. For this problem $x^* = (0,0)^T$ is a degenerate stationary point and the global minimizer of f over Ω. Assume that the step length $\alpha_k \le 1$ for all k, and let $\epsilon = \frac{1}{\sqrt{2}}$. Starting from $x_0 = (1,1)^T$, the projected-gradient method gives

$$x_k = \prod_{j=0}^{k}(1 - \tfrac{2}{5}\alpha_j)\begin{pmatrix}1\\1\end{pmatrix}, \quad \text{and} \quad \epsilon_k = \tfrac{2}{5}\|x_{k-1}\| = \tfrac{2\sqrt{2}}{5}\prod_{j=0}^{k-1}\big(1 - \tfrac{2}{5}\alpha_j\big)$$

for $k \ge 1$. Then $\{x_k\}$ converges to the degenerate stationary point x^*, and

$$[\,x_k\,]_i = \prod_{j=0}^{k}(1 - \tfrac{2}{5}\alpha_j) > \tfrac{2\sqrt{2}}{5}\prod_{j=0}^{k-1}\big(1 - \tfrac{2}{5}\alpha_j\big) = \epsilon_k, \quad i = 1, 2$$

for all $k \ge 1$. It follows that $\mathcal{A}_{\epsilon_k}(x_k) = \emptyset$ for all k, although $\mathcal{A}(x^*) = \{\, 1, 2 \,\}$.

6. Numerical Experiments

In this section we illustrate the numerical performance of the projected-search methods described in Section 4. All testing was done on problems taken from the CUTEst test collection (see Bongartz et al. [5] and Gould, Orban and Toint [18]). As of July 1, 2020, the CUTEst test set contains 154 bound-constrained problems of the form (BC). Although many problems allow for the number of variables and constraints to be adjusted in the SIF data file, our tests used the default dimensions set in the CUTEst distribution. This gave problems ranging in size from `BQ1VAR` (one variable) to `WALL100` (149624 variables).

The practical effectiveness of the quasi-Wolfe search was evaluated by running two limited-memory quasi-Newton methods, one with a quasi-Wolfe search and the other with a quasi-Armijo search. The resulting implementations, `LRHB-qWolfe`, and `LRHB-qArmijo` are based on the Fortran package `LRHB` (see Ferry et al. [12]). In the quasi-Wolfe search, the kink steps are sorted in decreasing order in $O(n \log n)$ flops using a heapsort algorithm (see, e.g., Williams [29], Knuth [21, Section 5.2.3]), adapted from a Fortran implementation by Byrd et al. [7]. For `LRHB-qWolfe`, the Armijo tolerance η_A was set at 10^{-4} and the Wolfe tolerance $\eta_W = 0.9$. In `LRHB-qArmijo`, $\eta_A = 0.3$. The scalar ϵ was set to the machine precision in the expression for ϵ_k in the calculation (4.1) of the working set.

In order to provide some measure of the efficiency of the projected-search method relative to a state-of-the-art method for bound-constrained optimization, the solvers `LRHB-qWolfe` and `LRHB-qArmijo` were compared with the limited-memory method `LBFGS-B` (Byrd et al. [7], Zhu et al. [32], and Morales and Nocedal [23]). All three solvers were applied to the 154 bound-constrained problems from the CUTEst test set. The runs were terminated at the first point x_T such that

(a) $\|P_{x_T}\big(-\nabla f(x_T)\big)\|_\infty \le 10^{-5}\big(1 + |f(x_T)|\big)$ and

(b) $|f(x_T) - f(x_{T-1})| \le 10^7 \epsilon_M \times \max\big\{\, |f(x_T)|, |f(x_{T-1})|, 1 \,\big\}$; or

(c) $\|P_{x_T}\big(-\nabla f(x_T)\big)\|_\infty < \sqrt{\epsilon_M}$,

where ϵ_M is the machine precision. In the first iteration of the algorithms, only condition (c) is tested. A nonoptimal termination was signaled by the violation of a time limit of 3600 seconds, a limit of 10^6 iterations, or an abnormal exit because of numerical difficulties.

The solver `LRHB-qArmijo` failed on nine problems, with six failing because of numerical difficulties (`BLEACHNG`, `BQPGAUSS`, `BRATU1D`, `GRIDGENA`, `RAYBENDL`, `WALL10`, and `WEEDS`). `LRHB-qWolfe` failed on six problems, with four failures caused by numerical difficulties (`GRIDGENA`, `PALMER5E`, `PROBPENL`, and `WALL10`). `LRHB-qWolfe` identified problem `BRATU1D` as being unbounded. For both solvers, `CYCLOOCTLS` and `WALL50` could not be solved within the one hour time limit. In the cases of numerical difficulties, the search algorithms were unable to compute an appropriate step. We note that for `LRHB-qWolfe`, the run for `PROBPENL` terminated at a near-optimal point that satisfied condition (a) and $\|P_{x_T}\big(-\nabla f(x_T)\big)\|_\infty = 1.99 \times 10^{-7}$. The solver `LBFGS-B` failed on 16 problems. Seven failures were caused by numerical difficulties

(BQPGAUSS, BRATU1D, GRIDGENA, PALMER5A, PALMER5B, PALMER7A, and WALL10), seven problems exceeded the iteration limit (CHEBYQAD, PALMER1E, PALMER2E, PALMER3E, PALMER4E, PALMER6E, and PALMER8E), and two problems exceeded the time limit (CYCLOOCTLS and WALL50). More details of the runs are given by Ferry et al. [14].

The relative performance of the solvers is summarized using performance profiles (in $\log_2$ scale), which were proposed by Dolan and Moré [10]. Let $\mathcal{P}$ denote a set of problems used for a given numerical experiment. For each method s we define the function $\pi_s : [0, r_M] \mapsto \mathbb{R}^+$ such that

$$\pi_s(\tau) = \frac{1}{n_p} \left| \left\{ p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau \right\} \right|,$$

where n_p is the number of problems in the test set and $r_{p,s}$ denotes the ratio of the number of function evaluations needed to solve problem p with method s and the least number of function evaluations needed to solve problem p. If method s failed for problem p, then $r_{p,s}$ is set to be twice of the maximal ratio. The parameter r_M is the maximum value of $\log_2(r_{p,s})$. Figure 3 gives the performance profiles for the 154

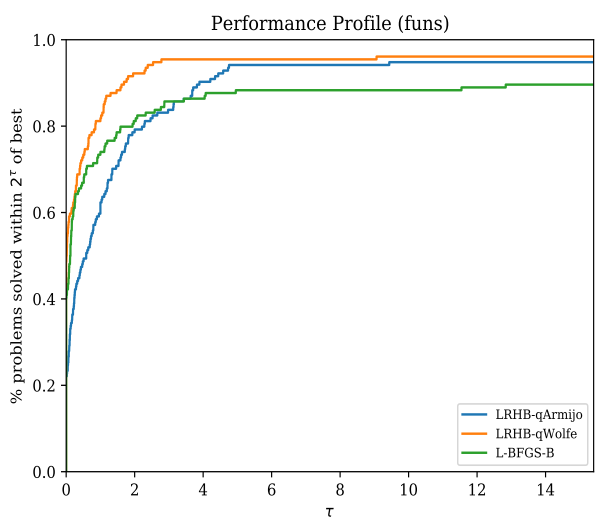

Figure 3: Performance profiles for the number of function evaluations required to solve 154 bound-constrained problems from the CUTEst test set. The figure gives the profiles for the three solvers LRHB-qWolfe, LRHB-qArmijo, and LBFGSB [7].

problems for LRHB-qWolfe, LRHB-qArmijo, and LBFGS-B. The profile utilized the total number of function evaluations for comparison. Additional information about the runs used to generate the performance profiles is given by Ferry et al. [14]. The results indicate that using a quasi-Wolfe search in LRHB resulted in a substantially better performance with respect to function calls than using a quasi-Armijo search, and comparable and more robust performance with respect to LBFGS-B.

A benefit of the Wolfe conditions in the unconstrained case is that the restriction on the directional derivative guarantees that the approximate curvature $\big(\nabla f(x_{k+1}) -$

$\nabla f(x_k)\big)^T(x_{k+1}-x_k)$ is positive, which is a necessary condition for the quasi-Newton update to give a positive-definite approximate Hessian. In the bound-constrained case, the use of a quasi-Wolfe projected search makes it more likely that the update can be applied, but it is not possible to guarantee an update in all cases. If the next iterate is given by $x_{k+1} = \mathbf{proj}_{\Omega}(x_k + \alpha_k p_k)$, where α_k is a quasi-Wolfe step, then $\big(\nabla f(x_{k+1}) - \nabla f(x_k)\big)^T(x_{k+1} - x_k)$ need not be greater than zero if the path $\mathbf{proj}_{\Omega}(x_k+\alpha_k p_k)$ changes direction for some $\alpha \in (0, \alpha_k)$. If it does change direction, $\psi'_+(0)$ and $\psi'_-(\alpha_k)$ may be directional derivatives of f in a direction other than $x_{k+1} - x_k$. This situation is illustrated in Figure 4, which depicts a two-dimensional region with lower bounds $x_1 = 0$ and $x_2 = 0$. In this example $\psi'_+(0)$ is a directional derivative of f in direction $[\,p_k\,]_1$ and $\psi'_-(\alpha_k)$ is a directional derivative of f in direction $[\,p_k\,]_2$. As a result, if the path changes direction for $\alpha \in (0, \alpha_k)$, then there is the possibility that the quasi-Newton update must be skipped.

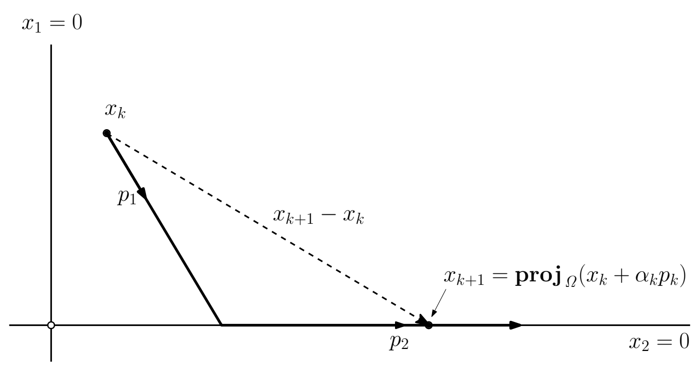

Figure 4: Example with no guarantee of an update for the approximate Hessian.

It is shown in Section 5 that if $\{x_k\}$ converges to a nondegenerate stationary point, then a quasi-Wolfe search identifies the active set at the solution in a finite number of iterations. After the active set stabilizes, a quasi-Wolfe search behaves exactly like a Wolfe line search in the sense that updates to the approximate Hessian are guaranteed if $f(x_k + \alpha p_k)$ is bounded below.

To estimate how often the update is likely to be skipped with the quasi-Wolfe search, statistics were collected from the test problems for which at least one of the search paths was "bent" by projection. The application of `LRHB-qWolfe` resulted in 259 of the potential 637268 updates being skipped ($\approx$ 0.04%). This can be compared to 6537 of the 679071 updates being skipped ($\approx$ 1.0%) for `LRHB-qArmijo`. (The number of updates reflects the number of iterations needed for convergence.)

7. Summary and Conclusions

A framework for the development of a general class of projected-search methods for bound-constrained minimization has been proposed. Methods within this framework compute a descent direction with respect to an extended active set and utilize a new quasi-Wolfe search that is appropriate for a function defined on a piecewise-linear continuous path. The behavior of the line search is similar to that of a conventional Wolfe line search, except that a step is accepted under a wider range of conditions. These conditions take into consideration steps at which the restriction of the objective function on the search path is not differentiable. As in the unconstrained case, the quasi-Wolfe step can be computed using safeguarded polynomial interpolation and the accuracy of the step can be adjusted. Standard existence and convergence results associated with a conventional Wolfe line search are extended to the quasi-Wolfe case. In addition, under a nondegeneracy assumption, any method within the framework will identify the optimal active set in a finite number of iterations. It follows that once the optimal active set has been identified, any method based on the proposed framework will have the same convergence rate as its unconstrained counterpart.

Numerical results indicate that a projected-search method implemented with a quasi-Wolfe search can require substantially fewer function evaluations compared to the same method with a quasi-Armijo search. Moreover, a particular method based on a limited-memory quasi-Newton method to obtain the feasible descent direction is shown to be competitive with the state-of-the-art package `LBFGS-B`.

References

[1] L. Armijo. Minimization of functions having Lipschitz continuous first partial derivatives. *Pacific Journal of Mathematics*, 16:1–3, 1966. 2

[2] D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Automatic Control*, AC-21(2):174–184, 1976. 2, 3

[3] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1982. 18

[4] D. P. Bertsekas. Projected Newton methods for optimization problems with simple constraints. *SIAM J. Control Optim.*, 20(2):221–246, 1982. 2, 3, 4

[5] I. Bongartz, A. R. Conn, N. I. M. Gould, and Ph. L. Toint. CUTE: Constrained and unconstrained testing environment. *ACM Trans. Math. Software*, 21(1):123–160, 1995. 26

[6] R. P. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1973. Prentice-Hall Series in Automatic Computation. 7

[7] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Comput.*, 16:1190–1208, 1995. 26, 27

[8] P. H. Calamai and J. J. Moré. Projected gradient methods for linearly constrained problems. *Math. Program.*, 39:93–116, 1987. 2, 3

[9] J. E. Dennis, Jr. and R. B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996. Corrected reprint of the 1983 original. 2

[10] E. D. Dolan and J. J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002. 27

[11] M. W. Ferry. *Projected-Search Methods for Box-Constrained Optimization*. PhD thesis, Department of Mathematics, University of California, San Diego, May 2011. 2

[12] M. W. Ferry, P. E. Gill, E. Wong, and M. Zhang. A limited-memory reduced-Hessian method for bound-constrained optimization. Center for Computational Mathematics Report CCoM 20-02, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA, 2020. 2, 19, 26

[13] M. W. Ferry, P. E. Gill, E. Wong, and M. Zhang. Projected-search methods for bound-constrained optimization. Center for Computational Mathematics Report CCoM 20-01, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA, 2020. 19

[14] M. W. Ferry, P. E. Gill, E. Wong, and M. Zhang. Supplementary numerical results for projected-search methods for bound-constrained optimization. Center for Computational Mathematics Report CCoM 20-08, Center for Computational Mathematics, University of California, San Diego, La Jolla, CA, 2020. 27

[15] N. Ghosh and W. W. Hager. A derivative-free bracketing scheme for univariate minimization. *Computers Math. Applic.*, 20(2):23–24, 1990. 7

[16] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright. A note on a sufficient-decrease criterion for a nonderivative step-length procedure. *Math. Programming*, 23(3):349–352, 1982. 3

[17] A. A. Goldstein. Convex programming in Hilbert space. *Bulletin of the American Mathematical Society*, 70(5):709–710, 1964. 2

[18] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEr and SifDec: A constrained and unconstrained testing environment, revisited. *ACM Trans. Math. Software*, 29(4):373–394, 2003. 26

[19] W. W. Hager. A derivative-based bracketing scheme for univariate minimization and the conjugate gradient method. *Computers Math. Applic.*, 18(9):779–795, 1989. 3, 7

[20] D. Kim, S. Sra, and I. S. Dhillon. Tackling box-constrained optimization via a new projected quasi-Newton approach. *SIAM J. Sci. Comput.*, 32(6):3548–3563, December 2010. 2, 3

[21] D. Knuth. *The Art of Computer Programming, 3*. Addison-Wesley Publishing Company, Redwood City, third edition, 1997. 17, 26

[22] E. S. Levitin and B. T. Polyak. Constrained minimization methods. *U.S.S.R. Comput. Math. and Math. Physics*, 6(5):1–50, 1966. 2

[23] J. L. Morales and J. Nocedal. Remark on "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization". *ACM Trans. Math. Softw.*, 38(1):7:1–7:4, December 2011. 26

[24] J. J. Moré and D. C. Sorensen. Newton's method. In G. H. Golub, editor, *Studies in Mathematics, Volume 24. MAA Studies in Numerical Analysis*, pages 29–82. Math. Assoc. America, Washington, DC, 1984. 5

[25] J. J. Moré and D. J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software*, 20(3):286–307, 1994. 3, 5, 7

[26] Q. Ni and Y.-X. Yuan. A subspace limited memory quasi-Newton algorithm for large-scale nonlinear bound constrained optimization. *Math. Comput.*, 66:1509–1520, 10 1997. 2, 3

[27] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999. 5

[28] J. M. Ortega and W. C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original. 2

[29] J. W. J. Williams. Algorithm 232 - Heapsort. *Communications of the Association for Computing Machinery*, 7:347–348, 1964. 17, 26

[30] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11:226–235, 1969. 3

[31] P. Wolfe. On the convergence of gradient methods under constraint. *IBM J. Res. Dev.*, 16:407–411, 1972. 5

[32] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal. Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Software*, 23(4):550–560, 1997. 26