An active set method for bound-constrained optimization

Arnold Neumaier
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: Arnold.Neumaier@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~neum`

Behzad Azmi
Department of Mathematics and Statistics, University of Konstanz
Universitätsstraße 10, D-78457 Konstanz, Germany
email: behzad.azmi@uni-konstanz.de

Morteza Kimiaei
Fakultät für Mathematik, Universität Wien
Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria
email: kimiaeim83@univie.ac.at
WWW: `http://www.mat.univie.ac.at/~kimiaei`

Abstract. In this paper, a class of algorithms is developed for bound-constrained optimization. The new scheme uses the gradient-free line search along bent search paths. Unlike traditional algorithms for bound-constrained optimization, our algorithm ensures that the reduced gradient becomes arbitrarily small. It is also proved that all strongly active variables are found and fixed after finitely many iterations. A Matlab implementation of a bound-constrained solver `LMBOPT` based on the new theory was discussed by the present authors in a companion paper (*Math. Program. Comput.* **14** (2022), 271–318).

Keywords. Bound constrained optimization, active set strategy, line search method, global convergence.

2000 AMS Subject Classification: primary 90C06, 90C26, 90C30.

Acknowledgment The third author acknowledges financial support of the Austrian Science Foundation under Project No. P 34317.

April 14, 2023

1 An overview of our method

This paper addresses the theoretical properties of a generic solver `BOPT` for bound-constrained optimization problems. `BOPT` typically proceeds through three phases with distinct characteristics:

(i) In the initial phase, the `BOPT` iterations move down into a valley; most optimal active variables will be properly adjusted if appropriately bent search paths (defined in Section 3.2) are used. We must be careful not to use poor search directions, such as the steepest descent direction that leads to zigzagging.

(ii) In the second phase, the `BOPT` iterations move along the valley toward the minimizer. This phase can be long if the valley is long, steep and curved, or short and even absent if the valley is completely round. To be sure of approaching the minimizer, the search directions must satisfy the conditions that allow the method to be convergent.

(iii) In the third phase, the `BOPT` iterations are near the minimizer, but must find it with the desired accuracy. Here, a good choice of search direction is crucial. Since the function near a minimizer is usually nearly quadratic, a good method at this stage must choose the search direction and step sizes to ensure a good behavior for quadratic functions.

1.1 Search direction

Three conditions on search directions must be applied to prove convergence in Section 6 and the identification of the active set at the solution after a limit number of iterations:

(i) Directions in the subspace of active variables must be zero.

(ii) The bounded angle condition (defined by (21) below) must be satisfied.

(iii) The componentwise product of direction and gradient for all active components, which are not optimally active, must be nonpositive (discussed in Section 3.1).

1.2 A gradient-free bent line search

We use the line search `CLS` of Neumaier & Kimiaei [34]. Similarly to the Goldstein (Goldstein [23]) and the Armijo (Armijo [1]) line search frameworks, `CLS` is a gradient-free, i.e., it does not require gradient computation at each trial iterate. Furthermore, compared to the Goldstein and the Armijo frameworks, it appears numerically more efficient for severely nonconvex problems, see [32]. Because of the bound constraints, the search path must be bent (Bertsekas [3]) to produce only feasible points. We therefore discuss in detail (Section 3.2) the properties of bent search paths that are relevant to an analysis of descent properties.

1.3 A new active set strategy and its global convergence

We propose a new active set strategy (Algorithm 1) against zigzagging for bound-constrained optimization problems using a gradient-free bent line search. We update the working set based on the two appropriate choices I_- (minimal set) and I_+ (maximal set) defined in Section 2.4. These updates are performed relying on a condition imposed within the algorithm. This condition is expressed in terms of the norms of the reduced gradient and the gradient reduced to components of the working set I and ensures that a reduction in the latter implies a reduction of the former. As long as this condition holds, the algorithm continues exploring the manifold prescribed by I by fixing variables to their bounds or reducing the cardinality of the working set I as far as possible. Otherwise, the algorithm changes the underlying manifold, or equivalently the working set I, by freeing some active variables for which the complementarity condition is not satisfied. Due to this property, the zigzagging of the examples discussed in Section 2.4 is vanished (see Section 4) and it will be shown (Theorem 6.1) that zigzagging is theoretically vanished at finitely many iterations, like Bertsekas [3] and Conn et al. [13] and unlike active set methods proposed by Byrd et al. [10], Cristofari et al. [15], and Hager & Zhang [30]. On the other hand, active set strategies [3, 10, 13, 15, 30] use the projected gradient, which is not useful in finite precision arithmetic (discussed in Section 2.3) and therefore may lead to incorrect active variables.

1.4 Implementation

An implementation based on the new theory must take care of many other questions not covered by the theory, in particular regarding finite precision effects. A discussion of such implementation questions, details for a particular implementation in Matlab, and a thorough comparison with other state-of-the-art solvers was given by Kimiaei et al. [32]. The results (cf. the conclusion section below) shows that in most cases considered `LMBOPT` is the best solver in terms of the number of gradient and function evaluations compared to the state-of-the-art solvers.

Based on the theory given here, `LMBOPT` [32] discusses a limited memory method for solving the bound-constrained optimization problems whose objective function is continuously differentiable with a Lipschitz continuous gradient. `LMBOPT` finds the active variables and solves unconstrained optimization problems in the subspace of non-active variables.

2 Background

Convergence results for both the unconstrained and bound-constrained optimization methods have a long history. For complete references, we refer the reader to the book of Nocedal & Wright [39]; here we focus on pursuing only the references relevant to the present work, which aim at an improved convergence theory for line search methods that hold under much weaker assumptions than before.

2.1 Basic notation

Inequalities between vectors or matrices are interpreted component-wise. Under assuming that **generalized Cauchy–Schwarz inequality**

$$|y^T s| \leq \|y\|_* \|s\|$$

holds, for an arbitrary norm $\|\cdot\|$ we define the **dual norm**

$$\|y\|_* := \sup_{x \neq 0} \frac{y^T s}{\|s\|}.$$

A **scaled 1-norm** with

$$\|s\| := \sum_k \left|\frac{s_k}{w_k}\right|$$

and its dual norm, a **scaled maximum norm** with

$$\|y\|_* := \max_k w_k |y_k|,$$

are useful pairs of norms, where $w_k > 0$ stands for a weight of the kth component of a trial point. This norm will be numerically reasonable and provides a reasonable measure of the distance between the points at which we evaluate functions.

The **ellipsoidal norms**

$$\|p\| = \sqrt{p^T B p}, \quad \|g\|_* = \sqrt{g^T B^{-1} g} \tag{1}$$

are another useful pair of norms that require the matrix $B \in \mathbb{R}^{n \times n}$ must be symmetric and positive definite. For the identity matrix $B = I$, (1) becomes the standard **Euclidean norm** $\|s\|_2 := \sqrt{s^T s}$, which is its own dual (for more details see [38]).

In a **line search**, starting from the current point $x = x(0)$, points $x(\alpha)$ on a curve of feasible points parameterized by a **step size** $\alpha > 0$ are searched. The goal is to find a value for the step size such that $f(x(\alpha))$ is sufficiently smaller than $f(x)$, although the notion of being **sufficient** remains to be specified. If the gradient $g = g(x)$ is nonzero, then the existence of such $\alpha > 0$ is guaranteed if the tangent vector

$$p := x'(0) \tag{2}$$

exists and the following condition

$$g^T p < 0 \tag{3}$$

is satisfied. We say that p is a **descent direction** if it does not violate (3). In the unconstrained case, the curve is often considered as a ray of x in a descent direction p, giving $x(\alpha) = x + \alpha p$. In this case, the line search is called **straight**. If the curve is a piecewise linear path, it is called **bent**; otherwise it is called **curved**.

The optimization methods improve an initial feasible point x^0 by constructing a sequence $x^0, x^1, x^2, \ldots$ of feasible points with decreasing function values. To ensure this, we search in each iteration along an appropriate search path $x(\alpha)$ starting at the current point $x(0) = x^\ell$, and take $x^{\ell+1} = x(\alpha_\ell)$ where α_ℓ is determined by a line search based on function values only. If the iteration index ℓ is fixed, we simply write x for the current point x^ℓ.

2.2 Bound-constrained optimization problem

We consider the **bound-constrained optimization problem**

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \mathbf{x}, \end{array} \tag{4}$$

where the **objective function** $f : C \subseteq \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, and

$$\mathbf{x} := [\underline{x}, \overline{x}] := \{x \in \mathbb{R}^n \mid \underline{x} \le x \le \overline{x}\}$$

is a bounded or unbounded **box** in $\mathbb{R}^n$ describing the bounds on the variables. One-sided or missing bounds are also accounted by allowing components of the vector $\underline{x}$ of lower bounds to take the value $-\infty$ and components of the vector $\overline{x}$ of upper bounds to take the value ∞. A point x is called **feasible** if it belongs to the box $\mathbf{x}$. To have a well-defined optimization problem, the box $\mathbf{x}$ must be part of the domain C of definition[1] of f. We assume that the **gradient**

$$g(x) := \partial f(x)/\partial x = f'(x)^T \in \mathbb{R}^n.$$

is **Lipschitz continuous** in the feasible domain, i.e.,

$$\|g(x') - g(x)\|_* \le \overline{\gamma}\|x' - x\| \quad \text{for } x, x' \in \mathbf{x}. \tag{5}$$

The **Lipschitz constant** $\overline{\gamma}$ depends on the norm used, but not the notion of Lipschitz continuity, since all norms in $\mathbb{R}^n$ are equivalent.

2.3 Optimality conditions for bound constraints

Given a feasible point x and an index i, we call the bound $\underline{x}_i$ or $\overline{x}_i$ **active** if $x_i = \underline{x}_i$ or $x_i = \overline{x}_i$, respectively. In both cases, we also call the index i and the component x_i **active**. Otherwise, i.e., if $x_i \in]\underline{x}_i, \overline{x}_i[$, the index i, the component x_i, and the bounds $\underline{x}_i$ and $\overline{x}_i$ are called **nonactive** or **free**. A **corner** of the box $\mathbf{x}$ is a point all of whose components are active. If the gradient $g = g(x)$ has a nonzero component g_i at a nonactive index i, we may change x_i slightly without leaving the feasible region. The value of the objective function is reduced by moving slightly to smaller or larger values depending on whether $g_i > 0$ or $g_i < 0$, respectively. However, if x_i is active, only changes of x_i in one direction are possible without losing feasibility. The value of the objective function can then possibly be reduced by moving slightly in the feasible direction only when

$$\begin{cases} g_i \le 0 & \text{if } x_i = \underline{x}_i, \\ g_i \ge 0 & \text{if } x_i = \overline{x}_i. \end{cases} \tag{6}$$

But a decrease is guaranteed only if the slightly stronger condition

$$\begin{cases} g_i < 0 & \text{if } x_i = \underline{x}_i, \\ g_i > 0 & \text{if } x_i = \overline{x}_i \end{cases} \tag{7}$$

[1] In practice, one may allow a smaller domain of definition if f satisfies the **coercivity condition** that, as x approaches the boundary of the domain of definition, $f(x)$ exceeds the function value $f(x^0)$ at a known starting point x^0. Also, Lipschitz continuity may be relaxed to local Lipschitz continuity if all evaluation points remain in a bounded region.

holds. We say that the active index i and the corresponding variable x_i are **strongly active** if

$$\begin{cases} g_i > 0 & \text{if } x_i = \underline{x}_i, \\ g_i < 0 & \text{if } x_i = \overline{x}_i; \end{cases} \tag{8}$$

they are called **degenerate** otherwise. Thus slightly changing a single strongly active variable only cannot lead to a better feasible point.

2.1 Theorem. *(***Optimality conditions for bound-constrained optimization***)*
(i) **First order necessary conditions.** *At any local minimizer x of* (4)*, the* **reduced gradient** $g_{\text{red}}(x)$ *at x, with components*

$$g_{\text{red}}(x)_i := \begin{cases} 0 & \text{if } x_i = \underline{x}_i = \overline{x}_i, \\ \min(0, g_i(x)) & \text{if } x_i = \underline{x}_i < \overline{x}_i, \\ \max(0, g_i(x)) & \text{if } x_i = \overline{x}_i > \underline{x}_i, \\ g_i(x) & \text{otherwise,} \end{cases} \tag{9}$$

vanishes.

(ii) **First order sufficient conditions.** *Every corner x of* $\mathbf{x}$ *such that all variables are strongly active is a local minimizer of* (4)*.*

Proof. (i) Combining the various cases discussed above, we see that if the reduced gradient has a nonzero component, a decrease is always possible with a small feasible change. Thus the reduced gradient must vanish at a local optimizer.

(ii) In this case, any feasible point $x + \alpha p \neq x$ ($\alpha > 0$) must have $p_i \geq 0$ if $\underline{x}_i$ is active, $p_i \leq 0$ if $\overline{x}_i$ is active, and at least one p_i is nonzero. Therefore

$$g(x)^T p = \sum_i g_i p_i > 0.$$

This implies that $f(x + \alpha p) - f(x) = \alpha g(x)^T p + o(\alpha) > 0$ for small $\alpha > 0$, hence $f(x)$ is locally minimal. □

If no bound is active, $g_{\text{red}}(x) = g(x)$ and (i) reduces to the condition that x is a stationary point of the function f. In generalization of this, we call a feasible point x with $g_{\text{red}}(x) = 0$ a **stationary point** of the optimization problem (4). By the above, a local minimizer x of (4) must be a stationary point of this problem. This statement is a concise expression of the first order optimality conditions.

Note that the reduced gradient need not be continuous – it may change abruptly when a bound becomes active: In the simple 1-dimensional example

$$f(x) = x, \quad \mathbf{x} = [0, \infty], \tag{10}$$

we have $g_{\text{red}}(x) = 1$ for $x > 0$ but $g_{\text{red}}(0) = 0$. It is therefore important that a weaker continuity statement still holds, expressed in the first part of the following theorem.

2.2 Theorem. *If the sequence x^ℓ converges to $\widehat{x}$ and $\lim_{\ell\to\infty} g_{\rm red}(x^\ell) = 0$ then $g_{\rm red}(\widehat{x}) = 0$. Moreover, for every index $i = 1, \ldots, n$,*

$$g_i(\widehat{x}) > 0 \quad \Rightarrow \quad x_i^\ell = \widehat{x}_i = \underline{x}_i \text{ for sufficiently large } \ell, \tag{11}$$

$$g_i(\widehat{x}) < 0 \quad \Rightarrow \quad x_i^\ell = \widehat{x}_i = \overline{x}_i \text{ for sufficiently large } \ell. \tag{12}$$

Proof. Every free index i of $\widehat{x}$ is also free for x^ℓ with sufficiently large ℓ. Since f is continuously differentiable, we conclude that $g_i(\widehat{x}) = \lim_{\ell\to\infty} g_i(x^\ell) = 0$ for all free i. If $\widehat{x}_i = \underline{x}_i$ then the x_i^ℓ converge to $\underline{x}_i$, hence satisfy $x_i^\ell < \overline{x}_i$; thus $g_i(\widehat{x}) = \lim_{\ell\to\infty} g_i(x^\ell) \geq 0$ for these i. Similarly, one sees that $g_i(\widehat{x}) \leq 0$ if $\widehat{x}_i = \overline{x}_i$. Together, this implies $g_{\rm red}(\widehat{x}) = 0$.

Now let i be an index i for which $g_i(\widehat{x}) > 0$. We conclude from $g_{\rm red}(\widehat{x}) = 0$ that $\widehat{x}_i = \underline{x}_i < \overline{x}_i$. The definition (9) of the reduced gradient implies that for sufficiently large ℓ,

$$g_{\rm red}(x^\ell)_i = \begin{cases} 0 & \text{if } x_i^\ell = \underline{x}_i, \\ g_i(x^\ell) & \text{otherwise.} \end{cases}$$

Now $g_{\rm red}(x^\ell)$ converges to zero, but by continuity of the gradient, $\lim_{\ell\to\infty} g_i(x^\ell) = g_i(\widehat{x}) > 0$. Hence the second case is impossible for large ℓ. Therefore $x_i^\ell = \underline{x}_i$ for all large ℓ, and (11) holds for sufficiently large k.

Similarly, if i is an index for which $g_i(\widehat{x}) < 0$ then (12) holds for sufficiently large ℓ. □

A stationary point is called **degenerate** if $g_i(x) = 0$ for some active index i, and **non-degenerate** otherwise, i.e., if all its active bounds are strongly active. This allows us to rephrase Theorem 2.2 as saying that *all strongly active variables are ultimately fixed* when the sequence x^ℓ converges and $\lim_{\ell\to\infty} g_{\rm red}(x^\ell) = 0$. For degenerate activities no condition like (11) or (12) can be proved; cf. the second example in Subsection 2.4.

In particular, in the most typical case of convergence to a *nondegenerate* stationary point, zigzagging through changes of the active set (as in the examples of Section 2.4) cannot occur infinitely often provided we can prove that $\lim_{\ell\to\infty} g_{\rm red}(x^\ell) = 0$. There is no known way how to avoid zigzagging in the degenerate case.

2.3 Corollary. *If the $x^\ell \in \mathbf{x}$ form a bounded sequence such that $\inf \|g_{\rm red}(x^\ell)\|_* = 0$ then either some $\widehat{x} = x^\ell$ or some limit point $\widehat{x} \in \mathbf{x}$ of a subsequence satisfies $g_{\rm red}(\widehat{x}) = 0$.*

Proof. If $g_{\rm red}(x^\ell) = 0$ for some ℓ then $\widehat{x} = x^\ell$ works. Otherwise there is a subsequence of the $g_{\rm red}(x^\ell)$ converging to zero. Boundedness implies that this subsequence has a convergent subsequence, and by Theorem 2.2, its limit $\widehat{x}$ satisfies the claim. □

The corollary justifies to accept a numerical approximation x to a stationary point $\widehat{x}$ as soon as a stopping test of the form

$$\|g_{\rm red}(x)\|_* \leq \varepsilon \tag{13}$$

holds for some fixed ε. In this stopping test, one traditionally uses for $\|\cdot\|_*$ the maximum norm, with $\varepsilon = 10^{-5}$ or $\varepsilon = 10^{-6}$. In a conceptual analysis of algorithms, however, one has no stopping test and investigates the behavior of an infinite number of approximations x^ℓ, with the goal of showing that the $g_{\rm red}(x^\ell)$, or at least a subsequence of them, converge to zero. This implies (at least in exact arithmetic) finite termination if the stopping test (13) is added to the algorithm.

We define the **projection** $\mathrm{proj}(z, \mathbf{z})$ of a point $z \in \mathbb{R}^n$ to a box $\mathbf{z}$ to be the vector

$$\mathrm{proj}(z, \mathbf{z}) := \sup(\underline{z}_i, \inf(z_i, \overline{z}_i)) \tag{14}$$

with components

$$\mathrm{proj}(z, \mathbf{z})_i = \max(\underline{z}_i, \min(z_i, \overline{z}_i)) = \begin{cases} \underline{z}_i & \text{if } z_i \le \underline{z}_i, \\ \overline{z}_i & \text{if } z_i \ge \overline{z}_i, \\ z_i & \text{otherwise.} \end{cases}$$

It is easy to see that

$$\mathrm{proj}(z, \mathbf{z}) = z \iff z \in \mathbf{z}.$$

The shorthand notation

$$\pi[x] := \mathrm{proj}(x, \mathbf{x}) \tag{15}$$

is used for the special case where the box is the feasible set. Thus $\pi[x] \in \mathbf{x}$, and $\pi[x] = x$ iff $x \in \mathbf{x}$. With this notation, the first order optimality conditions may be written (for any $\alpha > 0$) in the equivalent form

$$g^{(\alpha)}(x) := \pi[x - \alpha g(x)] - x = 0.$$

$g^{(1)}(x)$ is called the **projected gradient** at x. $g^{(\alpha)}(x)$ is continuous in x for any α. However, Example (ii) of Section 2.4 shows that convergence to a stationary point $\widehat{x}$ is possible even when $\inf_\ell g_{\rm red}(x^\ell) > 0$, reflecting the lack of continuity of the reduced gradient. Such a counterintuitive situation means that infinitely many x^ℓ have activities different from the limiting $\widehat{x}$.

Traditional bound-constrained solvers aim at ensuring that the projected gradient has a subsequence converging to zero. This is a slightly weaker convergence statement but the resulting convergence analysis [3, 12, 22] is simpler. This is probably the reason why usually, e.g., in Byrd et al. [10] and in Hager & Zhang [30], a different stopping criterion of the form $\|g^{(1)}(x)\|_\infty \le \delta$ is used in place of (13) for a given threshold $0 < \delta < 1$. For example, for (10), $x = \delta$ satisfies this criterion although $g_{\rm red}(x) = 1$. However, in finite precision arithmetic, this stopping criterion may accept very poor points as sufficiently stationary. For example, for (10), $x = 10^{17}$ also satisfies this criterion although x is extremely far from a stationary point! In this particular case, the reason is that, in double precision arithmetic, $g^{(1)}(x)$ numerically becomes identically zero due to severe cancellation of digits in the subtraction.

2.4 Well-known active set strategies

To find optimal points of bound-constrained optimization problems, active set methods repeatedly perform two main phases. In the first phase, a good approximation to the set of optimal active constraints is determined by defining a face that is likely to contain a stationary point of the problem. In a second phase, this area of feasible domain is explored by solving an unconstrained subproblem approximately.

The projected conjugate gradient method of Polyak [40] is a classical reference for active set methods for bound-constrained problems with a convex quadratic objective function, for further references see [19, 20, 21, 36, 37, 45], and the gradient projection method of Bertsekas [3] is a classical reference for active set methods for bound-constrained problems with a general nonlinear objective function, for future references see [3, 12, 22].

Many researchers [10, 14, 13] have been interested in using gradient projections to identify optimal active constraints. To achieve fast convergence, they have applied Newton-type methods combined with the gradient projection method. Byrd et al. [10] determined the active variables by calculating the Cauchy point using the gradient projection method and then explored the subspace of nonactive variables by performing line search along limited memory BFGS directions [11]. Instead of limited memory BFGS directions, which cannot be used in a variety of applications, many researchers [4, 5, 6, 7, 8, 16, 17, 23, 35, 42] have used the Barzilai–Borwein step size [2] with simple structure and low computational cost to construct scaled steepest descent directions. Conn et al. [14, 13], Birgin & Martínez [5], and Rahpeymaii et al [41] combined active set strategies with trust region type methods. Another developed active set method was proposed by Hager & Zhang [30]. In this approach, a combination of the cyclic Barzilai–Borwein method [18] and a gradient projection method was used to determine the active variables, and then a Wolfe line search along the conjugate gradient directions [27, 28, 29, 31] was employed for solving an unconstrained subproblem. A two-stage approximate active set method was developed by Cristofari et al. [15]. In this method, unlike the other active set methods, an active set estimate was found such that the function value is reduced. Then, a truncated-Newton method in the subspace of nonactive variables was used to solve an unconstrained subproblem.

Bertsekas [3] and Conn et al. [13] showed that all strongly active variables were found and fixed after finitely many iterations. Although Hager & Zhang [30] used a restart procedure in the first and second phases of their active set method when the activity change or the reduction of the gradient in the components indexed by the working set is at least asymptotic over the reduction of the projected gradient $g^{(1)}$, they failed to show that active constraints are identified in a finite number of iterations. Instead, they showed that convergence can be achieved with a finite number of iterations when strict complementarity holds and all constraints are active at a stationary point. On the other hand, as described in Section 2.3, this projected gradient suffers from some drawbacks in finite precision arithmetic. Therefore, the activities may be incorrect.

Examples. We consider two examples for zigzagging in poor active set methods, cf. Figure 1:

(i) For the bound-constrained optimization problem

$$\begin{array}{ll} \min & \frac{1}{2}(x_1 - x_2)^2 + \varepsilon x_1 x_2 \\ \text{s.t.} & x_1 \geq 0, \quad x_2 \geq 0 \end{array}$$

with small $\varepsilon > 0$, we have

$$g(x) = \begin{pmatrix} x_1 - (1-\varepsilon)x_2 \\ x_2 - (1-\varepsilon)x_1 \end{pmatrix}.$$

Started with $x^0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$, the search directions

$$p^{2\ell} = (1-\varepsilon)^{2\ell} \begin{pmatrix} -1 \\ 1-\varepsilon \end{pmatrix}, \quad p^{2\ell+1} = (1-\varepsilon)^{2\ell+1} \begin{pmatrix} 1-\varepsilon \\ -1 \end{pmatrix}$$

are scaled steepest descent directions, and produce step size $\alpha = 1$ the sequence

$$x^{2\ell} = \begin{pmatrix} (1-\varepsilon)^{2\ell} \\ 0 \end{pmatrix}, \quad x^{2\ell+1} = \begin{pmatrix} 0 \\ (1-\varepsilon)^{2\ell+1} \end{pmatrix}, \tag{16}$$

with arbitrarily slow linear convergence to the solution at zero.
(ii) For the optimization of the bound-constrained problem

$$\begin{array}{ll} \min & x_1 + x_2 \\ \text{s.t.} & x_1 \geq 0, \quad x_2 \geq 0, \end{array}$$

started with the initial point and the descent directions $p^{2\ell}$ and $p^{2\ell+1}$ defined in the previous example, and the fixed step size $\alpha = 1$, we obtain the same descent sequence (16) with arbitrarily slow linear convergence to the zero solution. Moreover,

$$g_{\text{red}}(x^{2\ell}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad g_{\text{red}}(x^{2\ell+1}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

so that $g_{\text{red}}(x^\ell)$ does not converge to zero.

We now discuss how an active set method updates the working set for bound-constrained optimization. Each iteration changes only a subset of the variables. To account for this we use a **working set** $I \subseteq \{1, \ldots, n\}$ satisfying

$$q_i = 0 \quad \text{for } i \notin I, \tag{17}$$

and denote by q_I the subvector of q indexed by I. We write

$$g = g(x), \quad g_{\text{red}} = g_{\text{red}}(x).$$

Sensible choices for the working set I include the minimal set

$$I_-(x) := \{i \mid \underline{x}_i < x_i < \overline{x}_i\}, \tag{18}$$

containing only the free indices of x and the maximal set

$$\begin{array}{rcl} I_+(x) & := & I_-(x) \cup \{i \mid (g_{\rm red})_i \neq 0\} \\ & = & \{i \mid \underline{x}_i < x_i < \overline{x}_i \text{ or } \underline{x}_i = x_i < \overline{x}_i, g_i < 0 \text{ or } \underline{x}_i < x_i = \overline{x}_i, g_i > 0\}, \end{array} \tag{19}$$

containing all free and freeable indices. Here we call the index i **freeable** and say that the variable x_i can be **freed** from its bound if $i \in I_+(x) \setminus I_-(x)$. This is the case iff i is active and (7) holds. Indeed, for any active index i, (7) says that the ith components of the reduced gradient is nonzero, so that the function value decreases when moving the corresponding components x_i into the interior. By definition of the reduced gradient,

$$\|g_I(x)\|_* = \|g_{\rm red}(x)\|_* \quad \text{if } I = I_+(x). \tag{20}$$

Using always $I = I_+(x)$ may at first seem to be the most natural choice since it most quickly corrects a poor active set. However, this choice is prone to severe zigzagging, a major cause of inefficiency. To see this we consider the first example, where $I_\ell = I_+(x^\ell) = \{1, 2\}$ and (28) holds. Thus the choice $I = I_+(x)$ permits slow zigzagging.

We therefore need to control the conditions under which variables enter the working set I, to avoid that the same subset of variables is alternately freed and fixed in a large number of successive iterations. In the example, the choice $I = I_-(x)$ eliminates the zigzagging directions since $I_-(x^\ell)$ has size one. In the second zigzagging example, $g_{\rm red}(x^\ell)$ does not even converge to zero since

$$g_{\rm red}(x^{2\ell}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad g_{\rm red}(x^{2\ell+1}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}.$$

This is related to the fact that here $I_\ell = \{1, 2\}$, while $I_-(x^\ell) = I_+(x^\ell)$ has size one. Choosing $I = I_-(x)$ (in this example identical with $I = I_+(x)$) eliminates the zigzagging behavior. Thus $I = I_-(x)$ seems to be a good choice. However, we cannot always choose $I = I_-(x)$ since this might even be the empty set! But our examples indicate that an appropriate alternation between the choices $I = I_+(x)$ and $I = I_-(x)$ could eliminate zigzagging. This is indeed the case with a proper criterion for deciding between the two choices.

Zigzagging is thus a possible source of inefficiency. Good optimization methods should therefore be designed to eliminate zigzagging (caused by a bad active set method) as much as possible.

3 Our active set strategy

3.1 Search direction

As discussed in the introduction and Section 1, our search directions must satisfy three condition that allow `BOPT` to be convergent (Section 6). The first condition is (17) defined in Section 2.4. Here we discuss two other conditions.

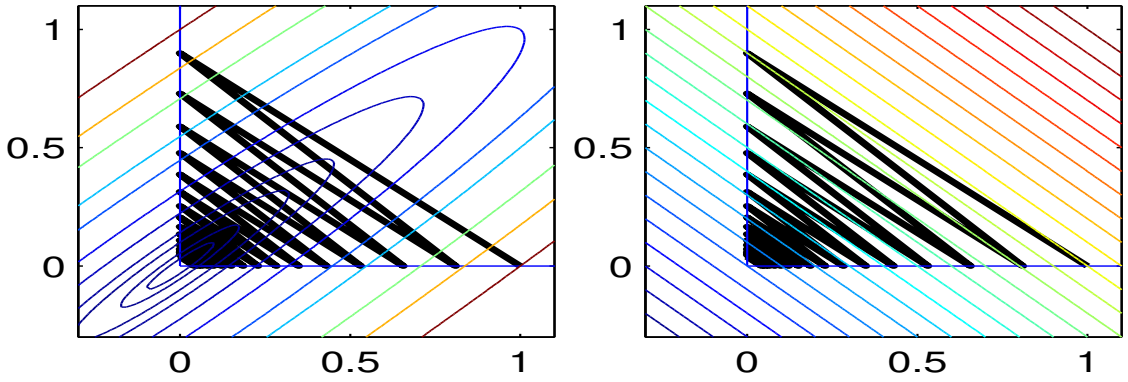

Figure 1: Inefficient zigzagging for convergence to a constrained minimizer in a corner: the example (i) (left) and the example (ii) (right).

In order to ensure local linear convergence of our new active set method when the working set I stays constant we require the bounded angle condition

$$\frac{g_I^T q_I}{\|g_I\|_* \|q_I\|} \leq -\delta < 0 \tag{21}$$

as the second necessary condition. In the following, $B \in \mathbb{R}^{n\times n}$ is a fixed but arbitrary symmetric, positive definite matrix, called the **preconditioner**. In practice, B is the identity matrix, a multiple of it, diagonal scaling matrix, or a matrix with the property that linear systems with coefficient matrix B are easy to compute. B is considered as a (more or less good) constant approximation of the Hessian matrix for the objective function. We may then apply the preceding with the ellipsoidal norms defined by (1). The **simplified Newton direction**

$$p_I := -B_{II}^{-1} g_I$$

satisfies the angle condition in the norms (1) with $\delta = 1$, hence leads to global convergence if used together with an efficient line search.

More generally, we may modify an arbitrary direction q by adding a multiple of the simplified Newton direction to get a direction

$$p_I := q_I - \lambda B_{II}^{-1} g_I \tag{22}$$

that satisfies the angle condition for a proper choice of the factor λ. Clearly it is enough to discuss the case $q_I \neq 0$. If

$$c := \frac{g_I^T q_I}{\sqrt{g_I^T B_{II}^{-1} g_I \cdot q_I^T B_{II} q_I}}$$

satisfies $c \leq -\delta$ we can take $\lambda = 0$ and $p_I := q_I$ satisfies the bounded angle condition. If this is not the case, we may use the following result.

3.1 Proposition. *Suppose that $g_I \neq 0$ and let $q_I \in \mathbb{R}^n \setminus \{0\}$, $0 < \delta < 1$. Put*

$$\pi_1 := g_I^T B_{II}^{-1} g_I > 0, \quad \pi_2 := q_I^T B_{II} q_I > 0, \quad \pi := g_I^T q_I. \tag{23}$$

Then

$$c = \frac{\pi}{\sqrt{\pi_1 \pi_2}} \in [-1, 1], \quad w := \frac{\pi_1 \pi_2 (1 - c^2)}{1 - \delta^2} \geq 0, \tag{24}$$

and (22) *satisfies the angle condition*

$$\frac{g_I^T p_I}{\sqrt{(g_I^T B_{II}^{-1} g_I)(p_I^T B_{II} p_I)}} = -\delta < 0 \tag{25}$$

when λ *is chosen as*

$$\lambda := \frac{\pi + \delta\sqrt{w}}{\pi_1}. \tag{26}$$

Proof. (24) follows directly from the generalized Cauchy–Schwarz inequality. In terms of π_i with $i = 1, 2$, the angle condition (25) reads

$$\frac{\pi - \lambda\pi_1}{\sqrt{\pi_1(\pi_2 - 2\lambda\pi + \lambda^2\pi_1)}} = -\delta. \tag{27}$$

Squaring, multiplying with the denominator, and subtracting $\delta^2(\pi - \lambda\pi_1)^2$ gives

$$\begin{aligned} (1 - \delta^2)(\pi - \lambda\pi_1)^2 &= \delta^2\pi_1(\pi_2 - 2\lambda\pi + \lambda^2\pi_1) - \delta^2(\pi - \lambda\pi_1)^2 \\ &= \delta^2(\pi_1\pi_2 - \pi^2) = \delta^2\pi_1\pi_2(1 - c^2). \end{aligned}$$

Since $\pi - \lambda\pi_1$ is negative by (27), we need $\pi - \lambda\pi_1 = -\delta\sqrt{w}$, hence (26). By construction, this choice indeed satisfies (27) and hence (25). □

In finite precision arithmetic, rounding errors occasionally result in a value of $c^2 > 1$. Therefore one should compute w from

$$w := \frac{\pi_1 \pi_2 \max\{\varepsilon, 1 - c^2\}}{1 - \delta^2},$$

where ε is the machine precision.

We need at one critical place the condition

$$g_i(x) q_i \leq 0 \quad \text{for all } i \quad \text{if } I = I_+(x) \neq I_-(x) \tag{28}$$

when some variable is freed as the third necessary condition. The conditions (17), (21), and (28) are satisfied with arbitrary I by directions of the form

$$q_i = \begin{cases} -g_i/d_i & \text{if } i \in I, \\ 0 & \text{otherwise} \end{cases} \tag{29}$$

with positive elements d_i in a fixed interval $[\underline{d}, \overline{d}]$, where $0 < \underline{d} < \overline{d} < \infty$.

3.2 The bent search path

For solving the bound-constrained optimization problem (4), a line search along a ray may lead to infeasible points. The most natural remedy, first suggested by Bertsekas [3], is to project the ray into the box. Thus we do each line search along a **bent search path**

$$x(\alpha) := \pi[x + \alpha q], \tag{30}$$

obtained by taking the ray $x + \alpha q$ ($\alpha \geq 0$) from the current point x into a direction $q \neq 0$ satisfying (17), (21), and (28) and projecting it into the feasible set using the projection (14).

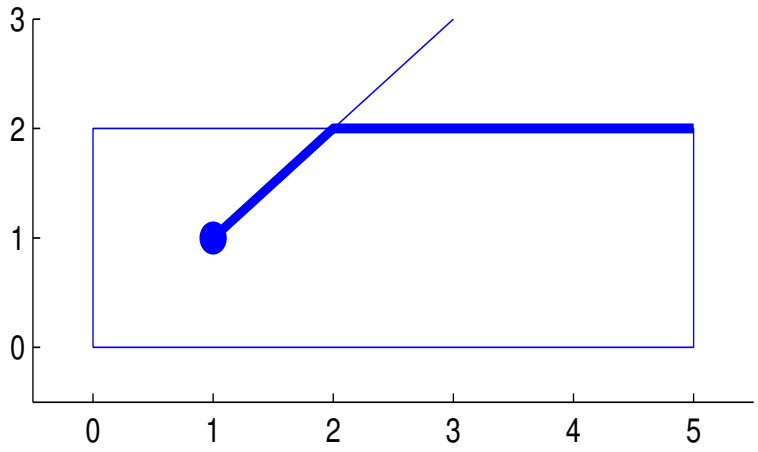

Figure 2: A bent search path $x(\alpha)$.

The bent search path is piecewise linear, with breakpoints at the elements of the set

$$S := \Big\{ \frac{\overline{x}_i - x_i}{q_i} \mid q_i > 0,\ x_i < \overline{x}_i < \infty \Big\} \cup \Big\{ \frac{\underline{x}_i - x_i}{q_i} \mid q_i < 0,\ x_i > \underline{x}_i > -\infty \Big\}.$$

If the breakpoints $\alpha_1, \ldots, \alpha_m$ are ordered such that

$$0 = \alpha_0 < \alpha_1 < \ldots < \alpha_m < \alpha_{m+1} = \infty,$$

the bent search path is linear on each interval $[\alpha_{i-1}, \alpha_i]$ ($i = 1, ..., m+1$). Note that when for some $\alpha > 0$, $x(\alpha)$ is a corner of the box then this corner is $x(\alpha_m)$ and $x(\alpha)$ stays constant for all $\alpha \geq \alpha_m$.

We use the curved line search `CLS` of Neumaier & Kimiaei [34] with the bent search path (30). `CLS` performs a simple bisection, process that finds a step size satisfying the **sufficient descent condition**

$$\mu(\alpha)|\mu(\alpha) - 1| \geq \beta \quad \text{with fixed } \beta > 0. \tag{31}$$

Here the **Goldstein quotient**

$$\mu(\alpha) := \frac{f(x(\alpha)) - f(x)}{\alpha g(x)^T p} \quad \text{for} \quad \alpha > 0, \tag{32}$$

first defined by GOLDSTEIN [25]. By [38, Theorem 1], the satisfaction of the condition (31) implies that there is a number $\delta' > 0$ such that

$$\frac{(f(x) - f(x(\alpha)))\|p\|^2}{(g(x)^T p)^2}; \geq \delta' \tag{33}$$

this result is essential for our convergence proof.

Since the bent path is piecewise linear, the line search can be improved a little by replacing each trial point α which is larger than α_1 by the nearest breakpoint interior to the bracket, as long as there is such a breakpoint.

4 The BOPT algorithm

In this section, we discuss our new active set algorithm which is against zigzagging arisen through poor search direction and poor active set strategy. Our algorithm updates the working set by one of two choices I_- and I_+ so that the size of reduced gradient is reduced.

We call any iteration where

$$I = I_+(x) \neq I_-(x)$$

a **freeing iteration**, since this is the condition that at least one bound is freed. In a freeing iteration one typically uses a search direction of the form (29), which guarantees the conditions required in the algorithm. In a non-freeing iteration, (28) is not a restriction, and one typically uses a search direction appropriate for an unconstrained method in the subspace defined by I, which, once the optimal activities are identified, leads to faster local convergence.

Since the goal is to decrease the size of the reduced gradient we need to ensure that shrinking the gradient in the components indexed by I shrinks the reduced gradient at least asymptotically. We therefore require (in an arbitrary monotone norm) the condition

$$\|g_{\text{red}}\|_*^2 \leq \rho \|g_I\|_*^2 \tag{34}$$

for some $\rho > 0$. This condition implies that the components of the reduced gradients missed by restricting to I are bounded by a multiple of $\|g_I\|_*$. By (20), (34) holds for $I = I_+(x)$ whenever $\rho > 1$; but a larger value of ρ is required that allows the choice $I = I_-(x)$ in examples where $I = I_+(x)$ leads to severe zigzagging. To find an appropriate condition on ρ we reconsider the first zigzagging example of Section 2.4, where this choice is necessary to eliminate zigzagging. When the 2-norm is used, (34) holds for this choice if

$$\rho > n. \tag{35}$$

The lower bound n (the number of variables) is needed in order to account for related n-dimensional examples of zigzagging with $n - 1$ freeable bounds. For the maximum norm $\|\cdot\|_*$, corresponding to the 1-norm $\|\cdot\|$, any positive $\rho < 1$ would suffice in these examples to allow $I = I_-(x)$, which eliminates zigzagging.

The algorithm guarantees that before every freeing iteration an efficient line search is done that does not result in a new activity. By (20) and the stopping test, we have

$$\|g_I(x)\|_* = \|g_{\rm red}(x)\|_* > 0 \quad \text{if } I = I_+(x). \tag{36}$$

In particular if (34) fails for $I = I_-(x)$, the resetting of I ensures that (34) holds for the working set I used in the next iteration. Therefore (34) holds at every iteration except possibly the first.

Taking into account these insights we propose the following algorithmic scheme, for which convergence and strong limitations on the possible forms of zigzagging will be proved.

Algorithm 1 `BOPT`, **bound-constrained optimization**

Purpose: `BOPT` minimizes a smooth $f(x)$ subject to $x \in \mathbf{x} = [\underline{x}, \overline{x}]$.

Input: $x^0 \in \mathbb{R}^n$ (starting point).

Tuning parameters: $\beta \in]0, \frac{1}{4}[$, $q > 1$ (line search parameters), $0 < \delta < 1$ (reduced angle parameters), $0 < \rho < 1/n$ (factor safeguarding (34)), and parameters specifying a pair of monotone dual norms.

Set $x = x^0$, $f = f(x)$, $g = g(x)$, $I = I_+(x)$, and `freeing` $= 0$;
while $g_{\rm red}(x) \neq 0$ **do**
    update x and f by running `CLS` with (30) along q satisfying (17), (21), and (28);
    choose $I = I_-(x)$; ▷ minimal set was chosen
    evaluate `freeing`=((34) fails);
    if `freeing` **then**, choose $I = I_+(x)$; **end**; ▷ maximal set was chosen
    compute $g = g(x)$;
end while
return x and f;

A suitable starting point is $x^0 := \pi[0]$, the point in $\mathbf{x}$ with smallest norm. If the solution $\widehat{x}$ of a previously solved related problem is available, $x^0 := \pi[\widehat{x}]$ may be used, which usually reduces the number of iterations needed.

5 Some auxiliary results

We now prove a few technical results that are needed for our convergence proof in the next section.

5.1 Proposition. *For every $x \in \mathbf{x}$, nonzero q, and $\alpha > 0$,*

$$p_q(\alpha) := \frac{\pi[x + \alpha q] - x}{\alpha}$$

satisfies (in any monotone norm)

$$|p_q(\alpha)| \leq |q|, \quad \|p_q(\alpha)\| \leq \|q\|, \tag{37}$$

and with $p \in \mathbb{R}^n$ defined by

$$p_i := \begin{cases} 0 & \text{if } \underline{x}_i = x_i = \overline{x}_i, \\ \max(0, q_i) & \text{if } \underline{x}_i = x_i < \overline{x}_i, \\ \min(q_i, 0) & \text{if } \underline{x}_i < x_i = \overline{x}_i, \\ q_i & \text{if } \underline{x}_i < x_i < \overline{x}_i, \end{cases} \tag{38}$$

we have

$$p(\alpha) = p \quad \text{for } 0 < \alpha \le \alpha_1. \tag{39}$$

Proof. We have

$$\alpha p_q(\alpha) = \pi[x + \alpha q] - x = \sup\big(\underline{x}, \inf(x + \alpha q, \overline{x})\big) - x = \sup\big(\underline{x} - x, \inf(\alpha q, \overline{x} - x)\big),$$

hence

$$p_q(\alpha) = \sup\Big(\frac{\underline{x} - x}{\alpha}, \inf\Big(q, \frac{\overline{x} - x}{\alpha}\Big)\Big) = \operatorname{proj}(q, \mathbf{x}(\alpha)),$$

where

$$\mathbf{x}(\alpha) := \Big[\frac{\underline{x} - x}{\alpha}, \frac{\overline{x} - x}{\alpha}\Big] = (\mathbf{x} - x)/\alpha.$$

Since $(\underline{x} - x)/\alpha \le 0$, $|p_q(\alpha)| = |\operatorname{proj}(q, \mathbf{x}(\alpha))| \le |q|$ holds. Further, (37) follows due to the fact that the norm is monotone.

For $0 < \alpha \le \alpha_1$, the bent search path is linear, hence

$$p_q(\alpha) = x + \alpha p$$

for some vector p. This gives (39). For $\alpha \to 0$, the boxes $\mathbf{x}(\alpha)$ have a well-defined limit $\mathbf{x}(0)$ whose components have the bounds

$$\underline{x}(0)_i = \lim_{\alpha \to 0} \frac{\underline{x}_i - x_i}{\alpha} = \begin{cases} 0 & \text{if } x_i = \underline{x}_i, \\ -\infty & \text{otherwise,} \end{cases}$$

$$\overline{x}(0)_i = \lim_{\alpha \to 0} \frac{\overline{x}_i - x_i}{\alpha} = \begin{cases} 0 & \text{if } x_i = \overline{x}_i, \\ +\infty & \text{otherwise.} \end{cases}$$

Therefore

$$p = \lim_{\alpha \to 0} p_q(\alpha) = \lim_{\alpha \to 0} \operatorname{proj}(q, \mathbf{x}(\alpha)) = \operatorname{proj}(q, \mathbf{x}(0)).$$

Expressed in components we find (38). □

5.2 Proposition. *If the index set $I \subseteq I_+(x)$ satisfies* (17) *and* (21) *then*

$$g^T q = g_I^T q_I < 0, \quad \|q\| = \|q_I\|. \tag{40}$$

If, in addition,

$$g_i(x) q_i \le 0 \quad \text{for all } i \tag{41}$$

then

$$p = q, \tag{42}$$

and we have $f(x(\alpha)) < f(x)$ for sufficiently small $\alpha > 0$.

Proof. By (17) and (21),

$$g^T q = \sum_i g_i q_i = \sum_{i \in I} g_i q_i = g_I^T q_I < 0,$$

giving (40). Since $I \subseteq I_+(x)$, any active i satisfies one of the conditions in (41). Thus if (41) holds then (38) gives (42). The piecewise linear structure of the search path now gives $x(\alpha) = x + \alpha p_q(\alpha) = x + \alpha q$ for all sufficiently small $\alpha > 0$, and therefore

$$f(x(\alpha)) = f(x + \alpha q) = f(x) + \alpha g^T q + o(\alpha) = f(x) + \alpha(g^T q + o(1)) < f(x)$$

for sufficiently small $\alpha > 0$. □

5.3 Proposition. *Suppose that*

$$g_i(x^\ell) q_i^\ell \le 0 \quad \textit{for } i \in I_+(x^\ell), \tag{43}$$

$$q_i^\ell = 0 \quad \textit{for } i \notin I_+(x^\ell).$$

If

$$\lim_{\ell \to \infty} x^\ell = x, \quad \lim_{\ell \to \infty} \alpha_\ell = 0, \quad \lim_{\ell \to \infty} q^\ell = q$$

then

$$r^\ell := \frac{\pi[x^\ell + \alpha_\ell q^\ell] - x^\ell}{\alpha_\ell \|q^\ell\|}$$

satisfies $\lim_{\ell \to \infty} r^\ell = q$.

Proof. We first simplify the assumptions by replacing q^ℓ with $q^\ell / \|q^\ell\|$ and α_ℓ with $\alpha_\ell \|q^\ell\|$. Then the assumptions on the q^ℓ and α_ℓ take the form

$$\|q^\ell\| = 1, \quad r^\ell = \frac{\pi[x^\ell + \alpha_\ell q^\ell] - x^\ell}{\alpha_\ell}, \quad \alpha_\ell > 0 \quad \text{for all } \ell,$$

$$\lim_{\ell \to \infty} q^\ell = q, \quad \lim_{\ell \to \infty} \alpha_\ell = 0,$$

By Proposition 5.1, $|r^\ell| \le |q^\ell|$, and by assumption, $r_i^\ell = q_i^\ell = 0$ for $i \notin I_+(x^\ell)$. Since the q^ℓ are bounded and $\alpha_\ell \to 0$, Proposition 5.1 also implies that for sufficiently large ℓ,

$$r_i^\ell := \begin{cases} 0 & \text{if } \underline{x}_i = x_i^\ell = \overline{x}_i, \\ \max(0, q_i^\ell) & \text{if } \underline{x}_i = x_i^\ell < \underline{x}_i, \\ \min(q_i^\ell, 0) & \text{if } \underline{x}_i < x_i^\ell = \overline{x}_i, \\ q_i^\ell & \text{if } \underline{x}_i < x_i^\ell < \overline{x}_i. \end{cases}$$

In view of (43), this implies $r_i^\ell = q_i^\ell$ for $i \in I_+(x^\ell)$ and sufficiently large ℓ. Taking the limit, we find $r^\ell \to q$, as claimed. □

6 Convergence

6.1 Theorem. *Let f be continuously differentiable in the box $\mathbf{x}$, with Lipschitz continuous gradient g. Let x^ℓ denote the value of x in Algorithm 1 after its ℓth update. Then one of the following three cases holds:*

(i) $g_{\mathrm{red}}(x^\ell) = 0$ *for some* ℓ.

(ii) We have

$$\lim_{\ell\to\infty} f(x^\ell) = \widehat{f} \in \mathbb{R}, \quad \inf_{\ell\geq 0} \|g_{\mathrm{red}}(x^\ell)\|_* = 0.$$

Some limit point $\widehat{x}$ *of the* x^ℓ *satisfies* $f(\widehat{x}) = \widehat{f} \leq f(x^0)$ *and* $g_{\mathrm{red}}(\widehat{x}) = 0$.

(iii) $\sup_{\ell\geq 0} \|x^\ell\| = \infty$.

Proof. We may assume that infinitely many iterations occur and the x^ℓ are bounded, since otherwise (i) or (iii) hold by the stopping test. Suppose that

$$\inf \|g_{\mathrm{red}}(x^\ell)\|_* = 0. \tag{44}$$

Since the x^ℓ are bounded there is a convergent subsequence x^{ℓ_k} with $\|g_{\mathrm{red}}(x^{\ell_k})\|_* \to 0$. Then due to Corollary 2.3, the limit $\widehat{x} \in \mathbf{x}$ satisfies $g_{\mathrm{red}}(\widehat{x}) = 0$. Hence (ii) holds. Thus it remains to show that (44) holds.

For the point x, the working set I, the direction q, and the tangent direction p given by (38) at iteration ℓ before updating I, we write x^ℓ, I_ℓ, q^ℓ, and p^ℓ, respectively. Since function values decrease monotonically by construction, the infimum $\widehat{f}$ of the $f(x^\ell)$ is finite, and we have

$$\lim_{\ell\to\infty} f(x^\ell) = \widehat{f}. \tag{45}$$

For any index set I, we consider the set L_I of indices ℓ satisfying

$$I = I_\ell = I_+(x^\ell) \neq I_-(x^\ell)$$

and distinguish several cases, depending on the amount of zigzagging.

Case 1 (limited zigzagging): All L_I are finite. Since every ℓ for which the ℓth iteration is freeing belongs to some L_I and there are only finitely many possibilities for I, the number of freeing iterations is finite. Thus there is a number N_f such that no iteration with index $\ell > N_f$ is freeing. Algorithm 1 and (34) imply that $I_\ell = I_-(x^\ell)$ for $\ell > N_f$. Therefore a line search in iteration $\ell > N_f$ never frees an already active bound; hence bounds can only be fixed. This can happen only finitely many times; so there is an N such that $I_-(x^\ell)$ remains fixed for all $\ell > N$,

$$I_\ell = I_-(x^\ell) = I \quad \text{for } \ell > N, \tag{46}$$

and no bound is fixed for $\ell > N$. Therefore the line search accepts a step size $\alpha < \alpha_1$, so that $p(\alpha) = p$ has components (38). Inserting (39) into (33) results in

$$\frac{(f(x^\ell) - f(x^{\ell+1}))\|p^\ell\|^2}{(g(x^\ell)^T p^\ell)^2} \geq \delta' > 0 \quad \text{for all } \ell > N.$$

(17) and (21) hold by the specification of Algorithm 1, and (40) follows by Proposition 5.2. Using (21), (40), and (34) (the latter holds by the remark after Algorithm 1), we find that for all $\ell > N$,

$$\begin{aligned} f(x^\ell) - f(x^{\ell+1}) &\geq \delta'\Big(\frac{g(x^\ell)^T p^\ell_{I_\ell}}{\|p^\ell_{I_\ell}\|}\Big)^2 = \delta'\Big(\frac{g_{I_\ell}(x^\ell)^T p^\ell_{I_\ell}}{\|p^\ell_{I_\ell}\|}\Big)^2 \\ &\geq \delta'\Big(\delta\|g_{I_\ell}(x^\ell)\|_*\Big)^2 \geq \delta'\Big(\delta\rho^{-1}\|g_{\mathrm{red}}(x^\ell)\|_*\Big)^2 \geq \Delta := \delta'(\delta\rho^{-1}\gamma^*)^2, \end{aligned}$$

where

$$\gamma^* := \inf_{\ell \geq 0} \|g_{\mathrm{red}}(x^\ell)\|_*. \tag{47}$$

For $\ell \to \infty$, (45) implies that the left hand side tends to zero, hence $\Delta = 0$ and therefore $\gamma^* = 0$. Hence (44) holds and we are done.

Case 2 (unlimited zigzagging): Some L_I is an infinite set. Handling this case requires a detailed look at what happens at the bounds. Since all conditions used in Algorithm 1 and `CLS` are invariant under appropriate scaling we may assume w.l.o.g. that all directions q^ℓ are scaled such that

$$\|q^\ell\| = 1. \tag{48}$$

According to Algorithm 1, (21) and (28) hold. (28) and Proposition 5.2 imply (40) and

$$p^\ell = q^\ell \quad \text{for } \ell \in L_I. \tag{49}$$

If the x^ℓ are unbounded, (iii) holds.

Otherwise the set of tuples $[x^\ell, q^\ell]$ is bounded. Thus there is an infinite sequence $\ell_k \in L_I$ $(k = 1, 2, \ldots)$ such that, for $k \to \infty$,

$$x^{\ell_k} \to \widehat{x}, \quad q^{\ell_k} \to q, \quad g^{\ell_k} \to \widehat{g} := g(\widehat{x}).$$

Using (45), we find $f(\widehat{x}) = \widehat{f}$, and we have

$$\|q\| = 1, \quad q_i = 0 \quad \text{for } i \notin I. \tag{50}$$

We first handle the special case $g_I = 0$ and afterwards the nontrivial case $g_I \neq 0$.

Case 2a: $\widehat{g}_I = 0$. By (20),

$$g_I(x^\ell) = g_{I_\ell}(x^\ell) = g_{\mathrm{red}}(x^\ell) \quad \text{for } \ell \in L_I.$$

Since L_I is infinite, $\widehat{g}_I = 0$ implies that (44) holds and we are done.

CASE 2B: $\widehat{g}_I \neq 0$. Taking limits in (17), (40), and (21) gives $\|q_I\| = \|q\| = 1$ and

$$\widehat{g}^T q = \widehat{g}_I^T q_I \leq -\delta \|\widehat{g}_I\|_* < 0. \tag{51}$$

We write α_ℓ, and μ_ℓ for the step size α chosen by the line search and the Goldstein quotient μ at iteration ℓ,

$$\mu_\ell := \mu(\alpha_\ell) = \frac{f(x^{\ell+1}) - f(x^\ell)}{\alpha_\ell g(x^\ell)^T p^\ell}. \tag{52}$$

Since the accepted step size satisfies the descent condition (31),

$$\mu_\ell |\mu_\ell - 1| \geq \beta > 0. \tag{53}$$

Hence μ_ℓ is bounded away from 0 and 1. Since $g(x^{\ell_k})^T p^{\ell_k} \to g^T q \neq 0$ by (51), we find from (45) that

$$\alpha_{\ell_k} = \frac{f(x^{\ell_k+1}) - f(x^{\ell_k})}{\mu_{\ell_k} g(x^{\ell_k})^T q^{\ell_k}} \to 0 \quad \text{for } k \to \infty. \tag{54}$$

Now Proposition 5.3 applies since by (17), $q_i^\ell = 0$ for $i \notin I$. Using (48), we therefore find that, by definition of r^ℓ,

$$x^{\ell_k+1} = \pi[x^{\ell_k} + \alpha_{\ell_k} q^{\ell_k}] = x^{\ell_k} + \alpha_{\ell_k} r^{\ell_k}, \quad r^{\ell_k} \to q.$$

Taylor expansion gives

$$f(x^{\ell_k+1}) = f(x^{\ell_k} + \alpha_{\ell_k} r^{\ell_k}) = f(x^{\ell_k}) + \alpha_{\ell_k} g(x^{\ell_k})^T r^{\ell_k} + o(\alpha_{\ell_k}) \quad \text{for } \ell_k \in L_I.$$

Comparing with (52), we find

$$\mu_{\ell_k} = \frac{f(x^{\ell_k+1}) - f(x^{\ell_k})}{\alpha_{\ell_k} g(x^{\ell_k})^T q^{\ell_k}} = \frac{g(x^{\ell_k})^T r^{\ell_k} + o(1)}{g(x^{\ell_k})^T q^{\ell_k}} \to 1.$$

Since this contradicts (53), and thus $\widehat{g}_I = 0$. □

6.2 Remark. If for a given initial iterate x^0, the set $\{x \in \mathbf{x} : f(x) \leq f(x^0)\}$ is bounded, and in particular if $\mathbf{x}$ is bounded, the sequence x^ℓ is bounded, so that (i) or (ii) holds. We conjecture that when neither (i) or (ii) holds then $f_\ell \to -\infty$.

The typical situation is that there is only one limit point $\widehat{x}$, so that $x^\ell \to \widehat{x}$. In exact arithmetic, the stationary points found are usually local minimizers as convergence of a subsequence to a nonminimizing stationary point is unstable under arbitrarily small generic perturbations. Thus one usually converges to a single local minimizer. In finite precision, one typically ends up anywhere in a region where the reduced gradient is dominated by noise due to rounding errors, so that the theory (which assumes exact arithmetic) no longer gives a reliable description of the finite precision behavior. This may in particular happen in very flat regions of the feasible domain where there is no nearby stationary point; numerical misconvergence is then possible. However, all optimization methods using only function values and gradients necessarily face this kind of difficulties.

Theorem 2.2 says that in case of convergence to a nondegenerate stationary point, all strongly active variables are ultimately fixed. Thus zigzagging through changes of the active set (as in the examples of Section 2.4) cannot occur infinitely often. For degenerate variables, our results assert nothing. However, (35) – though not used in the proof of Theorem 6.1 – excludes zigzagging in the degenerate the example of NEUMAIER et al. [38, Section 3.3]. Thus it might be possible to prove more in the degenerate case.

7 Discussion and conclusion

In this paper, we described the theoretical properties of `BOPT` for bound-constrained optimization problems whose objective function is continuously differentiable with a Lipschitz continuous gradient. `BOPT` uses a new active set strategy against zigzagging. Unlike other active set strategies, `BOPT` uses the reduced gradient in our active set strategy instead of the projected gradient, which does not always work in finite precision arithmetic, and ensures that the size of the reduced gradient becomes asymptotically small.

When the search directions q satisfy the conditions (17), (21), and (28), we prove global convergence. We also show that after finitely many iterations, `BOPT` finds and fixes all strongly active variables, similarly to BERTSEKAS [3] and CONN et al. [13].

`BOPT` was implemented by KIMIAEI et al. [32] in `LMBOPT` for particular choices of the search directions. For all freeing iterations, `LMBOPT` performs `CLS` with the bent search path (30) along the directions q computed by (29), while for all non-freeing iterations, it uses `CLS` with (30) along limited memory quasi-Newton types directions and the nonlinear conjugate gradient directions proposed by NEUMAIER et al. [38] for exploring the subspace of nonactive variables. All directions used in `LMBOPT` satisfy conditions (17), (21), and (28), and as a consequence, the theory discussed in this paper is valid for `LMBOPT`.

To illustrate the numerical efficiency of the `BOPT` framework, we report here from [32] numerical results that compare `LMBOPT` to other unconstrained and bound-constrained state-of-the-art solvers[2] applied to the bound-constrained test problems from the `CUTEst` collection [26]. These results are gathered in Table 1. We denote by `sec` the time in seconds, by `nf` the number of function evaluations, by `ng` the number of gradient evaluations, and write `nf2g` := `nf` + 2`ng`. Each algorithm was terminated once one of the termination criteria given in the first row of Table 1 was satisfied. These impose upper bounds on $\|g_{\text{red}}\|_\infty$, `sec`, and `nf2g`. For a given list S of solvers and each given cost measure c_s, the **efficiency**

$$e_s := \begin{cases} (\min_{\bar{s}\in S} c_{\bar{s}})/c_s & \text{if the solver } s \text{ solves the problem,} \\ 0 & \text{otherwise} \end{cases}$$

of the solver s measures the strength of the solver s relative to an ideal solver corresponding to the best solver for each problem in percent, rounded to integers. The other columns of

[2]In fact, the solvers `LMBFG-EIG-MS`, `LMBFG-DDOGL`, `LMBFG-BWX-MS`, `LMBFG-EIG-curve-inf`, `LMBFG-EIG-MS-2-2`, `CGdescent`, `LMBFG-EIG-inf-2`, `LMBFGS-TR`, `LMBFG-MTBT`, `LMBFG-MT` are unconstrained optimization solvers. In order to provide a comprehensive numerical report, we turned them into bound-constrained solvers. For each solver, this modification was done by combining the bent search path (30) with the corresponding line search strategy.

Table 1: The summary results for all problems

stopping test: $\|g_{\rm red}\|_\infty \leq 10^{-6}$, sec ≤ 300, nf2g $\leq 20n + 10^3$					
433 of 473 problems solved		mean efficiency in %			
dim∈[1,100001]		for cost measure			
solver	solved	nf2g	ng	nf	sec
LMBOPT [32]	**417**	53	**65**	38	13
ASACG [30]	402	53	54	48	**60**
LMBFG-EIG-MS [9]	402	57	57	55	34
LMBFG-DDOGL [9]	399	59	58	57	34
ASABCP [15]	395	41	38	43	50
LMBFG-BWX-MS [9]	395	47	43	55	32
LMBFG-EIG-curve-inf [9]	394	57	57	55	34
LMBFG-EIG-MS-2-2 [9]	381	44	41	51	32
SPG [8]	377	38	38	36	12
CGdescent [29]	365	44	47	40	50
LBFGSB [10]	354	**64**	60	**67**	43
LMBFG-EIG-inf-2 [9]	239	36	36	35	23
LMBFGS-TR [9]	208	32	31	31	23
LMBFG-MTBT [9]	185	30	29	30	18
LMBFG-MT [9]	180	28	25	30	21

the table contain the number of solved problems by the solvers, the nf2g efficiency, the ng efficiency, the nf efficiency, and the sec efficiency.

Since LMBOPT was able to solve 417 out of 473 bound-constrained test problems, it appears to be more robust than the other solvers. That CLS with the bent search path (30) is a gradient-free line search, explains that LMBOPT has the highest gradient efficiency. Thus, LMBOPT is highly recommended for problems with expensive gradient evaluations.

References

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

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.* **8** (198), 141–148.

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

[4] E. G. Birgin, I. Chambouleyron, and J. M. Martínez. Estimation of the optical constants and thickness of thin films using unconstrained optimization. *J. Comput. Phys.* **151** (1999), 862–880.

[5] E. G. Birgin and J. M. Martínez. A box-constrained optimization algorithm with negative curvature directions and spectral projected gradients. In *Topics in Numerical Analysis* (G. Alefeld and X. Chen, eds.), Vol. 15 of *Computing Supplementa*, pp. 49–60. Springer Vienna (2001).

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

[7] E. G. Birgin, J. M. Martínez, and M. Raydan. Nonmonotone spectral projected gradient methods on convex sets. *SIAM J. Optim.* **10** (1999), 1196–1211.

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

[9] O. Burdakov, L. Gong, S. Zikrin, and Y. Yuan. On efficiently combining limited-memory and trust-region techniques. *Math. Program. Comput.* **9** (2017), 101–134.

[10] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound-constrained optimization. *SIAM J. Sci. Comput.* **16** (1995), 1190.

[11] R. H. Byrd, J. Nocedal, and R. B. Schnabel. Representations of quasi-newton matrices and their use in limited memory methods. *Math. Program.* **63** (1994), 129–156.

[12] P. Calamai and J. Moré. Projected gradient methods for linearly constrained problems. *Math. Program.* **39** (1987), 93–116.

[13] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.* **25** (1988), 433.

[14] A. R. Conn, N. I. M. Gould, and Ph. L Toint. Testing a class of methods for solving minimization problems with simple bounds on the variables. *Mathematics of Computation* **50** (1988), 399–430.

[15] A. Cristofari, M. De Santis, S. Lucidi, and F. Rinaldi. A two-stage active-set algorithm for bound-constrained optimization. *J. Optim. Theory Appl.* **172** (2017), 369–401.

[16] Y. H. Dai and R. Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.* **100** (2005), 21–47.

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

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

[19] R. S. Dembo and U. Tulowitzki. On the minimization of quadratic functions subject to box constraints. Technical report, School of Organization and Management, Yale University, New Haven, CT (1983).

[20] Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM J. Optim* **7** (1997), 871–887.

[21] Z. Dostál. A proportioning based algorithm with rate of convergence for bound constrained quadratic programming. *Numer. Algorithms* **34** (2003), 293–302.

[22] J. C. Dunn. On the convergence of projected gradient processes to singular critical points. *J. Optim. Theory Appl.* **55** (1987), 203–216.

[23] W. Glunt, T. L. Hayden, and M. Raydan. Molecular conformations from distance matrices. *J. Comput. Chem.* **14** (1993), 114–120.

[24] A. Goldstein and J. Price. An effective algorithm for minimization. *Numer. Math.* **10** (1967), 184–189.

[25] A. A. Goldstein. On steepest descent. *J. SIAM, Ser. A: Control* **3** (1965), 147–151.

[26] N. I. M. Gould, D. Orban, and Ph. L. Toint. CUTEst: a constrained and unconstrained testing environment with safe threads for mathematical optimization. *Comput. Optim. Appl.* **60** (2015), 545–557.

[27] W. W. Hager and H. Zhang. CG_DESCENT user's guide. Technical report, Department of Mathematics, University of Florida, Gainesville, FL (2004).

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

[29] W. W. Hager and H. Zhang. Algorithm 851: CG_DESCENT, a conjugate gradient method with guaranteed descent. *ACM Trans. Math. Softw.* **32** (2006), 113–137.

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

[31] W. W. Hager and H. Zhang. A survey of nonlinear conjugate gradient methods. *Pac. J. Optim.* **2** (2006), 35–58.

[32] M. Kimiaei, A. Neumaier, and B. Azmi. LMBOPT – A limited memory method for bound-constrained optimization. *Math. Program. Comput.* **14** (2022), 271–318.

[33] A. Neumaier and B. Azmi. Line search and convergence in bound-constrained optimization. Unpublished manuscript, University of Vienna (2019). `http://www.optimization-online.org/DB_HTML/2019/03/7138.html`.

[34] A. Neumaier and M. Kimiaei. An efficient gradient-free line search. Preprint, University of Vienna (2022). `https://optimization-online.org/?p=21115`

[35] W. Liu and Y. H. Dai. Minimization algorithms based on supervisor and searcher cooperation. *J. Optim. Theory Appl.* **111** (2001), 359–379.

[36] J. J. Moré and G. Toraldo. Algorithms for bound-constrained quadratic programming problems. *Numer. Math.* **55** (1989), 377–400.

[37] J. J. Moré and G. Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optim.* **1** (1991), 93–113.

[38] A. Neumaier, M. Kimiaei, and B. Azmi. Nonlinear conjugate gradients without wolfe line search. Preprint, Vienna University, Fakultät für Mathematik, Universität Wien, Oskar-Morgenstern-Platz 1, A-1090 Wien, Austria (2022).

[39] J. Nocedal and S. J. Wright, eds. *Numerical Optimization*. Springer-Verlag (1999).

[40] B. T. Polyak. The conjugate gradient method in extremal problems. *USSR Comput. Math. Math. Phys.* **9** (1969), 94–112.

[41] F. Rahpeymaii, M. Kimiaei, and A. Bagheri. A limited memory quasi-newton trust-region method for box constrained optimization. *J. Comput. Appl. Math.* **303** (September 2016), 105–118.

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

[43] W. Warth and J. Werner. Effiziente Schrittweitenfunktionen bei unrestringierten Optimierungsaufgaben. *Computing* **19** (1977), 59–72.

[44] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.* **11** (1969), 226–235.

[45] E. K. Yang and J. W. Tolle. A class of methods for solving large, convex quadratic programs subject to box constraints. *Math. Program.* **51** (1991), 223–228.