Discontinuous piecewise linear optimization

Andrew R. Conn*
T. J. Watson Research Center, PO Box 218, Yorktown Heights, NY 10598

Marcel Mongeau†
Centre de Recherches Mathématiques, Université de Montréal, Canada H3C 3J7

Abstract

A theoretical framework and a practical algorithm are presented to solve discontinuous piecewise linear optimization problems dealing with functions for which the *ridges* are known. A penalty approach allows one to consider such problems subject to a wide range of constraints involving piecewise linear functions. Although the theory is expounded in detail in the special case of discontinuous piecewise *linear* functions, it is straightforwardly extendable, using standard nonlinear programming techniques, to *nonlinear* (discontinuous piecewise differentiable) functions.

The descent algorithm which is elaborated uses active-set and projected gradient approaches. It is a generalization of the ideas used by Conn to deal with nonsmoothness in the l_1 exact penalty function, and it is based on the notion of *decomposition* of a function into a smooth and a nonsmooth part. The constrained case is reduced to the unconstrained minimization of a (piecewise linear) l_1 exact penalty function. We also discuss how the algorithm is modified when it encounters degenerate points. Preliminary numerical results are presented: the algorithm is applied to discontinuous optimization problems from models in industrial engineering.

Key words: Discontinuous optimization, non-differentiable optimization, piecewise-linear programming, active-set approach, projected-gradient direction, degeneracy, exact penalty method.

---

*This research was supported in part by the Advanced Research Projects Agency of the Department of Defense and was monitored by the Air Force Office of Scientific Research under Contract No F49620-91-C-0079. The United States Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon.

†Supported by the Natural Sciences and Engineering Council of Canada and the Centre de Recherches Mathématiques, Université de Montréal. Current address: Laboratoire Approximation & Optimisation, Université Paul Sabatier, 31062 Toulouse cedex 04, France

Contents

1 Introduction

We consider the problem:

$$\begin{array}{rrcl} \inf & \tilde{f}(x) & & \\ \text{subject to} & f_i(x) & = & 0,\ i \in E \\ & f_i(x) & \geq & 0,\ i \in I, \end{array} \tag{1}$$

where the index sets E and I are finite and disjoint and $\tilde{f}$ and f_i, $i \in E \cup I$ are a collection of (possibly discontinuous) *piecewise linear* functions that map $I\!R^n$ to $I\!R$. A piecewise linear function f is a function whose derivative is defined everywhere except over a subset of a finite number of hyperplanes, and these hyperplanes partition the domain into subdomains over each of which f is linear. A *ridge* is a specified hyperplane

$$\{x \in I\!R^n : a^T x = b\},\ a \in I\!R^n,\ b \in I\!R,$$

containing a relative (to the hyperplane) neighbourhood where the derivative of f is not defined. We are concerned with finding a *local* infimum of the above optimization problem, where the ridges of the piecewise linear functions involved are assumed to be known. The theoretical results we are presenting deal with (possibly) discontinuous piecewise linear functions which are *decomposable*. For example, in the continuous case, our results apply under the hypotheses of *non-degeneracy* (i.e. the situation where the gradients of the active ridges are linearly independent), and also for some particular instances of degeneracy discussed in Section 6 for which a *decomposition* exists.

A simple example of such a piecewise linear function is given by the function $f : I\!R^2 \rightarrow I\!R$ defined as follows:

$$f(x_1, x_2) = \begin{cases} -x_1 - x_2 & \text{if } x_1 \geq 0 \text{ and } x_2 \geq 0, \\ x_1 - x_2 & \text{if } x_1 < 0 \text{ and } x_2 \geq 0, \\ -x_1 + x_2 & \text{if } x_1 \geq 0 \text{ and } x_2 < 0, \\ x_1 + x_2 & \text{otherwise.} \end{cases} \tag{2}$$

The graph of f is a square-base pyramid as shown on Figure 1 (page 9), while Figure 2 shows level curves of f. We can consider the two lines $x_1 = 0$ and $x_2 = 0$ as being the set of ridges of f.

In [68], Zowe notes that, since they are not descent methods, subgradient methods (developed before the mid 1970's) yield very poor convergence rates. He then motivates the *bundle* concept (see [46]) as an alternative direct approach to nondifferentiable optimization, overcoming some of the drawbacks of the subgradient methods. Bundle methods however involve a more complicated structure which demands a more sophisticated implementation than subgradient methods. Other methods, such as [10, 23, 40, 41, 50, 53], deal with specially structured nonsmooth functions, for example the minimax problem, the l_1 problem and elementary compositions of such problems. They are able to exploit the structure explicitly and consequently these methods are significantly more efficient than bundle methods (having 2-step superlinear rates of convergence [13, 23], for instance). Nevertheless, even for these problems, if they are extremely large, bundle methods, which can handle very general problems, may be more appropriate. For recent approaches to large-scale minimax problems, see also [7] and references therein.

The idea of decomposition introduced in this paper will permit a broader generalization of Conn's approach.

Fourer [33, 34, 35] derived an extension of the simplex method for linear programming to (continuous) *monotropic* (i.e. convex, separable and linearly constrained) piecewise linear programming. His computationally practical algorithm is more efficient than indirect approaches that rely on transformation of piecewise linear programs to equivalent linear programs. See also [26, 27] which describe an extension to permit convex separable terms in the constraints. Other recent work on (continuous) piecewise linear optimization problems include Coleman & Li's globally and quadratically convergent methods for linear l_1 problems [18] and for linear l_∞ problems [17] (see also [15]).

The problem of minimizing a polyhedral convex function considered in [51] facilitates the use of inherent structure in some practical convex piecewise linear problems. An explicit representation of the subdifferential (of the generalized gradients, in the case of [52]) of the objective function is obtained yielding results easily specialized for particular problems such as l_1 and l_∞ approximation (see [51] for more examples). In [52], Osborne & al. consider the extension to a non-convex problem. However, this approach is restricted to functions which are *regular* (i.e. the usual one-sided directional derivative is equal to the generalized directional derivative in the sense of Clarke [12]). This excludes simple non-convex piecewise linear functions, such as $-|x|$ (concave functions are not regular at points of non-differentiability). Osborne & al., and also Womersley & Fletcher in [65], therefore concentrate on *locally max* functions (functions which can be represented in a neighbourhood of a point as the maximum over a finite number of smooth functions). Womersley [64] considers the censored l_1 problem which deals with continuous piecewise linear functions having the form

$$\sum_i |y_i - \max(z_i, x^T a_i)|. \tag{3}$$

Both of the methods presented in [64] and in [51] develop an extension of the projected gradient algorithm, as in the work presented in the current paper. Womersley's way (3) of expressing a piecewise linear function allows only for some restricted form of non-convexity (e.g. $-|x|$ cannot be expressed this way, as we already mentioned, in contrast to the concept of decomposition we are to introduce). In the above references, the concept corresponding to our definition of ridge in order to describe local nonsmooth behaviour is the *structure functional*. Our way of expressing a piecewise linear function (via a decomposition), enables us, for a given piecewise linear function for which the ridges are known, to say when there *exists* a decomposition. More importantly, when the function pertains to this subclass of piecewise linear functions, which we shall call *decomposable*, our approach has the advantage of providing a way to *construct* this decomposition. In addition, we allow for certain forms of discontinuous piecewise linear functions.

Considering the optimization of functions which are nonsmooth and even discontinuous is motivated by applications in VLSI and floor-planning problems [11, 60], plant layout [47, 49], batch production [43], switching regression [59], discharge allocation for hydro-electric generating stations [44], fixed-charge problems [29, 42, 57], for example.

Leaving aside the heuristic methods (which many people facing practical discontinuous optimization problems rely upon), previous work on discontinuous optimization includes smoothing algorithms. The smoothing algorithms express discontinuities by means of a step function, and then they approximate the step function by a function which is not only continuous but moreover smooth so that the resulting problem can be solved by a gradient technique. Both Imo & Leech [43] and Zang [66] developed methods in which the objective function is replaced only in the neighbourhood of the discontinuities. Two drawbacks of these methods are the potential numerical instability when we want this neighbourhood to be small, and the cost of evaluating the smoothed functions. In many instances the discontinuities of the first derivative are exactly the regions of interest and smoothing has the effect of making such regions less discernible.

One of the most important applications of nonsmooth optimization is in nonlinear programming. It arises when solving a smooth constrained optimization problem via minimizing an exact penalty function. We present an algorithm for discontinuous optimization based on ideas used in algorithms solving this very particular instance of (continuous) nonsmooth optimization problems. More specifically, the theory developed in this paper is inspired by Conn's approach [19] to this problem. The discontinuous piecewise linear algorithm to be introduced here is thus a generalization of his algorithm.

In this paper we develop a theory to address the discontinuous piecewise linear optimization problem (1). We start from the idea used by Conn [19] that was applied to the nonsmooth problem resulting from the reduction of a constrained optimization problem to an unconstrained one via an l_1 exact penalty function. We shall also use this same penalty approach in order to solve problem (1), that is we consider the unconstrained function

$$f_\gamma(x) \equiv \gamma \tilde{f}(x) + \sum_{i \in E} |f_i(x)| - \sum_{i \in I} \min[0, f_i(x)] \tag{4}$$

for a succession of decreasing positive values of the *penalty parameter* γ. The function f_γ is clearly a piecewise linear function, whenever the functions $\tilde{f}$ and f_i's are themselves piecewise linear. This penalty approach allows us to concentrate on the *unconstrained* nonsmooth optimization problem: $\inf_x f(x)$, where f is a (possibly discontinuous) piecewise linear function.

The next section sets the terminology required and presents the concepts of *activities* and *restricted gradient*, which are fundamental to the theory to be presented. We introduce in Section 3 the definition of *decomposition* of a continuous piecewise linear function into a smooth function and a sum of single-ridged functions. We then present the theorem stating that such a decomposition can always be found at a non-degenerate point, a proposition giving an explicit way to construct this decomposition, the optimality conditions and the algorithm. The following section extends the theory to the *discontinuous* piecewise linear situation by generalizing the definition of decomposition, then the decomposition theorem and hence the optimality conditions and the algorithm. We present in Section 5, the adaptation, with a penalty approach, of the algorithm in the case where constraints are present. In Section 6, we discuss how the algorithm is modified when it encounters a degenerate point at which the objective function is decomposable. Section 7 explains how the results can be straightforwardly extended to yield a first-order algorithm for the nonlinear case. Preliminary numerical results obtained by applying our algorithm to discontinuous optimization problems from models in industrial engineering are presented in Section 8. We present some conclusions in Section 9.

2 Terminology

We define the function *sign* as follows:

$$\mathrm{sign}(x) \equiv \begin{cases} 1 & \text{if } x > 0, \\ -1 & \text{if } x < 0, \\ 0 & \text{otherwise.} \end{cases}$$

If V is a subspace in $\mathbb{R}^n$, then

$$V^\perp \equiv \{x \in \mathbb{R}^n : v^T x = 0, \text{ for all } v \in V\},$$

is termed the *orthogonal complement* of V. The *range space* of an $m \times n$ matrix M is the set of vectors that can be written as a linear combination of the columns of M. The null space of M, denoted by $\mathcal{N}(M)$, is defined by

$$\mathcal{N}(M) \equiv \{x \in \mathbb{R}^n : Mx = \vec{0}\}$$

(it is the orthogonal complement of the range space of M^T).

Let f map $\mathbb{R}^n$ to $\mathbb{R}$. The usual (one-sided) *directional derivative* or the *first-order change* of f at $x \in \mathbb{R}^n$ in the direction $v \in \mathbb{R}^n$ is

$$f'(x; v) \equiv \lim_{t \to 0^+} \frac{f(x + tv) - f(x)}{t},$$

when this limit exists.

An $n \times n$ matrix P is called an *orthogonal projector* if P is *symmetric* (i.e. $P^T = P$) and *idempotent* ($P \cdot P = P$). A consequence of P being an orthogonal projector onto a subspace $V \subseteq \mathbb{R}^n$ is that P projects a vector $v \in V$ onto itself ($v \in V$ implies $Pv = v$) and P projects a vector of the orthogonal complement of V onto $\vec{0}$ ($v \in V^\perp$ implies $Pv = \vec{0}$). If A is an $n \times m$ matrix with linearly independent columns, one can show that the orthogonal projector onto $\mathcal{N}(A^T)$ is given by

$$P \equiv I - A(A^T A)^{-1} A^T,$$

where I is the $n \times n$ identity matrix.

If $\{x \in \mathbb{R}^n : a^T x - b = 0\}$, $a \in \mathbb{R}^n$, $b \in \mathbb{R}$ is a ridge of f, then, by abuse of language, we shall call $a^T x - b$ a ridge of f. We identify two ridges $a_1^T x - b_1$ and $a_2^T x - b_2$ if $\{x \in \mathbb{R}^n : a_1^T x - b_1 = 0\} = \{x \in \mathbb{R}^n : a_2^T x - b_2 = 0\}$. Let $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ be the ridges of f, where $\mathcal{R}$ is a finite index set. We say that a ridge $a^T x - b$ is *active* at $\hat{x}$ if $a^T \hat{x} - b = 0$. Let $\mathcal{A}(\hat{x}) \subseteq \mathcal{R}$ be the (finite) index set of the ridges that are active at the current point $\hat{x}$, and $A(\hat{x})$ be the matrix having as columns the gradients of the ridges which are active at $\hat{x}$. Hence, $\{a_i\}_{i \in \mathcal{A}(\hat{x})}$ denotes the column vectors of the *matrix of activities* $A(\hat{x})$. We shall use A and $\mathcal{A}$ rather than $A(\hat{x})$ and $\mathcal{A}(\hat{x})$, when it is clear from the context which is the current point. Also, when there is no confusion possible, we shall often talk about activity k or ridge k, meaning the ridge indexed by $k \in \mathcal{R}$.

A direction $d \in \mathcal{N}(A^T(\hat{x}))$ is said to *preserve* each activity $i \in \mathcal{A}(\hat{x})$, since for each $i \in \mathcal{A}(\hat{x})$ we have, $a_i^T(\hat{x} + \alpha d) - b_i = a_i^T \hat{x} - b_i = 0$. If $\mathcal{A}(\hat{x}) \neq \emptyset$, then $\nabla f(\hat{x})$ is not necessarily defined. This is because we cannot talk about the gradient of the function at $\hat{x}$ since there is no vector $g \in I\!R^n$ such that $g^T d$ is the first-order change of f in direction d, for any $d \in I\!R^n$. Thus, we cannot use, as in the smooth situation, the negative gradient direction as a descent direction. We shall first consider directions $d \in \mathcal{N}(A^T)$ such that d is a descent direction. We term any $n \times 1$ vector $g_{\hat{x}}$ such that

$$f'(\hat{x}; d) = g_{\hat{x}}^T d, \;\; \text{for all } d \in \mathcal{N}(A^T)$$

a *restricted gradient* of f at $\hat{x}$, because it is the gradient of the restriction of f to the space $\mathcal{N}(A^T(\hat{x}))$.

To illustrate these concepts, consider the following simple example:

$$\min_{x \in I\!R^2} \; -f(x),$$

$$\text{subject to } \; x_1 + x_2 \geq 0,$$

where $f(x)$ is defined by (2). An l_1 exact penalty function approach yields the unconstrained optimization problem: $\min_{x \in I\!R^2} f_\gamma(x)$, where

$$f_\gamma(x) = -\gamma f(x) - \min(0, x_1 + x_2) \tag{5}$$

and $\gamma > 0$. The derivative of the penalty function f_γ is not defined only over the three ridges

$$a_i^T x - b_i, \;\; i \in \mathcal{R} \equiv \{1, 2, 3\},$$

where

$$a_1 = (1, 0)^T, \;\; a_2 = (0, 1)^T, \;\; a_3 = (1, 1)^T, \tag{6}$$

and $b_i = 0, \;\; i \in \mathcal{R}$. At the point $\tilde{x}^T = (0, 0)$, the three ridges are active, i.e. $\mathcal{A}(\tilde{x}) = \{1, 2, 3\}$ and hence the matrix of activities is

$$A(\tilde{x}) = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

At $\hat{x}^T = (0, 1)$, we have $\mathcal{A}(\hat{x}) = \{1\}$ and

$$A(\hat{x}) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

It will be useful for what follows to introduce the functions $\{\sigma_i\}_{i \in \mathcal{R}}$ defined by:

$$\sigma_i(x) \equiv \text{sign}(a_i^T x - b_i), \;\; i \in \mathcal{R}.$$

We denote by $\sigma(x)$ the $|\mathcal{R}| \times 1$ vector whose ith component is $\sigma_i(x)$, and $3^{|\mathcal{R}|}$ is the set of all possible such vectors. Given a (possibly discontinuous) piecewise linear function f

defined over $\mathbb{R}^n$ and the set $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ of its ridges, a *cell* of f is a non-empty set $C \subseteq \mathbb{R}^n$ such that for all $x, y \in C$ we have $\sigma_i(x) = \sigma_i(y) \neq 0$, for all $i \in \mathcal{R}$. Thus, f is differentiable over a cell.

For example, the interior of the four orthants constitute a set of cells of the function f given by (2).

Let $\sigma \in 3^{|\mathcal{R}|}$, $i \in \mathcal{R}$ and σ_i be the ith component of the vector σ. A set

$$\{x : \sigma_i(x) = 0 \text{ and } \sigma_j(x) = \sigma_j,\ j \in \mathcal{R} \setminus \{i\}\}$$

is called a *segment* of ridge i.

The half-line $\{x \in \mathbb{R}^2 : x_2 = 0 \text{ and } x_1 > 0\}$ is a segment of the ridge $x_2 = 0$ (for the function f given by (2)).

Figure 3 shows the values of the gradient of the penalty function, f_γ, given by (5) when restricted to each cell. One can easily verify that a restricted gradient of f_γ at $\hat{x}$ is $g_{\hat{x}} = (\gamma, \gamma)$. Indeed, along any direction $d \in \mathbb{R}^2$ preserving activity 1 (i.e. such that $a_1^T d = 0$) we have $f'(\hat{x}; d) = g_{\hat{x}}^T d$. Note that $g_{\hat{x}} = (\lambda, \gamma)$ is also a restricted gradient of f_γ at $\hat{x}$ for any $\lambda \in \mathbb{R}$.

Given a point $\hat{x} \in \mathbb{R}^n$ and a line $L = \{x \in \mathbb{R}^n : x = \hat{x} + \alpha d, \alpha \in \mathbb{R}\}$, we say that $x_0 \in L$ is a *breakpoint* of f along d when at least one ridge i of f is active at x_0 with $a_i^T d \neq 0$ (i.e. i is not active at $\hat{x}$). In our example, starting from $\hat{x}^T = (0, 1)$, $x_0^T = (-1, 1)$ is a breakpoint of f_γ along direction $d = (-1, 0)^T$, as we "hit" ridge 3 at x_0 (see Figure 3).

3 Continuous Piecewise Linear Optimization

We will assume that a continuous piecewise linear function, f, is given in the following form: The ridges, $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set, of f are given, i.e. we are given: $a_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$, for each $i \in \mathcal{R}$. At any point $\hat{x} \in \mathbb{R}^n$, the set of ridges which are active at $\hat{x}$, $\mathcal{A}(\hat{x})$, is given. Finally, consider the set

$$C_{\hat{x}}^P \equiv \{x \in B(\hat{x}) : a_i^T x > b_i, i \in P \text{ and } a_i^T x < b_i, i \in \mathcal{A}(\hat{x}) \setminus P\},$$

where $B(\hat{x})$ is a neighbourhood of $\hat{x}$ and $P \subseteq \mathcal{A}(\hat{x})$. If $B(\hat{x})$ is small enough, f is linear over $C_{\hat{x}}^P$, i.e. $f(x) = f(\hat{x}) + x^T c_{\hat{x}}^P$, $x \in C_{\hat{x}}^P$, for some vector $c_{\hat{x}}^P \in \mathbb{R}^n$. We shall consider that at any point $\hat{x} \in \mathbb{R}^n$ and for any $P \subseteq \mathcal{A}(\hat{x})$ such that $C_{\hat{x}}^P \neq \emptyset$, we can obtain the vector $c_{\hat{x}}^P$. (Hence, we are assuming that more information on the structure of the objective function is available than, for example, in a bundle method [46], which assumes that only one element of the subdifferential is known at any point.)

3.1 The Decomposition

Definition 1 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set. Let $\hat{x} \in \mathbb{R}^n$, $g_{\hat{x}} \in \mathbb{R}^n$ and $\psi_{\hat{x}}$ be a function defined on $\mathbb{R}^n$ such that we have:*

$$f(x) = f(\hat{x}) + g_{\hat{x}}^T (x - \hat{x}) + \psi_{\hat{x}}(x),$$

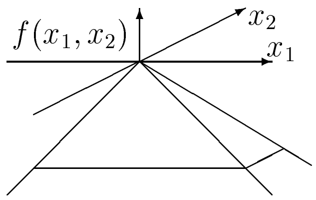

Figure 1: Example of a piecewise linear function f

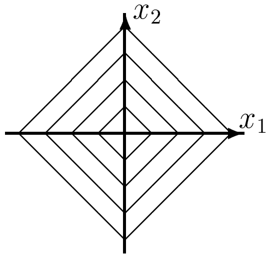

Figure 2: Level curves of f

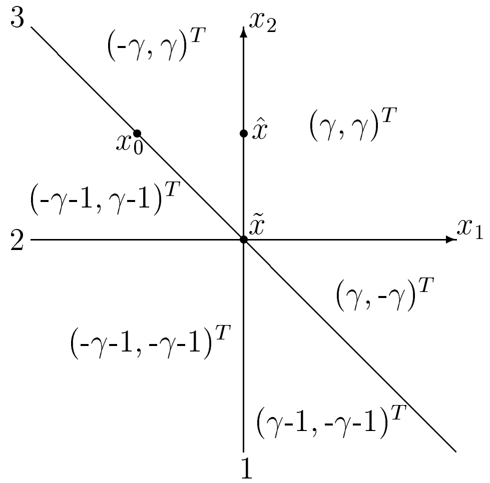

Figure 3: Values of the gradient of f_γ over the six cells

for all x in some neighbourhood, $B(\hat{x})$, of $\hat{x}$, where

$$\psi_{\hat{x}}(x) = \sum_{i \in \mathcal{A}(\hat{x})} \nu_{\hat{x}}^i \min(0, a_i^T(x - \hat{x}))$$

for some scalars $\{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x})}$.

We say that $g_{\hat{x}}, \{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x})}$ is a decomposition of f (into a smooth function and a sum of continuous functions having a single ridge) at $\hat{x}$.

To illustrate, one way of decomposing function f_γ given by (5) (see Figure 3) at $\tilde{x} = (0,0)^T$ could be the following:

$$f_\gamma(x) = g_{\tilde{x}}^T x + \sum_{i \in \mathcal{A}(\tilde{x})} \nu_{\tilde{x}}^i \min(0, a_i^T x), \tag{7}$$

for all $x \in I\!R^2$, where $\mathcal{A}(\tilde{x}) = \{1,2,3\}$; a_i, $i = 1,2,3$ are defined by (6); and

$$\begin{aligned} g_{\tilde{x}}^T &= (\gamma, \gamma); \\ \nu_{\tilde{x}}^1 &= -2\gamma; \\ \nu_{\tilde{x}}^2 &= -2\gamma; \\ \nu_{\tilde{x}}^3 &= -1. \end{aligned}$$

We omit the subscript 'γ' in $g_{\tilde{x}}$ and $\nu_{\tilde{x}}^i$ for notational simplicity.

We shall shortly give a practical way to find a decomposition at a given point $\hat{x}$. In the above example, we can easily provide a *global* decomposition for each $\hat{x}$ in the domain of f_γ, that is, a decomposition such that $\nu_{\hat{x}}^i = \nu_{\tilde{x}}^i$ at any points $\hat{x}, \tilde{x}$, $i \in \mathcal{R}$. We shall later deal with instances, in degenerate situations, where the scalars $\{\nu_{\hat{x}}^i\}_{i \in \mathcal{R}}$ do depend upon $\hat{x}$. In fact, even in these instances, they depend only on $\sigma(\hat{x})$. Indeed, we shall see that we can define a decomposition with common $\{\nu_{\hat{x}}^i\}_{i \in \mathcal{R}}$ for all $\hat{x}$ which are on the same segment of a ridge.

When there exists a (not necessary global) decomposition of a function f at a point $\hat{x} \in I\!R^n$, we say that f is *decomposable* at $\hat{x}$. Suppose that f is decomposable at a point $\hat{x}$, and let $g_{\hat{x}}, \{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x})}$ be the decomposition. Choosing a direction that would not change the value of *the nonsmooth part* of f, $\psi_{\hat{x}}$, (for example, a direction preserving the current activities) and reduce that of *the smooth part*, $f(\hat{x}) + g_{\hat{x}}^T(x - \hat{x})$, (whenever possible) would be enough to obtain descent in f. We have, for $\alpha > 0$ small enough:

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \alpha g_{\hat{x}}^T d + \alpha \sum_{i \in \mathcal{A}(\hat{x}) : a_i^T d < 0} \nu_{\hat{x}}^i a_i^T d.$$

Thus, restricting d to be in $\mathcal{N}(A^T(\hat{x}))$ we obtain

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \alpha g_{\hat{x}}^T d.$$

Clearly, $g_{\hat{x}}$ is a restricted gradient of f at $\hat{x}$. Given a decomposition, $g_{\hat{x}}, \{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x})}$, of f at $\hat{x}$, we shall call $g_{\hat{x}}$, by convention, *the* restricted gradient of f at $\hat{x}$. The next theorem will state that, given the set of ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ of f at $\hat{x}$, the decomposition of f at

$\hat{x}$ and, hence, the restricted gradient of f at $\hat{x}$, are unique provided that the gradients of the ridges which are active at $\hat{x}$ are linearly independent.

Unless it is essential to specify the subscript '$\hat{x}$' in $g_{\hat{x}}, \{\nu^i_{\hat{x}}\}_{i\in\mathcal{A}(\hat{x})}$, we shall sometimes omit it and talk only about a decomposition $g, \{\nu^i\}_{i\in\mathcal{A}}$ of f at $\hat{x}$.

Let A_{-k} be the matrix having as columns the gradients of the ridges which are active except for activity $k \in \mathcal{A}$. We call a direction $d \in \mathcal{N}(A^T_{-k})$, $k \in \mathcal{A}$, such that $a_k^T d \neq 0$, a *single-dropping* direction or a direction *dropping* only activity k, since it preserves all activities but activity k (if $\mathcal{A} = \{k\}$, then $\mathcal{N}(A_{-k})$ is simply $I\!R^n$). We say that activity k is *dropped positively* if moreover $a_k^T d > 0$, and that activity k is *dropped negatively* when d is such that $a_k^T d < 0$. With a decomposition $g, \{\nu^i\}_{i\in\mathcal{A}}$ of f at $\hat{x}$, we know that there is a vector $v \in I\!R^n$ such that if d is a single-dropping direction, then the change in the nonsmooth part is $v^T d$, where *v is a multiple of the gradient of the ridge being dropped.* This observation will permit the derivation of optimality conditions similar to Lagrange multiplier rules; a multiplier rule will be associated with each activity.

The minimization of some special cases of continuous piecewise linear functions have been studied before, for example, in l_1 data fitting or when using the l_1 exact penalty function to solve a linear programming problem [4, 19]. In the latter problem, we want to minimize a particular piecewise linear penalty function f_γ. The function f_γ is as given in (4), where $\tilde{f}$ and f_i, $i \in E \cup I$, are linear. Thus, the penalty function to minimize has the form

$$f_\gamma(x) = \gamma c^T x + \sum_{i\in E} |a_i^T x - b_i| - \sum_{i\in I} \min(0, a_i^T x - b_i), \tag{8}$$

for some $c \in I\!R^n$; $a_i \in I\!R^n$, $b_i \in I\!R$, $i \in E \cup I$. In order to solve this problem, Conn defined the restricted gradient of f_γ at x to be

$$g(x) = \gamma c + \sum_{i\in E} a_i \mathrm{sign}(a_i^T x - b_i) - \sum_{i\in I} a_i s(a_i^T x - b_i),$$

where the function $s(x)$ is defined to take the value 1 if x is negative and 0 otherwise. This way, he could express f_γ, in a neighbourhood of a point $\hat{x}$, as the sum of a differentiable part, $f_\gamma(\hat{x}) + g(\hat{x})^T(x - \hat{x})$, and a nondifferentiable part,

$$\sum_{i\in E\cap\mathcal{A}(\hat{x})} |a_i^T x - b_i| - \sum_{i\in I\cap\mathcal{A}(\hat{x})} \min(0, a_i^T x - b_i).$$

Clearly the change of the nondifferentiable part of f_γ in a single-dropping direction d can easily be written as the inner product of d with a multiple of the gradient of the activity to be dropped. For example, if at the current point $\hat{x}$, $a_k^T x - b_k$ is active and d drops negatively this activity, then

$$f'_\gamma(\hat{x}, d) = (g(\hat{x}) + (-1)a_k)^T d.$$

If d drops k positively, then

$$f'_\gamma(\hat{x}, d) = \begin{cases} (g(\hat{x}) + 1a_k)^T d & \text{if } k \in E, \\ (g(\hat{x}) + 0a_k)^T d & \text{if } k \in I. \end{cases}$$

Given an arbitrary continuous piecewise linear function f, two questions remain: How to find a decomposition of f into a smooth function and a sum of continuous functions having a single ridge and, above all, does such a decomposition always exist? At first sight, the answer to the latter question seems unlikely to be yes. Indeed, if m ridges of f are active at $\hat{x}$, it means that there are 2^m cells in any small neighbourhood of $\hat{x}$, assuming $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$ are linearly independent. One can then wonder how the vector $g_{\hat{x}}$ and the m scalars $\{\nu^i_{\hat{x}}\}_{i\in\mathcal{A}(\hat{x})}$, together with the gradients of the activities, $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$, completely characterize the behaviour of f over the 2^m cells in the neighbourhood of $\hat{x}$. The next theorem gives a sufficient condition for a continuous piecewise linear function to be decomposable at a given point.

Theorem 1 (Decomposition) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous piecewise linear function with ridges $\{a_i^T x - b_i\}_{i\in\mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in \mathbb{R}^n$. If $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$, the gradients of the ridges of f which are active at $\hat{x}$, are linearly independent, then f is decomposable at $\hat{x}$ and the decomposition is unique.*

To give an intuitive idea as to why the result is true, one should note that in constructing an arbitrary piecewise linear function having m linearly independent ridges active at $\hat{x}$, the requirement of the continuity of f reduces greatly the number of degrees of freedom. One cannot just assign arbitrary linear functions to each of the 2^m cells in the neighbourhood of $\hat{x}$. Consider for example the partition of $\mathbb{R}^2$ into four cells by the ridges $x_1 = 0$ and $x_2 = 0$. Defining a function over the second and fourth orthants determines completely a continuous piecewise linear function over the whole domain. In fact, assigning m linear functions to m different cells will be enough to determine the value of f on the $2^m - m$ remaining cells, if we require f to be continuous.

Proof: Let us suppose without loss of generality that $f(\vec{0}) = 0$, and $\hat{x} = \vec{0}$ (thus, $b_i = 0$ for all $i \in \mathcal{A}(\hat{x})$), and $\mathcal{A}(\hat{x}) \equiv \{1, 2, \ldots, m\}$. In a neighbourhood, $B(\hat{x})$, of $\hat{x}$, function f has the form

$$f(x) = x^T c^J$$

whenever

$$\begin{array}{rrcl} & a_i^T x & \geq & 0, \;\; i \in J \\ \text{and} & a_i^T x & < & 0, \;\; i \in \{1, 2, \ldots, m\} \setminus J, \end{array}$$

for some vectors $c^J \in \mathbb{R}^n$ given for each $J \subseteq \{1, 2, \ldots, m\}$. We can define an invertible linear transformation,

$$y = Lx,$$

such that

$$y_i \equiv a_i^T x, \;\; 1 < i \leq m$$

(and in the case where $m < n$,

$$y_i \equiv \tilde{a}_i^T x, \;\; m < i \leq n,$$

where the $\tilde{a}_i \in I\!R^n$ are such that $\{a_i,\ 1 \le i \le m\} \cup \{\tilde{a}_i,\ m < i \le n\}$ are linearly independent). Thus, we can write f, in a neighbourhood, $B(\hat{x})$, of $\hat{x}$, using the form

$$f(x) = (L^{-1}y)^T c^J$$

whenever

$$\begin{array}{rrcl} & y_i & \ge & 0,\ \ i \in J \\ \text{and} & y_i & < & 0,\ \ i \in \{1, 2, \dots, m\} \setminus J, \end{array}$$

where the $c^J \in I\!R^n$ are given for each $J \subseteq \{1, 2, \dots, m\}$.

Let

$$d^J \equiv L^{-T} c^J$$

for each $J \subseteq \{1, 2, \dots, m\}$. Then, in a neighbourhood of $\hat{x}$, $f(x) = h(Lx)$, where

$$h(y) = y^T d^J$$

whenever

$$\begin{array}{rrcl} & y_i & \ge & 0,\ \ i \in J \\ \text{and} & y_i & < & 0,\ \ i \in \{1, 2, \dots, m\} \setminus J. \end{array}$$

Since the function f is continuous, h is continuous. The continuity of h yields, for $I, J \subseteq \{1, 2, \dots, m\}$:

$$d_i^I = d_i^J \ \text{ whenever } i \in I \cap J \text{ or } i \in \{1, 2, \dots, m\} \setminus (I \cup J) \text{ or } m < i \le n. \tag{9}$$

Clearly, (9) is true if and only if for $I \subseteq \{1, 2, \dots, m\}$ and $1 \le i \le n$:

$$d_i^I = \begin{cases} d_i^{\{1,2,\dots,m\}} & \text{if } i \in I, \\ d_i^{\emptyset} & \text{otherwise.} \end{cases}$$

Hence,

$$h(y) = \sum_{i=1}^{n} d_i y_i,$$

where

$$d_i = \begin{cases} d_i^{\{1,2,\dots,m\}} & \text{if } y_i \ge 0, \\ d_i^{\emptyset} & \text{if } y_i < 0. \end{cases}$$

That is to say,

$$h(y) = y^T d^{\{1,2,\dots,m\}} + \sum_{i=1}^{m} (d_i^{\emptyset} - d_i^{\{1,2,\dots,m\}}) \min(0, y_i),$$

since for $m < i \le n$, $d_i^{\emptyset} = d_i^{\{1,2,\dots,m\}}$. Hence, in a neighbourhood of $\hat{x}$,

$$f(x) = x^T c^{\{1,2,\dots,m\}} + \sum_{i=1}^{m} \nu^i \min(0, a_i^T x), \text{where } \ \nu^i \equiv d_i^{\emptyset} - d_i^{\{1,2,\dots,m\}},$$

which is a decomposition of f at $\hat{x}$.

In order to prove the uniqueness of the decomposition, let $g, \{\nu^i\}_{i\in\mathcal{A}}$ and $g', \{\nu'^i\}_{i\in\mathcal{A}}$ be two decompositions of f at $\hat{x}$. By definition of the decomposition, we have for all x in some neighbourhood, $B(\hat{x})$, of $\hat{x}$:

$$f(x) = g^T x + \sum_{i\in\mathcal{A}} \nu^i \min(0, a_i^T x) = g'^T x + \sum_{i\in\mathcal{A}} \nu'^i \min(0, a_i^T x). \tag{10}$$

Using (10) with $x \in B(\hat{x})$ such that $a_i^T x \geq 0$ for all $i \in \mathcal{A}$ yields

$$\begin{aligned} g^T x &= g'^T x \\ \text{or} \;\; (g - g')^T x &= 0. \end{aligned} \tag{11}$$

Equation (11) is true for any $x \in B(\hat{x})$ in the set

$$S \equiv \{x \in I\!R^n : a_i^T x \geq 0, \text{ for all } i \in \mathcal{A}\}.$$

In the case where $|\mathcal{A}| = n$, we show that $g - g' = \vec{0}$ by considering each of the (linearly independent) directions

$$d^k \equiv P_{-k}(a_k), \quad k \in \mathcal{A},$$

where P_{-k} is the orthogonal projector onto $\mathcal{N}(A_{-k}(\hat{x}))$. If $|\mathcal{A}| \equiv m < n$, then consider each of the directions

$$\begin{aligned} d^k &\equiv \tilde{P}_{-k}(a_k), \quad k \in \mathcal{A} \;\text{ and} \\ d^k &\equiv \tilde{P}_{-k}(\tilde{a}_k), \quad k \in \{1, 2, \ldots, n\} \setminus \mathcal{A}, \end{aligned}$$

where $\tilde{P}_{-k}$ is the orthogonal projector onto $\mathcal{N}(\tilde{A}_{-k}(\hat{x}))$, $\tilde{A}_{-k}(\hat{x})$ has as its columns the vectors $a_1, \ldots, a_m, \tilde{a}_{m+1}, \ldots, \tilde{a}_n$ and the $\tilde{a}_k$'s extend the a_k's to form a basis of the entire space.

We show that $\nu^i = \nu'^i$ for any $i \in \mathcal{A}(\hat{x})$ by using (10) with $x = -P_{-i}(a_i)$. $\square$

The next proposition gives an explicit way to construct the decomposition of a continuous piecewise linear function at a non-degenerate point.

Proposition 1 *Let $f : I\!R^n \to I\!R$ be a continuous piecewise linear function with ridges $\{a_i^T x - b_i\}_{i\in\mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in I\!R^n$. Assume that $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$, the gradients of the ridges of f which are active at $\hat{x}$, are linearly independent and that, without loss of generality, f has the following form in a neighbourhood of $\hat{x}$:*

$$f(x) = f(\hat{x}) + (x - \hat{x})^T c^J \tag{12}$$

whenever

$$\begin{aligned} a_i^T(x - \hat{x}) &\geq 0, \quad i \in J \\ \text{and} \quad a_i^T(x - \hat{x}) &< 0, \quad i \in \mathcal{A}(\hat{x}) \setminus J, \end{aligned}$$

for some $c^J \in I\!R^n$ given for each $J \subseteq \mathcal{A}(\hat{x})$.

For every $i \in \mathcal{A}(\hat{x})$, let J_i be any subset of $\mathcal{A}(\hat{x})$ such that $i \in J_i$, and let the scalars $\{\lambda^i\}_{i\in\mathcal{A}(\hat{x})}$ be defined by

$$\lambda^i a_i = c^{J_i - i} - c^{J_i}. \tag{13}$$

Then, $c^{\mathcal{A}(\hat{x})}, \{\lambda^i\}_{i\in\mathcal{A}(\hat{x})}$ is the decomposition of f at $\hat{x}$.

Proof: Let $\{J_i\}_{i\in\mathcal{A}(\hat{x})}$ be such that $i \in J_i \subseteq \mathcal{A}(\hat{x})$, for each $i \in \mathcal{A}(\hat{x})$. Note first that since f is decomposable at $\hat{x}$, then there exists some decomposition $g, \{\nu^i\}_{i\in\mathcal{A}(\hat{x})}$ of f at $\hat{x}$, and hence, for any $J \subseteq \mathcal{A}(\hat{x})$ we have, by (12):

$$c^J = g + \sum_{k\in\mathcal{A}(\hat{x})\setminus J} \nu^k a_k.$$

Thus, for any $J \subseteq \mathcal{A}(\hat{x})$ such that $i \in J$ we have

$$\begin{aligned} c^{J-i} - c^J &= g + \sum_{k\in\mathcal{A}(\hat{x})\setminus(J-i)} \nu^k a_k - (g + \sum_{k\in\mathcal{A}(\hat{x})\setminus J} \nu^k a_k) \\ &= \nu^i a_i. \end{aligned} \tag{14}$$

This is true in particular for $J = J_i$; that is to say

$$c^{J_i-i} - c^{J_i} = \nu^i a_i,$$

which implies, by (13), that $\lambda^i a_i = \nu^i a_i$, and hence, by (14),

$$c^{J-i} - c^J = \lambda^i a_i, \tag{15}$$

for any $J \subseteq \mathcal{A}(\hat{x})$ such that $i \in J$.

Now let $\tilde{x}$ be an arbitrary point of a small neighbourhood of $\hat{x}$. In order to show that $c^{\mathcal{A}(\hat{x})}, \{\lambda^i\}_{i\in\mathcal{A}(\hat{x})}$ is a decomposition of f at $\hat{x}$, we would like to show that

$$f(\tilde{x}) = f(\hat{x}) + (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i\in\mathcal{A}(\hat{x})} \lambda^i \min(0, a_i^T(\tilde{x} - \hat{x})).$$

Let

$$K \equiv \{i \in \mathcal{A}(\hat{x}) : a_i^T(\tilde{x} - \hat{x}) \geq 0\}$$

and

$$K^c \equiv \{i \in \mathcal{A}(\hat{x}) : a_i^T(\tilde{x} - \hat{x}) < 0\}.$$

Suppose without loss of generality that

$$K^c \equiv \{1, 2, \ldots, l\},$$

and let

$$K_i \equiv K \cup \{1, 2, \ldots, i\},\ i \in K^c, \quad \text{and} \quad K_0 \equiv K.$$

We have

$$\begin{aligned} f(\hat{x}) &+ (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i\in\mathcal{A}(\hat{x})} \lambda^i \min(0, a_i^T(\tilde{x} - \hat{x})) \\ = \quad & f(\hat{x}) + (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i=1}^{l} \lambda^i a_i^T(\tilde{x} - \hat{x}) \\ = \quad & f(\hat{x}) + (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i=1}^{l} (c^{K_i-i} - c^{K_i})^T(\tilde{x} - \hat{x}), \ \text{ using (15) with } J = K_i, \\ = \quad & f(\hat{x}) + (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i=1}^{l} (c^{K_{i-1}} - c^{K_i})^T(\tilde{x} - \hat{x}) \\ = \quad & f(\hat{x}) + (\tilde{x} - \hat{x})^T c^{\mathcal{A}(\hat{x})} + (c^{K_0} - c^{K_l})^T(\tilde{x} - \hat{x}) \\ = \quad & f(\hat{x}) + (\tilde{x} - \hat{x})^T c^K, \ \text{ since } K_l \equiv \mathcal{A}(\hat{x}) \text{ and } K_0 \equiv K, \\ = \quad & f(\tilde{x}), \end{aligned}$$

by (12) and the definition of K. $\square$

Thus, intuitively, we have for a continuous piecewise linear function f that, in the neighbourhood of a point $\hat{x}$ such that $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$ are linearly independent, the difference between the gradient of f when f is restricted to a cell C_1, and the gradient of f when f is restricted to an "adjacent" cell C_2, i.e. C_1 and C_2 are "separated" only by *one* ridge, $a_i^T x - b_i$, is always a multiple of a_i, the gradient of the ridge. The factor involved is just the scalar $\nu_{\hat{x}}^i$ needed to obtain the decomposition of f. This also explains intuitively why the decomposition is unique at a point $\hat{x}$ where the active ridges, $\mathcal{R}(\hat{x})$, are linearly independent. We indeed (arbitrarily) defined *the* restricted gradient to be a particular restricted gradient, namely the gradient of f when restricted to a cell where the residuals, $\{a_i^T x - b_i\}_{i\in\mathcal{R}(\hat{x})}$, are all positive, in order to obtain a unique decomposition. Proposition 1 says that, in order to build a decomposition of a given continuous piecewise linear function f at a point at which the gradients of the activities are linearly independent, it suffices to know, for example, the $|\mathcal{A}(\hat{x})| + 1$ gradients (of restrictions of f): $c^{\mathcal{A}(\hat{x})}$ and $c^{\mathcal{A}(\hat{x})-i}$, $i \in \mathcal{A}(\hat{x})$. *We do not need to use the* $2^{|\mathcal{A}(\hat{x})|}$ *gradients* corresponding to restrictions of f to each cell.

To illustrate, here is an automatic way to obtain the decomposition of function f given by (2) at the point $\hat{x} = (0,0)$. Let $\{a_i^T x - b_i\}_{i\in\mathcal{R}}$, $\mathcal{R} = \{1,2\}$, be the ridges of f, where

$$a_1 = (1,0)^T,\ a_2 = (0,1)^T$$

and $b_i = 0$, $i = 1,2$. We have $\mathcal{A}(\hat{x}) = \{1,2\}$. Take $g_{\hat{x}}$ as the gradient of f when restricted to the cell

$$\{x \in I\!R^2 : a_i^T x - b_i > 0,\ i \in \mathcal{A}(\hat{x}),\}$$

that is to say, set $g_{\hat{x}} = (-1,-1)^T$. For each $i = 1,2$, in order to obtain the scalar $\nu_{\hat{x}}^i$, one only needs to pick up two cells "separated" only by ridge i, and to compute the difference between the gradients of f restricted to each of these cells. The result is then equal to $\nu_{\hat{x}}^i a_i$, by Proposition 1. For example, in order to determine $\nu_{\hat{x}}^1$, we can use the cells

$$\{x \in I\!R^2 : a_1^T x - b_1 < 0 \text{ and } a_2^T x - b_2 > 0\}$$

and

$$\{x \in I\!R^2 : a_1^T x - b_1 > 0 \text{ and } a_2^T x - b_2 > 0\},$$

which yields

$$\nu_{\hat{x}}^1 a_1 = (1,-1)^T - (-1,-1)^T = (2,0)^T.$$

Since $a_1 = (1,0)^T$, we obtain $\nu_{\hat{x}}^1 = 2$. We remark that using rather the cells

$$\{x \in I\!R^2 : a_1^T x - b_1 < 0 \text{ and } a_2^T x - b_2 < 0\}$$

and

$$\{x \in I\!R^2 : a_1^T x - b_1 > 0 \text{ and } a_2^T x - b_2 < 0\},$$

we would obtain the same multiple of a_1.

In a similar manner, we obtain $\nu_{\hat{x}}^2 = 2$ and hence,

$$f(x) = g_{\hat{x}}^T x + \sum_{i\in\mathcal{A}(\hat{x})} \nu_{\hat{x}}^i \min(0, a_i^T x),$$

where $g_{\hat{x}} = (-1,-1)^T$ and $\nu^1_{\hat{x}} = \nu^2_{\hat{x}} = 2$.

Note that it does not appear to be necessary to automate the construction of decompositions in applications. According to our experience, decompositions (when they exist) are easily constructed in practical problems (cf. Section 8).

The decomposition, $c^{\mathcal{A}(\hat{x})}$, $\{\lambda^i\}_{i\in\mathcal{A}(\hat{x})}$, constructed in the proof of Proposition 1, is valid for any other point $\tilde{x}$ such that $\sigma(\tilde{x}) = \sigma(\hat{x})$. Indeed, if $\tilde{x}$ is such that $\sigma(\tilde{x}) = \sigma(\hat{x})$, then any neighbourhoods of $\hat{x}$ and $\tilde{x}$, small enough, intersect exactly the same cells. For example, if the continuous piecewise linear function f is nondifferentiable at a point $\hat{x}$, and we have a decomposition g, ν^i of f at $\hat{x}$, then g, ν^i is a decomposition of f at any other point on the same segment.

In addition, when *every* point, x, of the domain of f is such that $\{a_i\}_{i\in\mathcal{A}(x)}$ are linearly independent, using an inductive argument, we can show that the $\nu^i_{\hat{x}}$'s involved in the decomposition of f at a point $\hat{x}$ are independent of $\hat{x}$. Note that from this we can prove that, assuming that the activities are linearly independent at every point $\hat{x} \in I\!R^n$, a continuous piecewise linear function f with ridges $\{a_i^T x - b_i\}_{i\in\mathcal{R}}$ has the form:

$$f(x) = b + c^T x + \sum_{i\in\mathcal{R}} \nu^i \min(0, a_i^T x - b_i), \quad \textit{for all } x \in I\!R^n,$$

for some vector $c \in I\!R^n$ and scalars b and $\{\nu^i\}_{i\in\mathcal{R}}$.

3.2 Optimality Conditions

Let $\hat{x} \in I\!R^n$, $g_{\hat{x}}$ be any restricted gradient of f at $\hat{x}$, and Z be an $n \times (n - |\mathcal{A}(\hat{x})|)$ matrix whose column vectors form a basis of $\mathcal{N}(A^T(\hat{x}))$. Analogous to active-set methods, where the dimension of a subproblem is reduced by considering only the space where the current activities are preserved, we call $Z^T g_{\hat{x}}$ a *reduced restricted gradient* of f at $\hat{x}$. When f is decomposable at $\hat{x}$, it is a reduced gradient of the smooth part of f. Note that Z is not unique and the reduced restricted gradient is not uniquely defined.

Let P be the orthogonal projector onto the space $\mathcal{N}(A^T(\hat{x}))$. Notice that

$$f'(\hat{x}, d) = 0 \;\; \forall\, d \in \mathcal{N}(A^T) \;\Leftrightarrow\; g_{\hat{x}}^T d = 0 \;\; \forall\, d \in \mathcal{N}(A^T) \;\Leftrightarrow\; Z^T g_{\hat{x}} = \vec{0}$$

(thus the fact that $Z^T g_{\hat{x}} = \vec{0}$ is independent of which particular restricted gradient, $g_{\hat{x}}$, of f we consider) and

$$Z^T g_{\hat{x}} = \vec{0} \;\Leftrightarrow\; g_{\hat{x}} \in \mathcal{R}(A(\hat{x})) \;\Leftrightarrow\; P(g_{\hat{x}}) = \vec{0}.$$

We consider two cases:

Case 1: The reduced restricted gradient is non-zero.

Consider the direction $p \equiv -P(g_{\hat{x}})$. We can easily show that along p, f decreases. Indeed, for $\alpha > 0$ small enough we have

$$\begin{aligned} f(\hat{x} + \alpha p) &= f(\hat{x}) + \alpha p^T g_{\hat{x}}, \quad \text{since } p \in \mathcal{N}(A^T), \\ &= f(\hat{x}) - \alpha \|P(g_{\hat{x}})\|^2 \\ &< f(\hat{x}). \end{aligned} \tag{16}$$

We can show that p is moreover the *steepest* descent direction among all directions in $\mathcal{N}(A^T)$.

Case 2: The reduced restricted gradient is zero.

In this case, we say that $\hat{x}$ is a *dead point*. It is not possible to obtain any non-zero change in the objective function f while keeping the same set of activities, as follows from equation (16), which is valid for any direction $p \in \mathcal{N}(A^T)$. Hence, we have to drop some activities if we want to obtain a descent direction. We henceforward assume that the columns of $A(\hat{x})$ are linearly independent. We return to the case of *degeneracy* (i.e. the situation where the gradients of the activities are linearly dependent) in Section 6. Thus, from the decomposition theorem (Theorem 1), f is decomposable at $\hat{x}$. Let $g_{\hat{x}}, \{\nu^i_{\hat{x}}\}_{i\in\mathcal{A}(\hat{x})}$ be a decomposition of f at $\hat{x}$. The point $\hat{x}$ is a stationary point of the smooth part of f, and we shall either establish the optimality of f or derive a descent direction.

Consider moving from the current dead point $\hat{x}$ to a displaced point $\hat{x} + \alpha d$. We have, for $\alpha > 0$ small enough:

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \alpha d^T [g_{\hat{x}} + \sum_{i\in\mathcal{A}(\hat{x}): a_i^T d < 0} \nu^i_{\hat{x}} a_i].$$

But the fact that $\hat{x}$ is a dead point implies that there exist scalars $\{u_i\}_{i\in\mathcal{A}(\hat{x})}$ such that

$$g_{\hat{x}} = \sum_{i\in\mathcal{A}(\hat{x})} u_i a_i$$

and hence,

$$f(\hat{x} + \alpha d) = f(\hat{x}) + \alpha[\sum_{i\in\mathcal{A}(\hat{x})} u_i a_i^T d + \sum_{i\in\mathcal{A}(\hat{x})} \lambda^{i,d}_{\hat{x}} a_i^T d], \tag{17}$$

where

$$\lambda^{i,d}_{\hat{x}} = \begin{cases} \nu^i_{\hat{x}} & \text{if } a_i^T d < 0, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

We can show that a direction d^{k^+} $(d^{k^-}) \in \mathcal{N}(A^T_{-k})$ dropping positively (defined in Section 3.1) (negatively) activity $k \in \mathcal{A}(\hat{x})$ exists.

Hence, using the fact that $a_i^T d^{k^\pm} = 0$ for all $i \in \mathcal{A}(\hat{x}) \setminus \{k\}$, equation (17) becomes (with $d = d^{k^\pm}$):

$$f(\hat{x} + \alpha d^{k^\pm}) = f(\hat{x}) \pm \alpha(u_k + \lambda^{k,d^{k^\pm}}_{\hat{x}})|a_k^T d^{k^\pm}|, \tag{19}$$

where '$\pm$' is a '$+$' when k is dropped positively and a '$-$' if it is dropped negatively. Thus, at a dead point, a direction dropping only activity $k \in \mathcal{A}(\hat{x})$ does not yield descent if and only if

$$0 \le u_k \le -\nu^k_{\hat{x}}.$$

The next theorem will state that in fact, it is sufficient to verify whether single-dropping directions yield descent, in order to determine if the current point is optimal.

In the same way, the optimality conditions for many nondifferentiable problems can be determined. When minimizing f_γ given by (8), one can show using (39) from Section 6 (and by proving the fact that one needs to consider only single-dropping directions as possible descent directions, as we shall do in the proof of Theorem 2) that a feasible dead point, x^*, is optimal for the original problem if and only if $0 \leq u_i^* \leq 2$ for all $i \in \mathcal{A}(x^*) \cap E$ and $0 \leq u_i^* \leq 1$ for all $i \in \mathcal{A}(x^*) \cap I$, where the scalars $\{u_i^*\}_{i\in\mathcal{A}(x^*)}$ are the coefficients in the linear combination of the restricted gradient (given by (38) from Section 6) of f at x^* in terms of the columns of $A(x^*)$ (note that the multiplier rule corresponding to an active equality constraint has the form $0 \leq u_i^* \leq 2$ rather than the more standard $-1 \leq u_i^* \leq 1$, due to the canonical form, $\sum_{i\in\mathcal{A}(\hat{x})} \nu_{\hat{x}}^i \min(0, a_i^T(x-\hat{x}))$, with which we fixed the definition of the nonsmooth part of f). The projection method for the uncapacitated facility location problem of Conn and Cornuéjols [22] determines optimality conditions and descent directions in a similar fashion, that is to say, from the observation of the behaviour of the function in each single-dropping direction.

Theorem 2 (Optimality Conditions) *Let $x^* \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ be a continuous piecewise linear function with ridges $\{a_i^T x - b_i\}_{i\in\mathcal{R}}$, where $\mathcal{R}$ is a finite index set. Assume linear independence of $\{a_i\}_{i\in\mathcal{A}(x^*)}$, and let $g_{x^*}, \{\nu_{x^*}^i\}_{i\in\mathcal{A}(x^*)}$ be the decomposition of f at x^*. The point x^* is a local minimum of f if and only if there exist scalars u_i^*, $i \in \mathcal{A}(x^*)$ such that*

1. *$g_{x^*} = \sum_{i\in\mathcal{A}(x^*)} u_i^* a_i$ (or, equivalently, the reduced restricted gradient of f at x^* is zero, i.e. x^* is a dead point) and*
2. *$0 \leq u_i^* \leq -\nu_{x^*}^i$, for all $i \in \mathcal{A}(x^*)$.*

A simple interpretation of the condition $0 \leq u_i^* \leq -\nu_{x^*}^i$ when x^* is a dead point, is that the directional derivative of f in the single-dropping directions $d^{i^\pm}$ (dropping $i \in \mathcal{A}(x^*)$) is non-negative.

Proof: The necessity of the optimality conditions follows from the earlier discussion. Note that if x^* is a dead point, then the coefficients $\{u_i^*\}_{i\in\mathcal{A}(x^*)}$ exist and are uniquely defined, by the assumption of the linear independence of $\{a_i\}_{i\in\mathcal{A}(x^*)}$. If we have $u_k^* < 0$ ($u_k^* > \nu_{x^*}^k$) for some $k \in \mathcal{A}(x^*)$, then a direction dropping activity k positively (negatively) is a descent direction.

In order to prove the sufficiency of the optimality conditions, suppose that the two conditions hold. Consider $d \in \mathbb{R}^n$, an arbitrary direction. Recall from equation (17), which was valid for an arbitrary d at a dead point, that

$$f(x^* + \alpha d) = f(x^*) + \alpha \sum_{i\in\mathcal{A}(x^*)} (u_i^* + \lambda_{x^*}^{i,d}) a_i^T d, \tag{20}$$

where $\lambda_{x^*}^{i,d}$ is defined by (18). Hence,

$$f(x^* + \alpha d) \geq f(x^*),$$

by hypothesis, and thus d is not a descent direction. $\Box$

Notice that these optimality conditions encompass the Kuhn-Tucker (KT) conditions for linear programming in the following sense: the KT conditions consist of m multiplier rules determining the behaviour of the objective function over *one* cell (which corresponds to the feasible region). [Take g_{x^*} here to be the gradient of the objective function, $\mathcal{A}(x^*)$ to index the active constraints, and keep only the m multiplier rules $u_i^* \geq 0$, for all $i \in \mathcal{A}(x^*)$.] Theorem 2 describes how f behaves over the 2^m cells in the neighbourhood of x^* with only $2m$ multiplier rules.

3.3 The Algorithm—Continuous Case

We now present an algorithm for minimizing a continuous piecewise linear function, $f : I\!R^n \to I\!R$. We assume that f is decomposable at each iterate and at each breakpoint encountered in the line search. Moreover, we assume that at every iterate which is a dead point, the gradients of the activities are linearly independent.

Let $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ be the ridges of f, where $\mathcal{R}$ is a finite index set, and g_{x^k}, $\{\nu_{x^k}^i\}_{i \in \mathcal{A}(x^k)}$ be the decomposition of f at x^k.

Continuous Piecewise Linear Minimization Algorithm

Step 1: Choose any $x^1 \in I\!R^n$ and set $k \leftarrow 1$.

Repeat:
Begin

Step 2: Identify the activities, $\mathcal{A}(x^k)$, and compute $d^k \equiv -P(g_{x^k})$, the projection of the restricted gradient onto the space orthogonal to the gradients of the activities.

If $d^k = \vec{0}$ (x^k is a dead point. Compute a single-dropping descent direction or establish optimality.), then

Step 3: Compute $\{u_i\}_{i \in \mathcal{A}(x^k)}$, the coefficients of $\{a_i\}_{i \in \mathcal{A}(x^k)}$ in the linear combination of g_{x^k} in terms of the columns of $A(x^k)$.

Step 4: If $u_i < 0$ or $u_i > -\nu_{x^k}^i$, for some $i \in \mathcal{A}(x^k)$ (violated optimality condition), then

Step 5: (Drop activity i) Redefine $d^k = P_{-i}(a_i)$, if the violated inequality found corresponds to $u_i \geq 0$, otherwise $d^k = -P_{-i}(a_i)$ and $u_i \leq -\nu_{x^k}^i$, where P_{-i} is the orthogonal projector onto the space orthogonal to the gradients of all the activities but activity i.

Else, stop: x^k is a local minimum of f.

Endif

Endif

Step 6: (Line search) Determine the step size α^k by (approximately) solving $\min_{\alpha > 0} f(x^k + \alpha d^k)$. This line search can be done from x^k, moving from one breakpoint of f to the next, in the direction d^k, until either we establish unboundedness of the objective function or the value of f starts increasing.

Step 7: Update $x^{k+1} = x^k + \alpha^k d^k$, $k \leftarrow k + 1$.
End

Remarks:

- In practice, the orthogonal projectors P and P_{-i} are computed with a suitable factorization and/or update of the matrix of activities, $A(x^k)$.
- In order to compute the coefficients $\{u_i\}_{i\in\mathcal{A}(x^k)}$ in step 3, one needs to solve a linear system which is possibly overdetermined but always feasible, since $g_{x^k} \in \mathcal{R}(A(x^k))$ at a dead point x^k. This can most easily be done by solving the least squares problem:

$$\min_u \|g_{x^k} - A(x^k)u\|$$

(where the vector u is indexed by $\mathcal{A}(x^k)$), since we already have a factorization of the matrix of activities.

- In step 5, when $u_i < 0$ or $u_i > -\nu^i_{x^k}$, we can show that $d^k = \pm P_{-i}(a_i)$ is the *steepest* descent direction in the space $\mathcal{N}(A^T_{-i})$.
- In step 6, we update the directional derivative of the objective function in the direction d^k from one breakpoint to the other. For example, if d^k was obtained from step 2, then the directional derivative of f at x^k in the direction d^k is

$$\begin{aligned} f'(x^k; d^k) &= g^T_{x^k} d^k = -g^T_{x^k} P(g_{x^k}) \\ &= -g^T_{x^k} P^T P(g_{x^k}) = -\|d^k\|^2 \end{aligned} \tag{21}$$

(using the idempotence and the symmetry of the orthogonal projector P). In the case where the descent direction is obtained from step 5, dropping activity i, we have

$$f'(x^k; d^k) = (u_i + \lambda^{i,d^k}_{x^k}) a^T_i d^k$$

(where $\lambda^{i,d^k}_{x^k}$ is defined by (18)). If we then encounter a breakpoint $\bar{x}$ at which f is decomposable, with decomposition $g_{\bar{x}}$, $\{\nu^i_{\bar{x}}\}_{i\in\mathcal{A}(\bar{x})}$, in the line search when "crossing" exactly one ridge $j \in \mathcal{A}(\bar{x})$ (i.e $\mathcal{A}(\bar{x}) \setminus \mathcal{A}(x^k) = \{j\}$), then the directional derivative becomes

$$\begin{aligned} f'(\bar{x}; d^k) &= (g_{\bar{x}} + \lambda^{j,d^k}_{\bar{x}} a_j)^T d^k \\ &= (g_{\bar{x}} + \lambda^{j,d^k}_{\bar{x}} a_j)^T d^k + \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k - \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k \\ &= (g_{\bar{x}} + \lambda^{j,-d^k}_{\bar{x}} a_j)^T d^k + \lambda^{j,d^k}_{\bar{x}} a^T_j d^k - \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k \\ &= -(g_{\bar{x}} + \lambda^{j,-d^k}_{\bar{x}} a_j)^T (-d^k) + \lambda^{j,d^k}_{\bar{x}} a^T_j d^k - \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k \\ &= -f'(\bar{x}; -d^k) + \lambda^{j,d^k}_{\bar{x}} a^T_j d^k - \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k \\ &= f'(x^k; d^k) + \lambda^{j,d^k}_{\bar{x}} a^T_j d^k - \lambda^{j,-d^k}_{\bar{x}} a^T_j d^k \qquad (22) \\ &= f'(x^k; d^k) - \nu^j_{\bar{x}} |a^T_j d^k|. \end{aligned}$$

Equality (22) follows immediately since f is piecewise linear. If at the first breakpoint, $\bar{x}$, we cross a *set* $\mathcal{J}$ of many ridges (i.e $\mathcal{A}(\bar{x}) \setminus \mathcal{A}(x^k) = \mathcal{J}$), then the update is done similarly:

$$f'(\bar{x}; d^k) = f'(x^k; d^k) - \sum_{j\in\mathcal{J}} \nu^j_{\bar{x}} |a^T_j d^k|. \tag{23}$$

In order to know whether the value of f starts increasing beyond breakpoint $\bar{x}$, we check the sign of $f'(\bar{x}; d^k)$, the directional derivative at $\bar{x}$ in the direction d^k.

Note that a *full* ordering of the step sizes corresponding to each breakpoint along the line search is not necessary. We need to have access to these step sizes one at a time, in order of increasing size, only until the value of f starts increasing. We use the "smallest-in/first-out" mechanism of a *heap* (e.g. see [45]).

When studying the uncapacitated facility location problem, Conn and Cornuéjols [22] had to minimize the continuous piecewise linear convex objective function

$$f(x) = \sum_{i \in I} x_i + \sum_{j \in J} [\sum_{i \in I} (c_{ij} - x_i)^+ - f_j]^+,$$

where $a^+ \equiv \max(0, a)$ and c_{ij} and f_j are constants. We are here in the presence of "nested ridges" (the derivative of f is not defined over $\{x : r(x) \equiv \sum_{i \in I}(c_{ij} - x_i)^+ - f_j = 0\}$, where the function r has ridges, namely: $c_{ij} - x_i,\ i \in I$). They derived optimality conditions which are specific to this problem in order to construct an algorithm. To illustrate how this problem fits our general framework, consider the following simple instance: $f(x, y) = [(-x)^+ + (-y)^+ - 1]^+$. Figure 4 shows the values f takes over the different cells. Figure 5 gives the decomposition, $g_{\hat{x}}$, $\{\nu^i_{\hat{x}}\}_{i \in \mathcal{A}(\hat{x})}$, of f at every point where the function is decomposable (i.e. every point of $I\!R^2 \setminus \{(0, -1)^T, (-1, 0)^T\}$). The circles in Figure 5 contain the different values the restricted gradient, $g^T_{\hat{x}}$, takes over the different subdomains where f is decomposable. In a region over which f is differentiable, $g_{\hat{x}}$ is just the gradient of the restriction of f to that region. The squares contain the ν^i values for each segment of ridge i (with $\nu^i = 0$ for the dashed line part of the ridge). Thus, apart from the two points $(0, -1)^T$ and $(-1, 0)^T$ (at which the gradients of the activities are linearly dependent), we have a decomposition of f at any point of the plane. At the two degenerate points $(0, -1)^T$ and $(-1, 0)^T$, one can easily show (we shall prove the "non-decomposability" of a function in Section 6.2) that f is not decomposable (it will also be possible to see this using the characterization of decomposable functions given by (37)). Thus, special attention will have to be paid at such singular points.

3.4 Finite Step Convergence

Theorem 3 (Convergence) *Let $f : I\!R^n \rightarrow I\!R$ be a continuous piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set. Assume that*

1. *$\{a_i\}_{i \in \mathcal{A}(x^k)}$ are linearly independent at each iterate, x^k, encountered in the course of the algorithm, and*

2. *f is decomposable at each breakpoint encountered along the line searches.*

Then the continuous piecewise linear minimization algorithm converges globally (i.e. from any starting point) in a finite number of iterations.

Proof: Starting at any point, the algorithm generates a sequence of points, each of which is obtained from its predecessor using step 7. The step size α^k in step 6 is chosen

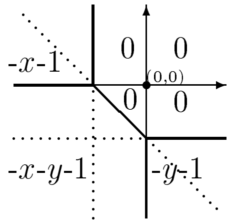

Figure 4: Simple instance of objective function for the uncapacitated facility location problem

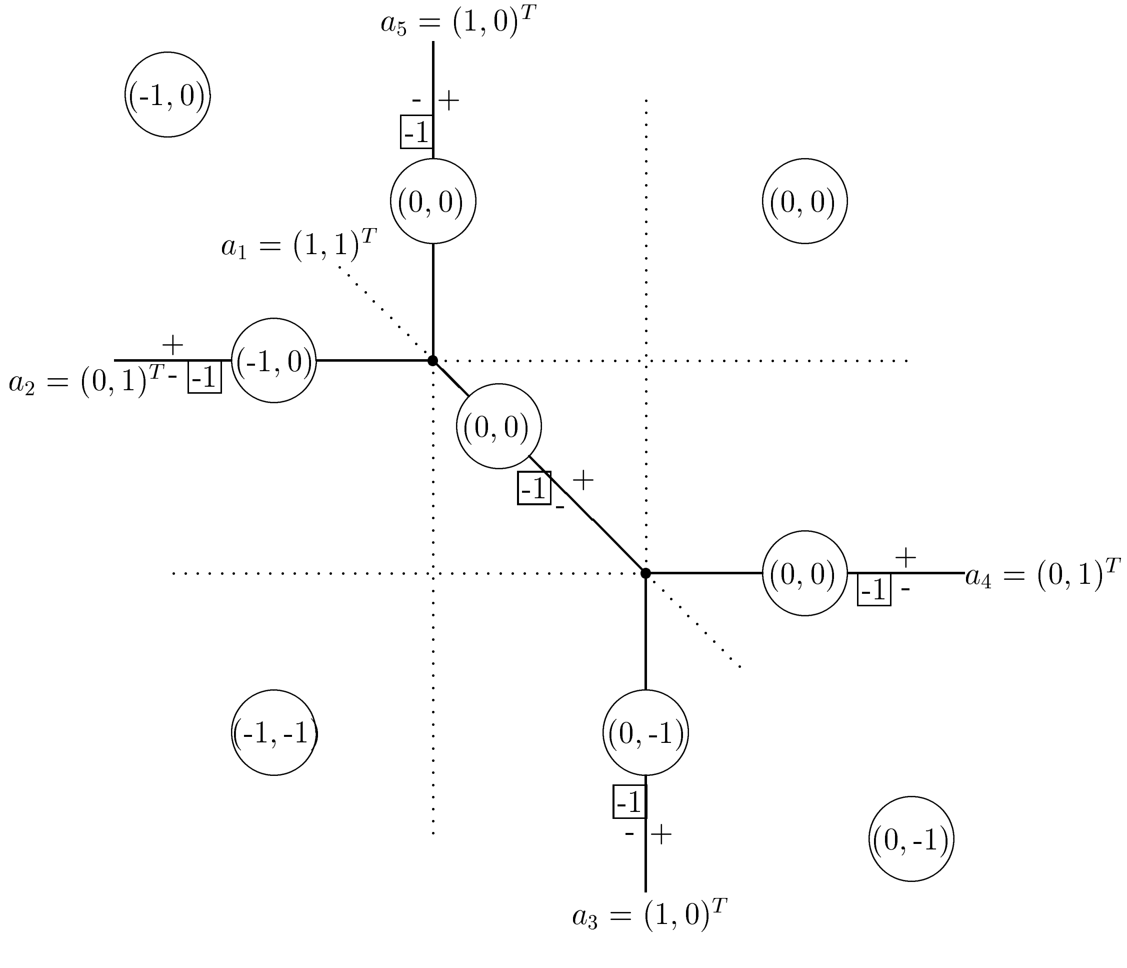

Figure 5: Example of a decomposition

so that a ridge that was not active at x^k is now active at x^{k+1}. If iterate x^k is not a dead point, then the direction vector d^k for generating x^{k+1}, the succeeding point, will be chosen at step 2. This will guarantee that all ridges which are active at x^k will be active at x^{k+1}. Hence, in this case

$$\mathcal{A}(x^k) \subset \mathcal{A}(x^{k+1}).$$

Thus, under the assumption of the linear independence of the gradients of the activities, the sequence step 2–step 6–step 7 cannot be executed consecutively more than n times.

Again by the linear independence hypothesis, when x^k is a dead point, the descent direction will drop only *one* activity and hence

$$|\mathcal{A}(x^k)| \leq |\mathcal{A}(x^{k+1})|$$

(at x^{k+1} there is exactly one activity fewer than at x^k, but at least one new activity was hit). By definition of a dead point, we have, for α small enough:

$$f(x^k + \alpha p) = f(x^k), \quad \text{for all } p \in \mathcal{N}(A(x^k)^T)$$

and hence, $f(x)$ is constant throughout

$$\{x : \sigma(x) = \sigma(x^k)\}.$$

The sequence of points generated by the algorithm determine a monotonic decreasing sequence of values of f. This means that one cannot return to a point x such that $\sigma(x) = \sigma(x^k)$. Since there is only a finite number of different vectors $\sigma \in 3^{|\mathcal{R}|}$, we may conclude that only a finite number of dead points can be generated. On the other hand, if the original problem is not unbounded, then by the optimality conditions theorem (Theorem 2) some dead point is optimal. The algorithm will not terminate until an optimal dead point has been found or a conclusion of unboundedness is reached. □

4 Discontinuous Piecewise Linear Optimization

We now consider the case where, again, f is piecewise linear (with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set) but with possibly discontinuities across some ridges. We shall term such ridges: *faults* and $\mathcal{F}(\hat{x})$ will denote the faults which are active at $\hat{x}$.

A (local) minimum does not always exist in the discontinuous case. Consider for example the following univariate function, having $x = 0$ as a fault:

$$f(x) = \begin{cases} x + 1 & \text{if } x \geq 0 \\ -x & \text{otherwise.} \end{cases} \tag{24}$$

Hence, we shall rather look for a local *infimum*. In order to find such a local infimum of a function f having some faults, we shall simply generalize the algorithm for the continuous problem by implicitly considering any discontinuity or *jump* across a fault i in f as the limiting case of a continuous situation.

Since we are looking for a local *infimum* of a given function f, it is equivalent to work rather with the function $\underline{f}$ defined by

$$\underline{f}(x) \equiv \liminf_{\bar{x} \to x} f(\bar{x}).$$

Thus we need only to look for a local *minimum* of $\underline{f}$. This convention will simplify the exposition. Without loss of generality, we shall henceforward only consider functions f such that $f(x) = \underline{f}(x)$ (in other words, we consider the *lower semicontinuous envelope* of f).

4.1 Soaring Directions and Faults

The algorithm will be essentially the same as in the continuous case except that we consider dropping an active fault from a dead point, x, only if we do so along a direction d such that

$$\lim_{\delta \to 0^+} f(x + \delta d) = f(x)$$

(i.e. as $\delta > 0$ is small, the value of f does not *jump up* from x to $x + \delta d$). Thus, virtually only step 4 must be adapted from the continuous problem algorithm in order to solve the discontinuous case. Note that in step 4, there is no single-dropping direction that corresponds to "jumping down" since by our convention ($f = \underline{f}$) we have

$$f(x^k) \equiv \liminf_{x \to x^k} f(x).$$

To make more rigorous the intuitive concept of directions jumping up or down, we define the set of *soaring directions* from a point $\hat{x}$ to be:

$$S(\hat{x}) \equiv \{d \in I\!R^n : \exists \epsilon > 0, \bar{\delta} > 0 \text{ such that } \forall\ 0 < \delta < \bar{\delta},\ f(\hat{x} + \delta d) - f(\hat{x}) > \epsilon\}.$$

If we define, for a non-degenerate point $\hat{x}$,

$$\mathcal{S}^+(\hat{x}) \equiv \{i \in \mathcal{A}(\hat{x}) :\ \text{if } d^{i^+} \in \mathcal{N}(A_{-i}^T) \text{ and } a_i^T d^{i^+} > 0 \text{ then } d^{i^+} \in S(\hat{x})\}$$

and

$$\mathcal{S}^-(\hat{x}) \equiv \{i \in \mathcal{A}(\hat{x}) :\ \text{if } d^{i^-} \in \mathcal{N}(A_{-i}^T) \text{ and } a_i^T d^{i^-} < 0 \text{ then } d^{i^-} \in S(\hat{x})\},$$

then the set of soaring single-dropping directions from $\hat{x}$ are simply the directions dropping an activity $i \in \mathcal{S}^+(\hat{x})$ positively and the directions dropping an $i \in \mathcal{S}^-(\hat{x})$ negatively.

A fault can now be defined more rigorously: a *positive (negative) fault of f at a point $\hat{x}$* is a ridge $i \in \mathcal{R}$ such that for any neighbourhood, $B(\hat{x})$, of $\hat{x}$, there exists a nondegenerate point $x' \in B(\hat{x})$ with $i \in \mathcal{S}^+(x')$ (with $i \in \mathcal{S}^-(x')$). The set of all positive (negative) faults at $\hat{x}$ is denoted by $\mathcal{F}^+(\hat{x})$ ($\mathcal{F}^-(\hat{x})$). The set of *faults of f at a point $\hat{x}$* is denoted by

$$\mathcal{F}(\hat{x}) \equiv \mathcal{F}^+(\hat{x}) \cup \mathcal{F}^-(\hat{x}).$$

[Note that the definition of a fault $i \in \mathcal{F}(\hat{x})$ is described via *single*-dropping directions, dropping activity i, in order to ensure that the jump is indeed "caused" by activity i (and is not due to some other fault which would also be active at $\hat{x}$). Also, $\mathcal{F}^+$ and $\mathcal{F}^-$ are defined in terms of soaring directions in a *neighbourhood* of $\hat{x}$ and not at $\hat{x}$ alone. Moreover, we need to refer to some point x' in a neighbourhood of $\hat{x}$, so that the definition makes sense at a degenerate point $\hat{x}$—recall that a single-dropping direction does not necessarily exist from a degenerate point].

We would like to modify the continuous problem algorithm in such a way that, at a non-degenerate dead point, x^k, we do not need to verify the optimality conditions corresponding to soaring single-dropping directions ($u_i \geq 0$, $i \in \mathcal{S}^+(x^k)$ and $u_i \leq -\nu^i$, $i \in \mathcal{S}^-(x^k)$), so that we would never consider such single-dropping directions in order to establish whether x^k is optimal. This is reasonable since we are looking for a *local* minimum. The line-search step (step 6) will be modified similarly: when we encounter a breakpoint $\bar{x}$ on a fault along a direction $d \in S(\bar{x})$ (jump up), we stop; while if d is such that $-d \in S(\bar{x})$, (jump down), we carry on to the next breakpoint, and update properly the directional derivative along d. We first need to define a decomposition of a discontinuous function at a point on a fault.

4.2 The Decomposition

In the non-degenerate case, we can always find a decomposition of a continuous piecewise linear function f such that the change of the nonsmooth part of f in a single-dropping direction, d, can be written as the dot product of d with a multiple of the gradient of the activity to be dropped. The proof of this fact was based on the continuity of f, as we saw in the proof of the decomposition theorem (Theorem 1). This is what enabled us to isolate the effect of each activity, yielding an optimality condition corresponding to each of the possible single-dropping directions. Apart from the fact that it relies on the scalars $\{u_i\}_{i\in\mathcal{A}}$, each of these conditions was independent of the activities other than the one being dropped. Nevertheless, in the discontinuous case, when $a_i^T x - b_i$ is a fault at a non-degenerate point, $\hat{x}$, there is in general also a way to define a decomposition, $g, \{\nu^i\}_{i\in\mathcal{A}}$, of f at $\hat{x}$ so that when we drop $i \notin \mathcal{S}^+(\hat{x})$ positively or $i \notin \mathcal{S}^-(\hat{x})$ negatively (i.e. in a non-soaring single-dropping direction), the change of the nonsmooth part of f can be expressed as the dot product of the single-dropping direction with a multiple of a_i.

Note that if $i \in \mathcal{R}$ is a positive fault at some point $\hat{x}$, and a negative fault at some other point $\tilde{x}$ on the same segment (i.e. $\sigma(\hat{x}) = \sigma(\tilde{x})$), R, then, in practice, we shall consider R as divided further into two segments. To do so, we can introduce another (*artificial*) ridge, $a_0^T x - b_0$, the sign of whose residuals will be used in order to know to which segment belongs an iterate $x^k \in R$. Note that one has to be careful at a "contact" point $x_c \in I\!R$ (defined below) such that $a_0^T x - b_0 = 0$. At x_c, contrary to at some other points of R, we can drop activity i *both* positively and negatively.

The function $f : I\!R^2 \to I\!R$, given by

$$f(x) = \begin{cases} 2x_2 & \text{if } x_1 > 0 \text{ or } (x_1 = 0 \text{ and } x_2 \leq 0), \\ -x_2 & \text{otherwise,} \end{cases} \tag{25}$$

illustrates well the situation. Figure 6 shows the graph of f in a neighbourhood of $x_c \equiv (0,0)^T$ (the dotted lines are simply lines that could be seen if the hatched surface were

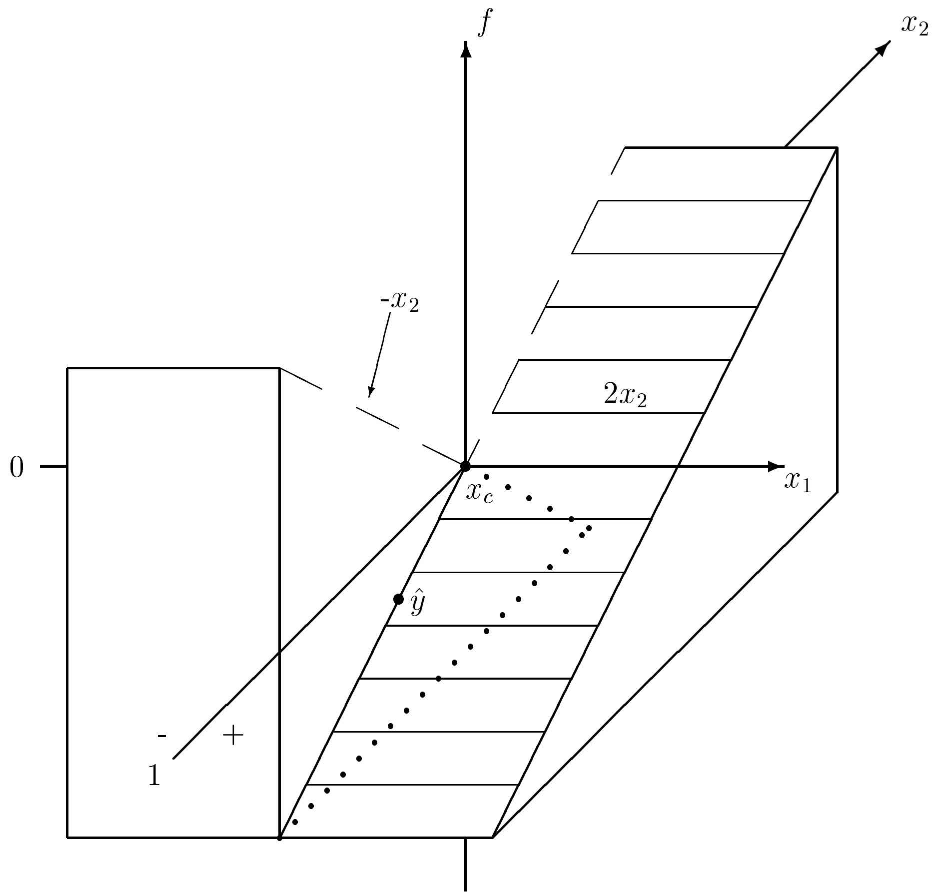

Figure 6: Graph of a function having a contact point x_c

transparent). We introduce the artificial ridge

$$a_0^T x - b_0 \equiv x_2 = 0,$$

so that the fault $x_1 = 0$ is partitioned into three segments according to whether $a_0^T x - b_0$ is positive, negative or zero. The point x_c is a contact point with respect to the fault $x_1 = 0$. (The choice of a particular vector a_0 for an artificial ridge is rather arbitrary but necessary.)

Formally, we define $x_c \in I\!R^n$ to be a *contact point of f with respect to $i \in \mathcal{A}(x_c)$*, when $i \in \mathcal{F}(x_c)$ such that either

1. $i \in \mathcal{F}^+(x_c) \cap \mathcal{F}^-(x_c)$, or
2. there exist $\sigma^+, \sigma^- \in 3^{|\mathcal{R}|}$ such that $\sigma_i^+ = 1$, $\sigma_i^- = -1$ and

   $$\lim_{x \to x_c, \sigma_i(x) = \sigma^+} f(x) = \lim_{x \to x_c, \sigma_i(x) = \sigma^-} f(x)$$

   (continuity when crossing ridge i, which is a fault, at x_c).

We term $\hat{x} \in I\!R^n$ a *non-contact point* of f when $\hat{x}$ is not a contact point of f with respect to any $i \in \mathcal{A}(x_c)$.

Note that the fault $x_1 = 0$ and the point $x_c = (0,0)^T$ satisfy both conditions 1 and 2 in the above definition of a contact point for the function f defined by (25). (To see that the ridge $x_1 = 0$ is both in $\mathcal{F}^+(x_c)$ and in $\mathcal{F}^-(x_c)$, observe that the ridge $x_1 = 0$ is in $\mathcal{S}^+(x')$ and also in $\mathcal{S}^-(x'')$, for any $\epsilon > 0$, where $x' \equiv x_c + (0, \epsilon)^T$ and $x'' \equiv x_c + (0, -\epsilon)^T$.) They however satisfy only condition 1 for the function $f : I\!R^2 \to I\!R$, defined by

$$f(x) = \begin{cases} 1 & \text{if } x_1 \geq 0 \text{ and } x_2 \geq 0, \\ 2 & \text{if } x_1 \leq 0 \text{ and } x_2 \leq 0 \text{ and } x \neq (0,0)^T, \\ 3 & \text{if } x_1 > 0 \text{ and } x_2 < 0, \\ 4 & \text{otherwise}, \end{cases}$$

with faults $x_1 = 0$ and $x_2 = 0$. For the function $f : I\!R^2 \to I\!R$ given by

$$f(x) = \begin{cases} -x_2 & \text{if } x_1 > 0 \text{ and } x_2 < 0, \\ 0 & \text{otherwise}, \end{cases}$$

with fault $x_1 = 0$, we satisfy only condition 2 ($\mathcal{F}^-(x_c)$ is empty).

For the function f defined by (25), we have the following trivial "decomposition" (we shall define it shortly for the discontinuous case) at the point $\hat{y}$ on the fault $x_1 = 0$ (see Figure 6) such that $a_0^T \hat{y} - b_0$ is negative (recall that $a_0 = (0,1)^T$ and $b_0 = 0$):

$$f(x) = f(\hat{y}) + g_{\hat{y}}^T (x - \hat{y}) + \nu_{\hat{y}}^1 \min(0, a_1^T (x - \hat{y})), \tag{26}$$

for all $x \notin S(\hat{y}) + \hat{y}$, where

$$a_1^T = (1, 0), \; g_{\hat{y}} = (0, 2)^T \text{ and } \nu_{\hat{y}}^1 = 0.$$

Since $\{x \in I\!R^2 : x \notin S(\hat{y}) + \hat{y}\}$ is simply the set of points x satisfying: $a_1^T x - b_1 \geq 0$ (recall that $b_1 = 0$), the scalar $\nu_{\hat{y}}^1$ is clearly meaningless and superfluous (since in (26), $\min(0, a_1^T(x - \hat{y}))$ is always zero for such points). This "decomposition" simply gives the behaviour of f (equation (26)) for every point x in the neighbourhood of $\hat{y}$ such that $x - \hat{y}$ is not a soaring direction from $\hat{y}$. An algorithm similar to the one introduced in the continuous case, but which does not consider *soaring* single-dropping directions, will encounter no difficulty with the discontinuity in f at any non-contact point (i.e. for (25) at any point other than x_c). We shall see that the fact that the decomposition is undefined for the x's such that $x - \hat{x}$ is a soaring direction from a point $\hat{x}$, will not affect our method (we never consider the soaring single-dropping directions). The ridge $x_1 = 0$ is a negative fault of f (Figure 6) at the point $\hat{y}$. It can only possibly be dropped positively (negatively we would "jump up" the fault). Thus, if $\hat{y}$ were a dead point, we would only verify whether the optimality condition $u_1 \geq 0$ is satisfied in order to see if dropping the fault $x_1 = 0$ yields descent. Note that at the contact point, $x_c = (0, 0)$, the fault could be dropped either positively or negatively as none of the two single-dropping directions, $(1, 0)^T$ and $(-1, 0)^T$, from x_c are a soaring direction. At a point $\hat{z}$ on the fault such that $a_0^T \hat{z} - b_0$ is positive, we can also "decompose" f exactly as in (26), with, this time, $g_{\hat{z}} = (0, -1)^T$, if we defined ridge 1 to be such that $a_1^T = (-1, 0)$.

We assume that a (possibly) discontinuous piecewise linear function, f, is given using the same form as in the continuous case, except that here, we assume moreover that at any point $\hat{x} \in I\!R^n$, we can obtain $\mathcal{F}^+(\hat{x})$ and $\mathcal{F}^-(\hat{x})$, the set of the positive faults and the set of the negative faults of f at $\hat{x}$.

Without loss of generality, we shall henceforward assume that at a *given non-contact point* $\hat{x}$, all the faults of f at $\hat{x}$ are negative faults at $\hat{x}$ (otherwise we could replace a_i by $-a_i$ and b_i by $-b_i$), that is to say $\mathcal{F}(\hat{x}) = \mathcal{F}^-(\hat{x})$.

We now redefine the notion of decomposition at a point $\hat{x}$ to take into account the possibility of having $\hat{x}$ on a fault.

Definition 2 *Let $f : I\!R^n \rightarrow I\!R$ be a (possibly discontinuous) piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in I\!R^n$. Let $g_{\hat{x}} \in I\!R^n$ and $\psi_{\hat{x}}$ be a function defined on $I\!R^n$ such that we have:*

$$f(x) = f(\hat{x}) + g_{\hat{x}}^T (x - \hat{x}) + \psi_{\hat{x}}(x), \tag{27}$$

for all $x \in B(\hat{x})$ such that $a_i^T(x - \hat{x}) \geq 0$, $i \in \mathcal{F}(\hat{x})$, for some neighbourhood, $B(\hat{x})$, of $\hat{x}$, where

$$\psi_{\hat{x}}(x) = \sum_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})} \nu_{\hat{x}}^i \min(0, a_i^T(x - \hat{x}))$$

for some scalars $\{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})}$.

We say that $g_{\hat{x}}, \{\nu_{\hat{x}}^i\}_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})}$ is a decomposition of f (into a smooth function and a sum of functions having a single ridge) *at $\hat{x}$.*

Based on this definition of the decomposition of a discontinuous piecewise linear function f, we say again that f is *decomposable* at $\hat{x}$ when there exists a decomposition of f at $\hat{x}$.

To illustrate, consider the discontinuous function $f : I\!R^2 \to I\!R$:

$$f(x) = \min(0, x_1 - x_2) + |x_2| + \begin{cases} x_2 & \text{if } x_1 \geq 0, \\ -x_2 + 1 & \text{otherwise,} \end{cases}$$

which has the set of ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R} \equiv \{1, 2, 3\}$,

$$a_1 = (1, -1)^T, \quad a_2 = (0, 1)^T, \quad a_3 = (1, 0)^T, \quad b_1 = b_2 = b_3 = 0.$$

Here is a decomposition of this function at $\hat{x} \equiv (0, 0)^T$ (note that $\mathcal{F}(\hat{x}) = \mathcal{F}^-(\hat{x}) = \{3\}$):

$$f(x) = g_{\hat{x}}^T x + \sum_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})} \nu^i \min(0, a_i^T x)$$

for all x in some small enough neighbourhood of $\hat{x}$ and such that $a_i^T(x - \hat{x}) \geq 0, \ i \in \mathcal{F}(\hat{x})$ (i.e. such that $x_1 \geq 0$), where

$$g_{\hat{x}} = (0, 2)^T \quad \nu^1 = 1 \ \text{ and } \ \nu^2 = -2.$$

The decomposition theorem (Theorem 1) and Proposition 1 proved in the continuous case still hold at non-contact points when referring to definition 2.

Theorem 4 (Decomposition—Discontinuous Case) *Let $f : I\!R^n \to I\!R$ be a (possibly discontinuous) piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in I\!R^n$ be such that $\mathcal{F}^+(\hat{x}) \cap \mathcal{F}^-(\hat{x}) = \emptyset$. Without loss of generality, assume that $\mathcal{F}(\hat{x}) = \mathcal{F}^-(\hat{x})$. If $\{a_i\}_{i \in \mathcal{A}(\hat{x})}$, the gradients of the ridges of f which are active at $\hat{x}$, are linearly independent, then f is decomposable at $\hat{x}$ and the decomposition is unique.*

Proof: The proof goes exactly as the proof of the continuous case (Theorem 1), except for the fact that we are now dealing with the points x in a neighbourhood of $\hat{x}$ such that $a_i^T(x - \hat{x}) \geq 0, \ i \in \mathcal{F}(\hat{x})$, and the $c^J \in I\!R^n$ are given for each $J \subseteq \mathcal{A}(\hat{x})$ such that $\mathcal{F}(\hat{x}) \subseteq J$. We use the continuity of f over

$$\{x \in N(\hat{x}) : a_i^T(x - \hat{x}) > 0, i \in \mathcal{F}(\hat{x})\}$$

to obtain: for $I, J \subseteq \{1, 2, \ldots, m\}$ such that $\mathcal{F}(\hat{x}) \subseteq I \cap J$,

$$d_i^I = d_i^J \ \text{ whenever } i \in I \cap J \text{ or } i \in \{1, 2, \ldots, m\} \setminus (I \cup J) \text{ or } m < i \leq n,$$

which is true if and only if for $I \subseteq \{1, 2, \ldots, m\}$ such that $\mathcal{F}(\hat{x}) \subseteq I$, and $1 \leq i \leq n$:

$$d_i^I = \begin{cases} d_i^{\{1,2,\ldots,m\}} & \text{if } i \in I, \\ d_i^{\mathcal{F}(\hat{x})} & \text{otherwise.} \end{cases}$$

Hence, for all $y \in I\!R^n$ such that $y_i \geq 0, \ i \in \mathcal{F}(\hat{x})$, we have

$$h(y) = \sum_{i=1}^n d_i y_i,$$

where

$$d_i = \begin{cases} d_i^{\{1,2,\ldots,m\}} & \text{if } y_i \geq 0, \\ d_i^{\mathcal{F}(\hat{x})} & \text{if } y_i < 0. \end{cases}$$

Thus, as in the proof of Theorem 1, we obtain, with

$$\nu^i \equiv d_i^{\mathcal{F}(\hat{x})} - d_i^{\{1,2,\ldots,m\}},\ i \in \{1,2,\ldots,m\} \setminus \mathcal{F}(\hat{x}) :$$

$$f(x) = x^T c^{\{1,2,\ldots,m\}} + \sum_{i \in \{1,2,\ldots,m\} \setminus \mathcal{F}(\hat{x})} \nu^i \min(0, a_i^T x), \tag{28}$$

for all x in a neighbourhood of $\hat{x}$ such that $a_i^T(x - \hat{x}) \geq 0,\ i \in \mathcal{F}(\hat{x})$.

Hence, (28) is a decomposition of f at $\hat{x}$.

The proof of the uniqueness follows exactly as in of Theorem 1. □

Proposition 2 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a (possibly discontinuous) piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in \mathbb{R}^n$ be such that $\mathcal{F}^+(\hat{x}) \cap \mathcal{F}^-(\hat{x}) = \emptyset$. Without loss of generality, assume that $\mathcal{F}(\hat{x}) = \mathcal{F}^-(\hat{x})$. Assume that $\{a_i\}_{i \in \mathcal{A}(\hat{x})}$, the gradients of the ridges of f which are active at $\hat{x}$, are linearly independent and that, without loss of generality, f has the following form for all x such that $a_i^T(x - \hat{x}) \geq 0,\ i \in \mathcal{F}(\hat{x})$, in a neighbourhood of $\hat{x}$:*

$$f(x) = f(\hat{x}) + (x - \hat{x})^T c^J$$

whenever

$$\begin{array}{rrcl} & a_i^T(x - \hat{x}) & \geq & 0,\ \ i \in J \\ \textit{and} & a_i^T(x - \hat{x}) & < & 0,\ \ i \in \mathcal{A}(\hat{x}) \setminus J, \end{array}$$

for some $c^J \in \mathbb{R}^n$ given for each $J \subseteq \mathcal{A}(\hat{x})$ such that $\mathcal{F}(\hat{x}) \subseteq J$.

For every $i \in \mathcal{A}(\hat{x})$, let J_i be any subset of $\mathcal{A}(\hat{x})$ such that $i \in J_i$ and $\mathcal{F}(\hat{x}) \subseteq J_i$, and let the scalars $\{\lambda^i\}_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})}$ be defined by

$$\lambda^i a_i = c^{J_i - i} - c^{J_i}.$$

Then, $c^{\mathcal{A}(\hat{x})}, \{\lambda^i\}_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})}$ is the decomposition of f at $\hat{x}$.

Proof: We adapt straightforwardly the proof of Proposition 1. Since f is decomposable at $\hat{x}$, there exists some decomposition $g, \{\nu^i\}_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})}$ of f at $\hat{x}$. Hence, for any $\mathcal{F}(\hat{x}) \subseteq J \subseteq \mathcal{A}(\hat{x})$ we have

$$c^J = g + \sum_{k \in \mathcal{A}(\hat{x}) \setminus J} \nu^k a_k,$$

since

$$f(x) = f(\hat{x}) + g_{\hat{x}}^T(x - \hat{x}) + \sum_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})} \nu^i \min(0, a_i^T(x - \hat{x}))$$

for all $x \in B(\hat{x})$ such that $a_i^T(x - \hat{x}) \geq 0,\ i \in \mathcal{F}(\hat{x})$, for some neighbourhood, $B(\hat{x})$, of $\hat{x}$. Thus, we can show that this implies $c^{J-i} - c^J = \lambda^i a_i$, for any $\mathcal{F}(\hat{x}) \subseteq J \subseteq \mathcal{A}(\hat{x})$ such that $i \in J \setminus \mathcal{F}(\hat{x})$.

We then show that for an arbitrary point, $\tilde{x}$, of a small neighbourhood of $\hat{x}$ such that $a_i^T(\tilde{x}-\hat{x}) \geq 0$, $i \in \mathcal{F}(\hat{x})$, we have:

$$f(\tilde{x}) = f(\hat{x}) + (\tilde{x}-\hat{x})^T c^{\mathcal{A}(\hat{x})} + \sum_{i \in \mathcal{A}(\hat{x}) \setminus \mathcal{F}(\hat{x})} \lambda^i \min(0, a_i^T(\tilde{x}-\hat{x})).$$

Let

$$K \equiv \{i \in \mathcal{A}(\hat{x}) : a_i^T(\tilde{x}-\hat{x}) \geq 0\}.$$

Using the fact that $\mathcal{F}(\hat{x}) \subseteq K$, the rest of the proof then follows exactly as in the proof of Proposition 1. $\square$

As in the continuous case, note that the decomposition constructed in the proof is valid for any other point $\tilde{x}$ such that $\sigma(\tilde{x}) = \sigma(\hat{x})$, as long as we assume that artificial ridges were introduced in order to avoid the situation where for a part of a segment we can only possibly drop negatively, while on another part of the same segment we can only possibly drop positively (specifically, we include as many artificial ridges as necessary so that for any given point $\hat{x} \in I\!R^n$, the vector $\sigma(\hat{x})$ determines $\mathcal{F}^+(\hat{x})$ and $\mathcal{F}^-(\hat{x})$).

4.3 The Algorithm—Discontinuous Case

Theorem 5 (Optimality Conditions—Discontinuous Case) *Let $f : I\!R^n \to I\!R$ be a (possibly discontinuous) piecewise linear function with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $x^* \in I\!R^n$ be a non-contact point of f. Assume linear independence of $\{a_i\}_{i \in \mathcal{A}(x^*)}$ and, without loss of generality, $\mathcal{F}(x^*) = \mathcal{F}^-(x^*)$. Let $g_{x^*}, \{\nu_{x^*}^i\}_{i \in \mathcal{A}(x^*) \setminus \mathcal{F}(x^*)}$ be the decomposition of f at x^*. The point x^* is a local minimum of f if and only if there exist scalars u_i^*, $i \in \mathcal{A}(x^*)$ such that*

1. *$g_{x^*} = \sum_{i \in \mathcal{A}(x^*)} u_i^* a_i$ (or, equivalently, the reduced restricted gradient of f at x^* is zero, i.e. x^* is a dead point) and*
2. *For each $i \in \mathcal{A}(x^*)$,*
   (i) $u_i^* \leq -\nu_{x^*}^i$, if $i \notin \mathcal{F}(x^*)$
   (ii) $u_i^* \geq 0$.

When x^* is a dead point, condition 2 means that the directional derivative of f in each single-dropping directions which is not a soaring direction, is non-negative.

Proof: The necessity of the first condition holding follows exactly the same argument as in the continuous version of the optimality condition theorem (Theorem 2). In order to prove the necessity of condition 2 (i), let $i \in \mathcal{A}(x^*) \setminus \mathcal{F}(x^*)$. Consider moving from a dead point x^* to a displaced point $x^* + \alpha d$, where d satisfies $a_k^T d \geq 0$, $k \in \mathcal{F}(x^*)$. We have, for $\alpha > 0$ small enough:

$$f(x^* + \alpha d) = f(x^*) + \alpha d^T [g_{x^*} + \sum_{k \in \mathcal{A}(x^*) \setminus \mathcal{F}(x^*) : a_k^T d < 0} \nu_{x^*}^k a_k].$$

Using the fact that x^* is a dead point and using $d = d^-$, where d^- drops negatively activity i (we indeed have $a_k^T d^- \geq 0$, for all $k \in \mathcal{F}(x^*)$, as $i \notin \mathcal{F}(x^*)$), we obtain

$$f(x^* + \alpha d^-) = f(x^*) - \alpha(u_i^* + \nu_{x^*}^i)|a_i^T d^-|.$$

If we do not have $u_i^* \leq -\nu_{x^*}^i$, then x^* is not a local minimum of f. We prove the necessity of condition 2 (ii), by using rather a direction, d^+, dropping positively activity i.

Suppose now that both conditions 1 and 2 hold. Consider $d \in I\!R^n$, an arbitrary direction. We want to show that d is not a descent direction. This is clear if $d \in S(x^*)$, thus we can assume that $d \notin S(x^*)$. But this implies $a_i^T d \geq 0$, for all $i \in \mathcal{F}(x^*)$: for suppose that there existed a fault $i \in \mathcal{F}(x^*)$ such that $a_i^T d < 0$. Then, by our convention on the definition of the ridges ($\mathcal{F}(x^*) = \mathcal{F}^-(x^*)$) and the fact that $d \notin S(x^*)$, it means that x^* is a contact point with respect to ridge i (take in the definition (part 2) of a contact point, $\sigma^- = \sigma(x^* + \delta d)$ and $\sigma^+ = \sigma(x^* + \delta \hat{d})$, for $\delta > 0$ small enough, and $\hat{d}$ such that $a_k^T \hat{d} \geq 0, k \in \mathcal{F}(x^*)$ and $a_i^T \hat{d} > 0$). This contradicts the hypothesis that x^* is not a contact point. Hence, using the decomposition of f at x^*, we have, for $\alpha > 0$ small enough:

$$\begin{aligned} f(x^* + \alpha d) &= f(x^*) + \alpha d^T [g_{x^*} + \sum_{i \in \mathcal{A}(x^*) \setminus \mathcal{F}(x^*) : a_i^T d < 0} \nu_{x^*}^i a_i] \\ &= f(x^*) + \alpha \sum_{i \in \mathcal{A}(x^*)} (u_i^* + \lambda_{x^*}^{i,d}) a_i^T d, \end{aligned}$$

since x^* is a dead point, where

$$\lambda_{x^*}^{i,d} = \begin{cases} \nu_{x^*}^i & \text{if } a_i^T d < 0 \text{ and } i \notin \mathcal{F}(x^*), \\ 0 & \text{otherwise.} \end{cases}$$

Thus, by hypothesis and using the fact that $a_i^T d \geq 0$ for all $i \in \mathcal{F}(x^*)$, we have

$$f(x^* + \alpha d) \geq f(x^*).$$

□

From these optimality conditions, we derive an algorithm for minimizing a (possibly discontinuous) piecewise linear function, $f : I\!R^n \to I\!R$ which is very similar to that for the continuous case. Let $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ be the ridges of f, where $\mathcal{R}$ is a finite index set. We assume again that f is decomposable at each iterate and at each breakpoint encountered in the line search. Moreover, we assume that at every iterate which is a dead point, the gradients of the activities are linearly independent and that all points encountered in the algorithm are non-contact points. We shall discuss the degenerate situation in Section 6.

We assume, without loss of generality, that at the kth iterate, x^k, $\mathcal{F}(x^k) = \mathcal{F}^-(x^k)$.

Discontinuous Piecewise Linear Minimization Algorithm

The only step of the continuous algorithm which need to be modified is:

"**Step 4:** If $u_i < 0$ for some $i \in \mathcal{A}(x^k)$, or $u_i > -\nu_{x^k}^i$ for some $i \in \mathcal{A}(x^k) \setminus \mathcal{F}(x^k)$ (violated optimality condition), then".

The same finite step convergence result as in the continuous case is valid, except now based on Theorem 5.

5 The Constrained Case

Up to now we concentrated on the *unconstrained* nonsmooth optimization problem: $\inf_x f(x)$, where f is a (possibly discontinuous) piecewise linear function. The practical optimization problems we consider in Section 8 however involve constraints. The approach taken here transforms such constrained optimization problems to unconstrained ones via a penalty function.

5.1 The l_1 Exact Penalty Function

Penalty functions are used to provide a transformation of a nonlinear programming problem to an unconstrained minimization problem. Indeed, one approach to solving the nonlinearly constrained problem

$$\begin{array}{rlcl} \min\limits_{x \in \mathbb{R}^n} & f(x) & & \\ \text{subject to} & f_i(x) & = & 0, \quad i \in E \\ & f_i(x) & \geq & 0, \quad i \in I \end{array} \tag{29}$$

(where the functions f and $\{f_i\}_{i \in E \cup I}$ are continuously differentiable and the index sets E and I are finite and disjoint) is to construct an *exact penalty function* f_γ, i.e. a function minimized locally by a solution x^* to the above constrained problem, provided $\gamma > 0$ is suitably chosen. The most popular (nondifferentiable) exact penalty function is the *absolute value* or l_1 *exact penalty function*:

$$f_\gamma(x) \equiv \gamma f(x) + \sum_{i \in E} |f_i(x)| - \sum_{i \in I} \min[0, f_i(x)], \;\; \gamma > 0. \tag{30}$$

Under mild conditions (see [30], page 143) in the case of a convex problem (cf. [28] for the more general case), there is a threshold value $\bar{\gamma} > 0$ such that for any $0 < \gamma < \bar{\gamma}$, an unconstrained local minimum of f_γ is a local minimum for (29).

With such a penalty approach, we wish to minimize, as a subproblem, the nondifferentiable function f_γ for a given *penalty parameter* value $\gamma > 0$.

A point x^* is termed a *first-order minimizer* of the exact penalty function f_γ when, for all directions $d \in \mathbb{R}^n$, we have

$$f'_\gamma(x^*; d) \geq 0.$$

Note that this directional derivative exists by properties of (30).

Once we find out that one of the two following situations holds, then we decrease γ:

Situation 1: The current iterate is a local optimum of f_γ which is infeasible (with respect to the original problem).

Situation 2: We have found an infeasible (with respect to the original problem) ray along which f_γ is decreasing.

We decrease γ in these cases because we know that eventually, for a positive γ value small enough, a first-order minimizer of f_γ is, generally, a Kuhn-Tucker point of the original constrained problem.

There is a possibility, however, that a point x^* is a first-order minimizer of f_γ for all $\gamma > 0$ small enough, while x^* is not feasible for the original problem for any $\gamma > 0$, small enough, even though feasible points do exist. This can occur in the presence of non-convexity, as x^* is a *local* optimum of f_γ. Consider, for example, the problem

$$\begin{array}{rrcl} \min & x & & \\ \text{subject to} & x^3 - x & \leq & 0, \\ & x & \geq & 0. \end{array}$$

The corresponding penalty function is

$$f_\gamma(x) = \gamma x - \min(0, -x^3 + x) - \min(0, x),$$

for which one can show that $x^* = -1$ is a local minimizer for all $\gamma \leq 1$. This local minimizer is not feasible for the original problem. Nevertheless, there do exist feasible points for the original problem ($\{x : 0 \leq x \leq 1\}$). To circumvent this disadvantage of the penalty approach requires global minimization and whenever this undesirable situation occurs in practice, usually one tries a different starting point. In what follows we shall refer to this situation as *the exceptional case*.

Advantages of the penalty approach are that it does not require a feasible point to begin and *best* solutions can be determined when no feasible point exists (best in the sense of: the l_1 norm of the infeasibilities is minimized). In the nonlinear case, *exact* penalty functions have moreover the feature of not introducing ill-conditioned (projected) Hessian matrices, since the penalty parameter does not need to be made arbitrarily small to achieve convergence to a solution of the original problem.

5.2 The Penalty Algorithm

In order to solve the constrained problem (1), we consider the minimization of the (unconstrained) function

$$f_\gamma(x) \equiv \gamma \tilde{f}(x) + \sum_{i \in E} |f_i(x)| - \sum_{i \in I} \min[0, f_i(x)]$$

for a succession of decreasing positive values of the penalty parameter γ. The l_1 exact penalty function, f_γ, is clearly a piecewise linear function, whenever the functions $\tilde{f}$ and f_i's are themselves piecewise linear. For any fixed value of the penalty parameter, we can hence use the algorithm described in Section 4.3 to get a (discontinuous) piecewise linear minimization algorithm for solving the constrained problem (1):

Let $0 < \rho < 1$. We start the minimization process with some initial value, $\gamma^1 > 0$, of the penalty parameter. If, at iteration k, we are in one of situations 1 or 2 (for $\gamma = \gamma^k$, the current penalty parameter value) described in Section 5.1, then we reduce γ^k to $\gamma^{k+1} = \rho\gamma^k$, we set $x^{k+1} = x^k$; we replace k by $k+1$ and we reapply the (discontinuous) piecewise linear minimization algorithm to the new f_{γ^k}.

Remarks:

• If we obtain that the current iterate, x^k is a local minimum of f_{γ^k} and that it is feasible, then we stop: x^k is a local minimum of the original problem (1): Indeed, if there existed a feasible direction d such that, for $\alpha > 0$ small enough, $\tilde{f}(x^k + \alpha d) < \tilde{f}(x^k)$, it would imply that $f_{\gamma^k}(x^k + \alpha d) \equiv \gamma^k \tilde{f}(x^k + \alpha d) + 0 < \gamma^k \tilde{f}(x^k) + 0 \equiv f_{\gamma^k}(x^k)$, which contradicts the fact that x^k is a local optimum for f_{γ^k}.

• Before decreasing the value of the penalty parameter, it is wise to try if setting $\gamma = 0$ would yield again the same iterate. If the answer is no, we can decrease the value of the penalty parameter. However, if the answer is yes, we then stop: problem (1) is infeasible or, in the non-convex case, x^k is possibly an infeasible local minimum of the penalty function f_γ for all $0 < \gamma \leq \gamma_k$ (the exceptional case discussed in Section 5.1). This avoids a long sequence of iterations that would end with the discovery, when the penalty parameter is negligibly small, that the original problem is infeasible or that we have an infeasible local minimum of the penalty function f_γ for all $0 < \gamma \leq \gamma^k$—the penalty function being not necessarily convex.

• If we determine unboundedness of the penalty function, f_{γ^k}, but we are *not* in the situation 2 described in Section 5.1 (i.e. we have a *feasible* ray along which the penalty function decreases), then problem (1) is unbounded and we stop. We deduce the unboundness of the *original* problem from the fact that $f_{\gamma^k}(x^k + \alpha d^k)$ tends towards $-\infty$ as α tends toward ∞, where d^k is the descent direction that yielded unboundedness of the penalty function, f_{γ^k}. Indeed, let us write the penalty function as

$$f_\gamma(x) = \gamma \tilde{f}(x) + \hat{f}(x),$$

where $\tilde{f}$ is the original objective function and $\hat{f}$ is the penalty part of f_γ (corresponding to the constraints). For all $\alpha > \bar{\alpha}$ we have, for a certain $\bar{\alpha} > 0$:

$$\begin{aligned} f_{\gamma^k}(x^k + \alpha d^k) &= \gamma^k \tilde{f}(x^k + \alpha d^k) + \hat{f}(x^k + \alpha d^k) \\ &= \gamma^k \tilde{f}(x^k + \alpha d^k), \end{aligned}$$

as $x^k + \alpha d^k$ is feasible for all $\alpha > \bar{\alpha}$. Hence, $\lim_{\alpha \to \infty} \tilde{f}(x^k + \alpha d^k) = -\infty$.

• There are two ways to reach the step in which we decrease the value of the penalty parameter. In the case where x^k is optimal for the penalty function but is not feasible (situation 1), it seems natural to set $x^{k+1} = x^k$, as suggested above. On the other hand, in the case where we obtain an infeasible ray along which the penalty function is unbounded (situation 2), it is not clear whether this choice is the best one. It is reasonable if $x^k + \alpha d^k$ is infeasible for all $\alpha > 0$ (where d^k is the descent direction) but if a part of the ray intersects the feasible region, one could consider taking the last feasible breakpoint encountered on the ray (if any) as the next iterate (x^{k+1}). However it is conceivable that x^k, though infeasible, was much closer to a local optimum than the last feasible breakpoint, not to mention that since the penalty parameter is too big, d^k is not a direction we should rely on. It seems wiser setting x^{k+1} to be the *first* feasible breakpoint encountered, if any—otherwise retain x^k. However, checking the feasibility of many breakpoints along the ray might involve too much work for the little gained. We shall thus accept the simplest update, namely $x^{k+1} = x^k$.

To make the constrained algorithm more concrete, consider the following simple example.

$$\min_{x \in I\!R^3} \tilde{f}(x) \equiv 3x_1 + x_2 + k(x), \tag{31}$$

$$\begin{aligned} \text{subject to } x_1 + x_3 &\geq 2, \\ 3x_1 - x_3 &\geq 0, \\ x_2 &\geq -6, \\ x_3 &\geq -1, \end{aligned}$$

where

$$k(x) = \begin{cases} x_2 & \text{if } x_1 + x_2 \geq 0, \\ -x_1 & \text{otherwise.} \end{cases}$$

It yields the unconstrained optimization problem: $\min_{x \in I\!R^3} f_\gamma(x)$, where

$$f_\gamma(x) \equiv \gamma[3x_1 + x_2 + k(x)] - \sum_{i=2}^{5} \min(0, a_i^T x - b_i), \tag{32}$$

$$a_1 = (1,1,0)^T, \quad a_2 = (1,0,1)^T, \quad a_3 = (3,0,-1)^T, \quad a_4 = (0,1,0)^T, \quad a_5 = (0,0,1)^T,$$

$b_1 = 0$, $b_2 = 2$, $b_3 = 0$, $b_4 = -6$, $b_5 = -1$ and $\gamma > 0$. We take $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ as the set of ridges of f_γ, where $\mathcal{R} \equiv \{1, 2, \ldots, 5\}$. At any point $\hat{x} \in I\!R^3$ we have the following decomposition of f_γ: $g_{\hat{x}}$, $\{\nu^i\}_{i \in \mathcal{A}(\hat{x})}$, where

$$g_{\hat{x}} \equiv \gamma(3,2,0)^T + \sum_{i \in \mathcal{R}} \nu^i a_i s(a_i^T \hat{x} - b_i)$$

(the function $s(x)$ is defined to take the value 1 if x is negative and 0 otherwise), $\nu^1 = -\gamma$ and $\nu^i = -1$, $i = 2,3,4,5$.

(In other words, in the neighbourhood of any point $\hat{x} \in I\!R^3$ we can write:

$$f_\gamma(x) = f(\hat{x}) + g_{\hat{x}}^T (x - \hat{x}) + \psi_{\hat{x}}(x),$$

where

$$\psi_{\hat{x}}(x) = \sum_{i \in \mathcal{A}(\hat{x})} \nu^i \min(0, a_i^T (x - \hat{x}))$$

is the nonsmooth part of f_γ.)

We take $\gamma^1 = 2$ as the initial value for the penalty parameter and $x^1 = (4,-6,-2)^T$ as our initial point. The initial objective and penalty function values are respectively $\tilde{f}(x^1) = 2$ and $f_2(x^1) = 5$. Ridges 2 and 4 are active at x^1. Since $g_{x^1} \equiv (4,2,-1)^T$ cannot be written as a linear combination of a_2 and a_4 (the projection of the current restricted gradient onto the null space of the activities, $P(g_{x^1})$, is non-zero), x^1 is not a dead point and hence, we maintain the activities by choosing our search direction in the space orthogonal to both a_2 and a_4. Thus, we take our search direction to be $d^1 = -P(g_{x^1}) = 5/2(-1,0,1)^T$. We have $f'_{\gamma^1}(x^1; d^1) = -\|d^1\|^2 = -25/2$, from equation (21).

We now descend as far as possible in that direction. After a step size of 2/5, we hit ridge 5 at the breakpoint $(3,-6,-1)^T$. We obtain this breakpoint by computing ratios analogous to those in linear programming: the half-line $\{x : x = x^k + \alpha d^k, \alpha > 0\}$ intersects a ridge i inactive at x^k when

$$\alpha = \frac{-a_i^T x^k + b_i}{a_i^T d^k} > 0.$$

Thus, the first breakpoint reached corresponds to the smallest positive ratio of the form

$$\frac{-a_i^T x^k + b_i}{a_i^T d^k}, \; i \in \mathcal{R} \setminus \mathcal{A}(x^k).$$

But we can descend further in the same direction since [using (23)]

$$\begin{aligned} f'_{\gamma^1}((3,-6,-1); d^1) &= f'_{\gamma^1}(x^1; d^1) - \nu^5 |a_5^T d^1| \\ &= -25/2 - (-1)|5/2| = -10, \end{aligned}$$

as $\nu^5 = -1$. Hence, we can go beyond this breakpoint as $f'_{\gamma^1}((3,-6,-1); d^1) < 0$. However after the following breakpoint, $(1/2,-6,3/2)^T$, corresponding to a step size of 7/5 and ridge 3 becoming active, we cannot descend any more along d^1 since

$$\begin{aligned} f'_{\gamma^1}((1/2,-6,3/2); d^1) &= f'_{\gamma^1}((3,-6,-1); d^1) - \nu^3 |a_3^T d^1| \\ &= -10 - (-1)|-10| = 0. \end{aligned}$$

Thus, we set $x^2 \equiv (1/2,-6,3/2)^T$ and $\gamma^2 = \gamma^1 = 2$. We have $\tilde{f}(x^2) = -5$ and $f_{\gamma^2}(x^2) = -10$. Three ridges are active at x^2: 2, 3 and 4 and it is a dead point: $g_{x^2} \equiv (4,2,0)^T = a_2 + a_3 + 2a_4$. Here we have $u_4 \equiv 2 \not\leq 1 \equiv -\nu^4$, (violated optimality condition). Thus, we drop activity 4 negatively by taking $d^2 = -P_{-4}(a_4) = (0,-1,0)^T$. But going in the direction d^2 we do not hit any breakpoint and we conclude that the penalty function is unbounded. However, $x^2 + \alpha d^2$ is infeasible for all $\alpha > 0$ (the constraint $x_2 \geq -6$ being violated) and hence, we reduce the penalty parameter, to $\gamma^3 = 1$ (i.e. we take $\rho = 1/2$) for example, in order to put more weight on satisfying the constraints, and set $x^3 = x^2$.

The restricted gradient is now $g_{x^3} \equiv (2,1,0)^T = 1/2a_2 + 1/2a_3 + a_4$, and we have $0 \leq u_i \leq -\nu^i$ for all $i \in \mathcal{A}(x^3) \equiv \{2,3,4\}$, as $u_2 = u_3 = 1/2$, $u_4 = 1$ and $\nu^i = -1$, $i = 2,3,4$. Thus, the optimality conditions are satisfied at x^3. Therefore, according to the optimality condition theorem (Theorem 5), the point x^3 is a local minimum of the penalty function f_{γ^3} defined by equation (32). Since moreover x^3 is feasible, it follows that it is a local optimum for problem (31). At optimality we have $\tilde{f}(x^3) = -5$ and $f_{\gamma^3}(x^3) = -5$.

We can restate the finite step convergence result of Section 3.4 in the constrained case:

Theorem 6 (Convergence) *Let $f_\gamma : I\!R^n \to I\!R$ be the exact l_1 penalty function corresponding to problem (1) with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set. Assume that*

1. *$\{a_i\}_{i \in \mathcal{A}(x^k)}$ are linearly independent at each iterate, x^k, encountered in the course of the algorithm,*
2. *f_γ is decomposable at each breakpoint encountered along the line searches,*

3. (in the discontinuous case) no iterate encountered in the course of the algorithm is a contact point of f_γ, and

4. the exceptional case described in Section 5.1 does not occur (that is to say, no infeasible iterate x^k encountered in the course of the algorithm is a local optimum of f_γ for all $0 < \gamma \leq \gamma^k$).

Then the discontinuous piecewise linear minimization algorithm for the constrained case converges globally in a finite number of iterations.

Proof: Finiteness of the algorithm follows from the finiteness of the unconstrained minimization algorithm described in Section 4.3 and from the exactness of the penalty function. Indeed, we need to consider only a finite number of values of γ before reaching a penalty parameter value γ' such that a local unconstrained minimizer of $f_{\gamma'}$ corresponds to a local minimizer of the original (constrained) problem. The algorithm will not terminate until an optimal dead point has been found or a conclusion of unboundedness or infeasibility is reached (as we discarded the eventuality that the algorithm ends with an iterate x^k which is found to be an infeasible local minimum of f_γ for all $0 < \gamma \leq \gamma^k$). □

5.3 Improvement

The following proposition is a generalization of a result in linear programming [6] (that was first proved in the case of linear networks, see [36]) which will yield improvement to our algorithm. Its proof follows straightforwardly from the earlier discussion. For the sake of simplicity we present it in the continuous case.

Proposition 3 *Let $f_\gamma : I\!\!R^n \to I\!\!R$ be the penalty function corresponding to problem (1) with ridges $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$, where $\mathcal{R}$ is a finite index set. Assume that f_γ is continuous. Let $y \in I\!\!R^n$ be such that $\{a_i\}_{i \in \mathcal{A}(y)}$ are linearly independent, and let g_y, $\{\nu_y^i\}_{i \in \mathcal{A}(y)}$ be the decomposition of f_γ at y.*

Suppose that there exist scalars $\{u_i\}_{i \in \mathcal{A}(y)}$ such that

$$g_y = \sum_{i \in \mathcal{A}(y)} u_i a_i, \tag{33}$$

$$u_i \geq 0 \text{ for all } i \in \mathcal{A}(y) \text{ and } u_i \leq -\nu_y^i \text{ for all } i \in \mathcal{A}(y) \setminus I_{ineq}, \tag{34}$$

where I_{ineq} is an index set corresponding to the ridges of the penalty function that correspond to inequality constraints of the form $a_i^T x - b_i \geq 0$, for some $i \in \mathcal{R}$.

Then the only possible descent directions are violating some constraint that was not violated at y.

Proof: Consider $d \in I\!\!R^n$, an arbitrary direction. We have, for $\alpha > 0$ small enough:

$$\begin{aligned} f_\gamma(y + \alpha d) &= f_\gamma(y) + \alpha d^T [g_y + \sum_{i \in \mathcal{A}(y): a_i^T d < 0} \nu_y^i a_i] \\ &= f_\gamma(y) + \alpha \sum_{i \in \mathcal{A}(y)} (u_i + \lambda_y^{i,d}) a_i^T d, \end{aligned} \tag{35}$$

where

$$\lambda_y^{i,d} = \begin{cases} \nu_y^i & \text{if } a_i^T d < 0, \\ 0 & \text{otherwise,} \end{cases}$$

using (33). Consequently,

$$\begin{aligned} f_\gamma(y + \alpha d) = f_\gamma(y) \quad &+ \quad \alpha \sum_{i \in \mathcal{A}(y): a_i^T d < 0} -(u_i + \nu_y^i)|a_i^T d| \\ &+ \quad \alpha \sum_{i \in \mathcal{A}(y): a_i^T d \geq 0} u_i |a_i^T d|. \end{aligned}$$

Whence, using (34),

$$f_\gamma(y + \alpha d) \geq f_\gamma(y) + \alpha \sum_{i \in \mathcal{A}(y) \cap I_{ineq}: a_i^T d < 0} -(u_i + \nu_y^i)|a_i^T d|,$$

since the omitted terms are non-negative. Thus, the only way we can have $f_\gamma(y + \alpha d) < f_\gamma(y)$ is for d to be such that $a_k^T d < 0$, for some $k \in \mathcal{A}(y) \cap I_{ineq}$. That is to say, to have a direction d violating the constraint $a_k^T x - b_k \geq 0$, for some k. □

Thus, the fact that the current point y satisfies (33) and (34) indicates that, with the possible exception of the primal feasibility condition not being satisfied, one is at a local minimum of f_γ. It follows that, ignoring the exceptional case, either one is optimal (when feasibility is satisfied) or the penalty parameter γ is above the threshold value (if feasibility is not satisfied). Consequently there seems little to be gained by not reducing the penalty parameter immediately.

With a similar proof for the case of equality constraints, which can be written as pairs of inequality constraints, we can restate the proposition, replacing (34) by:

$$u_i \geq 0 \text{ for all } i \in \mathcal{A}(y) \setminus I_{equ} \text{ and } u_i \leq -\nu_y^i \text{ for all } i \in \mathcal{A}(y) \setminus (I_{ineq} \cup I_{equ}), \tag{36}$$

where I_{equ} is the index set of the ridges of the penalty function that correspond to equality constraints of the form $a_i^T x - b_i = 0$, for some $i \in \mathcal{R}$. When a point $y \in I\!R^n$ satisfies (33) and (36) it is said to be *dual feasible*. The terminology is a natural extension from linear programming since $u_i, i \in I_{equ} \cup I_{ineq}$ and $u_i + \nu^i, i \in I_{equ}$ are the analogues of dual variables: u_i and $u_i + \nu^i$ are unrestricted in sign for $i \in I_{equ}$, while u_i is restricted to be non-negative if $i \in I_{ineq}$ ($u_i + \nu^i$ is unrestricted in sign for $i \in I_{ineq}$ since whatever is the sign of $u_i + \nu^i$, dropping negatively a single ridge i corresponding to the constraint $a_i^T x - b_i \geq 0$ cannot yield descent in f_γ). Thus, this yields a modification to the algorithm that can be summarized as follows: Unless we are in the situation where conditions (36) are satisfied for f_{γ^k} as well as for f_0 (i.e. with $\gamma = 0$)—in which case there is no point on trying to decrease γ—, the only dropping directions one considers are the ones corresponding to conditions (36), i.e. the ones which do not yield new violations of linear constraints.

If, in the example (31) above, an algorithm modified accordingly were used, we would not consider verifying whether the optimality condition $u_4 + \nu^4 \leq 0$ is satisfied, since $4 \in I_{ineq}$ (dropping activity 4 negatively implies violation of constraint $x_2 + 6 \geq 0$). We would find out that none of the inequalities in (36) is violated at x^2, and, since moreover x^2 is feasible, we would conclude right away that it is a local optimum for the problem.

We now present an algorithm for solving the *constrained* problem (1) and which incorporates the improvement introduced in this section.

Let f_γ be the exact l_1 penalty function corresponding to problem (1), and $\{a_i^T x - b_i\}_{i \in \mathcal{R}}$ be the ridges of f_γ, where $\mathcal{R}$ is a finite index set. We assume that f_γ is decomposable at each iterate and at each breakpoint encountered in the line search. Moreover, we assume that at every iterate which is a dead point, the gradients of the activities are linearly independent and that all points encountered in the algorithm are non-contact points. The degenerate situation is discussed in the next section. Let I_{ineq}^+ and I_{ineq}^- be the index sets corresponding to the ridges of the penalty function that correspond to inequality constraints of the form $a_i^T x - b_i \geq 0$ and $a_i^T x - b_i \leq 0$ respectively, for some $i \in \mathcal{R}$.

Let g_{x^k}, $\{\nu_{x^k}^i\}_{i \in \mathcal{A}(x^k) \setminus \mathcal{F}(x^k)}$ be the decomposition of f_γ at x^k. We assume, without loss of generality, that $\mathcal{F}(x^k) = \mathcal{F}^-(x^k)$ (that is to say, all faults yield a "jump up" when dropped negatively).

Discontinuous Piecewise Linear Minimization Algorithm for the Constrained Case

Step 1: Choose any $x^1 \in I\!R^n$, $\gamma^1 > 0$ and $0 < \rho < 1$. Set $k \leftarrow 1$ and tryv $\leftarrow$ false.

Repeat (until stop):
Begin
  Repeat (until reduce γ^k or stop):
  Begin
  Step 2: Identify the activities, $\mathcal{A}(x^k)$, and compute $d^k \equiv -P(g_{x^k})$, the projection of the restricted gradient onto the space orthogonal to the gradients of the activities.

  If $d^k = \vec{0}$ then (x^k is a dead point. Compute a non-soaring single-dropping descent direction or establish optimality)

    Step 3: Compute $\{u_i\}_{i \in \mathcal{A}(x^k)}$, the coefficients of $\{a_i\}_{i \in \mathcal{A}(x^k)}$ in the linear combination of g_{x^k} in terms of the columns of $A(x^k)$, and identify the faults at x^k, $\mathcal{F}(x^k)$.

    If not (tryv) then,
      If $u_i < 0$, $i \notin I_{equ} \cup I_{ineq}^-$, or $u_i > -\nu_{x^k}^i$, $i \notin I_{equ} \cup I_{ineq}^+ \cup \mathcal{F}(x^k)$,
      for some $i \in \mathcal{A}(x^k)$ (violated optimality condition), then
        Drop i: go to step 8.
      Else
        Step 4: (Dual feasibility) If x^k is feasible then x^k is a local minimum of problem (1) and stop.
        Step 5: If trying with $\gamma = 0$ would *not* yield again the same iterate, then reduce γ^k: go to step 12.
        Set tryv $\leftarrow$ true.
      Endif
    Endif

Step 6: (Consider directions which would violate a satisfied constraint)
If, for each $i \in \mathcal{A}(x^k)$: $u_i \geq 0$, $i \in I_{equ} \cup I_{ineq}^-$, and
$u_i \leq -\nu_{x^k}^i$, $i \in (I_{equ} \cup I_{ineq}^+) \setminus \mathcal{F}(x^k)$, then
If trying with $\gamma = 0$ would yield again the same iterate, then
Step 7: Stop: problem (1) is infeasible or, in the non-convex case,
x^k is possibly an infeasible local minimum of the penalty function f_γ
for all $0 < \gamma \leq \gamma_k$ (the exceptional case).
Else reduce γ^k: go to step 12.
Endif
Endif

Step 8: (Drop an activity) Redefine $d^k = P_{-i}(a_i)$, if the violated
inequality found corresponds to $u_i \geq 0$, otherwise $d^k = -P_{-i}(a_i)$ and
$u_i \leq -\nu_{x^k}^i$, where P_{-i} is the orthogonal projector onto the space
orthogonal to the gradients of all the activities but activity i.
Endif

Step 9: (Line search) Determine the step size α^k by solving
$\min_{\alpha > 0} \; f_{\gamma^k}(x^k + \alpha d^k)$. This line search can be done from x^k, moving from
one breakpoint of f_{γ^k} to the next, in the direction d^k, until either we establish
unboundedness of the penalty function or the value of f_{γ^k} starts increasing.

If unbounded then
Step 10: If $x^k + \alpha d^k$ is feasible for all $\alpha > \bar{\alpha}$ for a certain $\bar{\alpha} > 0$ (i.e. we
have a feasible ray along which the penalty function decreases), then
problem (1) is unbounded and stop.
Else reduce γ^k: go to step 12.
Endif
Else
Step 11: Update $x^{k+1} = x^k + \alpha^k d^k$, $k \leftarrow k + 1$, tryv $\leftarrow$ false.
Endif
End

Step 12: Reduce γ^k to $\gamma^{k+1} = \rho \gamma^k$, $x^{k+1} = x^k$, $k \leftarrow k + 1$.
End

6 Degeneracy

Theorem 1 provides a sufficient condition (the linear independence of the activities) for a continuous piecewise linear function to be decomposable at a point $\hat{x}$. It would be interesting to have a condition characterizing "decomposability".

Define, for any $J \subseteq \mathcal{A}(\hat{x})$, the set

$$S^J \equiv \{x \in I\!R^n : a_i^T(x - \hat{x}) > 0, \; i \in J \text{ and } a_i^T(x - \hat{x}) < 0, \; i \in \mathcal{A}(\hat{x}) \setminus J\}.$$

Such a set S^J, if non-empty, is in fact a cell of f whose closure contains $\hat{x}$. In the case where $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$ are linearly dependent, we do not necessarily have $S^J \neq \emptyset$ for each $J \subseteq \mathcal{A}(\hat{x})$. Assume that f has the form (12) in a neighbourhood of $\hat{x}$, for some $c^J \in I\!R^n$ given for each $J \subseteq \mathcal{A}(\hat{x})$ such that $S^J \neq \emptyset$. Then, we can show that f is decomposable at $\hat{x}$ if and only if there exist scalars ν^i, $i \in \mathcal{A}(\hat{x})$, such that

$$\nu^i a_i = c^{J-i} - c^J, \quad \text{for all } i \in J \subseteq \mathcal{A}(\hat{x}) \tag{37}$$

such that $S^J \neq \emptyset$ and $S^{J-i} \neq \emptyset$.

This necessary and sufficient condition for f to be decomposable at a point $\hat{x}$ is however not very practical, as it involves the verification of equation (37) for *every* possible subset J of $\mathcal{A}(\hat{x})$ (such that S^J and S^{J-i} are non-empty) containing i, and this for every $i \in \mathcal{A}(\hat{x})$. We shall omit the proof.

Note however that condition (37) is satisfied at any point $\hat{x} \in I\!R^n$ for the decomposable function given by (5) (see Figure 3). Also, it is clear from the form of the function f_γ given by (8) that it is decomposable at any point, even when the gradient of the activities are linearly dependent. Condition (37) is straightforwardly verified at any $\hat{x} \in I\!R^n$ in this latter example. A decomposition $g_{\hat{x}}, \{\nu^i\}_{i\in\mathcal{A}(\hat{x})}$ of this function at any point $\hat{x} \in I\!R^n$ can be obtained automatically in a manner similar to the linearly independent case. If we set

$$g_{\hat{x}} \equiv \gamma c + \sum_{i\in\mathcal{A}(\hat{x})\cap E} a_i + \sum_{i\in E\setminus\mathcal{A}(\hat{x})} a_i \mathrm{sign}(a_i^T x - b_i) - \sum_{i\in I\setminus\mathcal{A}(\hat{x})} a_i s(a_i^T x - b_i), \tag{38}$$

and, for each $i \in \mathcal{A}(\hat{x})$, let J be any subset of $\mathcal{A}(\hat{x})$ such that $i \in J$, $S^J \neq \emptyset$ and $S^{J-i} \neq \emptyset$. Set, for each $i \in \mathcal{A}(\hat{x})$, ν^i to be such that

$$\nu^i a_i = c^{J-i} - c^J.$$

We thus obtain

$$\nu^i = \begin{cases} -1 & \text{if } i \in I, \\ -2 & \text{if } i \in E. \end{cases} \tag{39}$$

Hence, for all $\hat{x} \in I\!R^n$, we have:

$$f_\gamma(x) = f_\gamma(\hat{x}) + g_{\hat{x}}^T(x - \hat{x}) + \sum_{i\in\mathcal{A}(\hat{x})} \nu^i \min(0, a_i^T(x - \hat{x})),$$

for all x in some neighbourhood of $\hat{x}$ small enough, where $g_{\hat{x}}$ is defined by (38) and the scalars $\{\nu_{\hat{x}}^i\}_{i\in\mathcal{A}(\hat{x})}$ are given by (39). Note that this is an example of a function with a decomposition $g_{\hat{x}}, \{\nu^i\}_{i\in\mathcal{A}(\hat{x})}$ such that the scalars ν^i's are independent of $\hat{x}$. (We can show that this is a consequence of the fact that this function is decomposable at each point. The proof would be analogous to the proof of the fact, mentioned at the end of Section 3.1, that the ν^i's are not dependent upon $\hat{x}$ in the case where $\{a_i\}_{i\in\mathcal{A}(\hat{x})}$ are linearly independent at each point of the domain of a function.)

When at a certain point $x_0 \in I\!R^n$ the columns of matrix $A(x_0)$ are linearly dependent, we call the point x_0 a *degenerate* point. If a solution (a local minimum) of problem (1) is degenerate, we then say that problem (1) is a *degenerate* problem. If, in the course of the algorithm presented, we encounter a degenerate point, then certain parts of our previous discussion break down. In this section we also discuss how the algorithm is modified when it encounters a point at which f is not decomposable.

From the decomposition theorem we know that if $\hat{x}$ is non-degenerate, then f is decomposable at $\hat{x}$. However, as we already mentioned, there are instances of piecewise differentiable functions which are decomposable at degenerate points. On the other hand, there are piecewise linear functions which are not decomposable at degenerate points. An example is given in Section 6.2.

Let us now see how degeneracy affects our algorithm. If the current iterate, x_0, is degenerate but is not a dead point, nothing is changed: from a restricted gradient, g_{x_0}, of f at x_0, we can obtain a descent direction, $d = -P(g_{x_0})$, preserving the current set of activities, as before. (In practice, in order to compute the projection of the restricted gradient onto the space orthogonal to the gradients of the activities, we use a subset of the activities whose gradients constitute a basis of the range space of A. We remark that this is not necessarily easy computationally.)

6.1 The Decomposable Case

Suppose that f is decomposable at a degenerate point x_0 and let $g_{x_0}, \{\nu_{x_0}^i\}_{i\in\mathcal{A}(x_0)}$ be the decomposition of f at x_0.

If x_0 is a dead point then:

- a direction d dropping only one activity may not exist (if a_k is linearly dependent upon the other columns of A, there does not exist a $d \in \mathcal{N}(A_{-k}^T)$ such that $a_k^T d \neq 0$);
- the $\{u_i\}_{i\in\mathcal{A}(x_0)}$ are no longer unique (there is not a unique representation of a given vector in terms of a set of linearly dependent vectors).

This suggests defining a matrix B (non-uniquely) by selecting a basis for the columns of A. Let $\mathcal{B}$ be the index set of the columns of B. As in equation (35), we have that, for $d \in I\!R^n$ and $\alpha > 0$ small enough (recall that x_0 is a dead point):

$$f(x_0 + \alpha d) = f(x_0) + \alpha \sum_{i\in\mathcal{A}} (u_i + \lambda_{x_0}^{i,d}) a_i^T d, \tag{40}$$

where

$$\lambda_{x_0}^{i,d} = \begin{cases} \nu_{x_0}^i & \text{if } a_i^T d < 0, \\ 0 & \text{otherwise} \end{cases}$$

(we assume the continuous case for the sake of simplicity). Consider dropping activity $k \in \mathcal{B}$, while keeping the remaining columns of B active. We obtain, for $\alpha > 0$ small enough,

$$\begin{aligned} f(x_0 + \alpha d) &= f(x_0) + \alpha(u_k + \lambda_{x_0}^{k,d}) a_k^T d + \alpha \sum_{i\in\mathcal{A}\setminus\mathcal{B}} (u_i + \lambda_{x_0}^{i,d}) a_i^T d \\ &= f(x_0) + \alpha(u_k + \lambda_{x_0}^{k,d}) a_k^T d + \alpha \sum_{i\in\mathcal{A}\setminus\mathcal{B}} \lambda_{x_0}^{i,d} a_i^T d, \end{aligned} \tag{41}$$

writing g_{x_0} as a linear combination in terms of the $\{a_i\}_{i\in\mathcal{B}}$ (equivalently, we set $u_i = 0$ for each $i \in \mathcal{A} \setminus \mathcal{B}$). Thus, the second condition of the optimality condition theorem (Theorem 5), namely:

$$0 \le u_i \le -\nu^i, \quad \text{for all } i \in \mathcal{A},$$

is no longer necessary for optimality. Indeed, the fact that $u_k < 0$ or $u_k > -\nu^k_{x_0}$ does not imply that d is a descent direction. Before stating that a single-dropping direction is descending we have to verify the effect of the dependent terms (the summation terms in equation (41)).

The conditions of the optimality condition theorem, when verified for each $i \in \mathcal{A}$, are sufficient for optimality. Indeed, we can write g_{x_0} uniquely in terms of the columns of B, and, for an arbitrary direction, d, we have (40) with $u_i = 0$ if $i \in \mathcal{A} \setminus \mathcal{B}$. Clearly, $u_i \ge 0$ and $u_i \le -\nu^i_{x_0}$ for all $i \in \mathcal{A}$ then implies $f(x_0 + \alpha d) \ge f(x_0)$.

Nevertheless, if the conditions of the optimality condition theorem are *not* verified, we do not know whether we are optimal. In such a case, we are left with the problem of exhausting the various combinations of columns of A that form a basis of $\mathcal{N}(A^T)$ in order to establish the optimality of the current point. (More precisely we would have to look at the effect [equation (41)] of each direction corresponding to dropping one activity from a basis of the activities and this, for each possible basis $\mathcal{B}$. In fact there is a one-to-one correspondence between each such dropping directions and each subset $\mathcal{S} \subseteq \mathcal{A}$ of linearly independent activities such that $\mathcal{S}$ contains one element fewer than a basis of the range space of A. Finding a good way to determine the set of all these dropping directions in order to know if a descent direction exists, is a combinatorial optimization problem.)

Typically in practice, degeneracy has been resolved by perturbation techniques. Only as many columns of A are added to matrix B as will be linearly independent. Extra columns are treated as having been subjected to infinitesimal perturbations. We shall in fact perturb the *right-hand sides* of some columns of A (i.e. the constant terms, b_i, of the active ridges $a_i^T x - b_i$, $i \in \mathcal{A}$) until the set of columns which are considered active are linearly independent.

Note that once a perturbation is performed, f refers to the current (perturbed) function and not the original function until either optimality (for which optimality for the corresponding unperturbed problem is immediate) is identified or we leave the neighbourhood defined by the perturbation—this neighbourhood can be seen here as the convex hull of the points (of the perturbed problem) whose set of activities corresponds to the degenerate dead point x_0 (of the original problem). Thus once we obtain, in the perturbed problem, an iterate x_k which is out of this neighbourhood and a corresponding dropping direction, we know which activity to drop and we come back to the degenerate dead point x_0 *with the original data*. The dropping direction that brought us out of the neighbourhood defined by the perturbation is the descent direction we should use in the original problem. There is no need to recompute it when we revert to the original problem, since the computing of the dropping direction is not affected by changes to the constant term of the ridges (b_i).

When $g_{x_0}, \{\nu^i_{x_0}\}_{i\in\mathcal{A}(x_0)}$ is a decomposition of the original function at x_0, the perturbed function, $\bar{f}$, is defined as follows:

$$\bar{f}(x) = f(x_0) + g_{x_0}^T(x - x_0) + \sum_{i \in \mathcal{A}(x_0)} \nu_{x_0}^i \min(0, a_i^T x - b_i'),$$

where b_i' is the right-hand side b_i subject to a perturbation (for $i \in \mathcal{A}(x_0) \setminus \mathcal{B}$, where $\mathcal{B}$ is a basis for $\{a_i\}_{i \in \mathcal{A}(x_0)}$).

However, using perturbations in a general non-convex case can cause a problem. Consider the following function defined over $I\!R^2$:

$$f(x) = -x_1 - x_2/2 + \min(0, x_1 + x_2) - \min(0, x_1) - \min(0, x_2).$$

The gradients of the activities are linearly dependent at $x_0 \equiv (0,0)^T$. We consider the perturbed function

$$\bar{f}(x) = -x_1 - x_2/2 + \min(0, x_1 + x_2 - \epsilon) - \min(0, x_1) - \min(0, x_2),$$

for some $\epsilon > 0$. The arrows in Figures 7 and 8 show the direction of negative gradient on each cell of the original problem and the perturbed subproblem respectively. It is clear that x_0 is not a local optimum, since $d \equiv (0,1)^T$ is clearly a descent direction for f. However, since $\bar{x}_0 \equiv (0,0)^T$ is a local minimum of the perturbed function, if the algorithm is applied to the perturbed subproblem starting at $\bar{x}_0$, it would not find a descent direction leaving the neighbourhood defined by the perturbation. It would terminate with $\bar{x}_0$ as local optimum of the subproblem. The descent direction, d, is not found in the perturbed problem because of a non-convexity feature introduced along the line $\hat{D} = \{x : x = \bar{x}_0 + \alpha d, \alpha \in I\!R\}$ by the perturbation (the objective function restricted to the line $\hat{D}$ is non-convex, whereas it was convex in the original problem when restricted to the line $D = \{x : x = x_0 + \alpha d, \alpha \in I\!R\}$—though it was not convex in the neighbourhood of x_0). *Thus, non-convexity in the neighbourhood of a degenerate point x_0 may yield the conclusion that x_0 is a local optimum while there exists a descent direction.*

Nevertheless, this optimization problem could be seen as an incarnation of a non-degenerate problem. Thus the solution found by the algorithm would be the local optimum of some problem having exactly the same representation inside the machine as the original problem but that is now non-degenerate. Thus although the conclusion reached may not be ideal, it has to be considered as perfectly reasonable.

Let us now look into Osborne's [51] interpretation of the perturbation technique that Wolfe [61] introduced for resolving degeneracy in linear programming. Osborne first *localizes* the problem by ignoring inactive ridges. We consider, in the perturbed subproblem, no ridges other than the ones active at x_0 in the original problem, since the aim is to find out whether or not any descent is possible from the current degenerate dead point. This has the immediate advantage of avoiding computations of ratios (for the line search) corresponding to inactive ridges, in the case where we obtain a descent direction with respect to the subproblem. We then continue to apply the steps of the algorithm to the points of the perturbed problem until one of the following two situations occurs.

(i) We derive optimality of a dead point of the perturbed problem, and hence, the optimality of the degenerate point x_0 in the original problem (subject to the above observation about the fact that, in the non-convex case, at least x_0 is a local optimum of some problem having exactly the same representation in the machine as the original problem).

(ii) An unbounded direction of descent has been constructed in the perturbed problem. We now revert to the original problem, and continue with this descent direction, allowing us to leave the degenerate dead point x_0.

Osborne considers a third possible situation, that of encountering a degenerate dead point in the perturbed problem. He then recursively invokes the same procedure but with strictly fewer ridges to consider, yielding the finiteness of the depth to which the recursive procedure can be invoked.

If one uses random perturbations, this third situation is unlikely to happen. Ryan and Osborne's application of this technique to highly degenerate linear programs arising from the set partitioning model of scheduling problems [58], seems to corroborate this: the above perturbation technique proved its effectiveness on these problems and no further degeneracy was detected in the perturbed subproblems.

We implemented successfully Osborne's interpretation of Wolfe's perturbation technique. This technique has the advantage of avoiding the computation of ratios associated with inactive ridges in the perturbed subproblem, and handling highly degenerate problems efficiently. Also, since the activities in the subproblem are a subset of those at the original degenerate point, we can easily update the factorization of the matrix of activities by deleting from matrix $A(x_0)$ the linearly dependent columns corresponding to the perturbed constant terms of the ridges. In addition this approach has the elegant feature of not necessitating *small* perturbations, since the subproblem does not take into account the inactive ridges of the original problem. The magnitude of the perturbations does not affect the conclusion of optimality at some point of the subproblem, nor does it affect the descent direction (in the case where there is one). This is of particular interest when tackling the nonlinear case. Indeed, the notion of *small* perturbations can be cumbersome in the framework of a second-order algorithm.

Notice that in a perturbed subproblem (in the case of degeneracy), when evaluating restricted gradients we refer, for the ridges which were inactive at the original degenerate point x_0, to the sign of the residuals (the ith residual is the value $a_i^T x - b_i$) corresponding to the *original* problem at the degenerate point. (We said earlier that in the perturbed subproblem we do not consider those ridges which were inactive at x_0, but we might need to refer to the sign of their residuals in order to evaluate the restricted gradients, and we want the sign of the residuals to reflect the fact that we do not cross any of these ridges even though the perturbation is big.)

Finally, recall that we do not need to use a perturbation at a degenerate point, x_0, in each of the following two cases:

- the function is decomposable at x_0, and x_0 is being encountered as a breakpoint along the line search,

- x_0 is an iterate which is not a dead point.

To summarize, here is how the algorithm given at the end of Section 5 can be modified when it encounters a degenerate point x^k at which f_{γ^k} is decomposable.

In step 1, set subprob $\leftarrow$ false. In step 2, if $\{a_i\}_{i\in\mathcal{A}(x^k)}$ are linearly dependent, then a basis, $\mathcal{B}(x^k) \subset \mathcal{A}(x^k)$, for the columns of $A(x^k)$ is selected. It is a factorization of the corresponding matrix, $B(x^k)$, which is then used to compute the projection of the restricted gradient. If $d^k = \vec{0}$ and subprob = false, then: Set subprob $\leftarrow$ true, perturb the right-hand sides b_i, $i \in \mathcal{A}(x^k) \setminus \mathcal{B}(x^k)$, redefine $\mathcal{A}(x^k)$ to be simply $\mathcal{B}(x^k)$ and store $x_0 \leftarrow x^k$. In step 4, in the case where f_{γ^k} is non-convex and subprob = true, there is a possibility that although x^k is a local minimum of the perturbed subproblem, x_0 is a spurious local minimum of the original problem (as it is the case for $\bar{x}_0$ in Figure 8). A similar remark apply to step 7. If subprob = true, then the line search of step 9 is done ignoring the ridges which were inactive at x_0 in the unperturbed problem. If subprob = true, then step 10 is replaced by: "Revert to the unperturbed right-hand sides, set subprob $\leftarrow$ false, and $x^k \leftarrow x_0$. Go to step 9."

6.2 The Non-Decomposable Case and Contact Points

We now give an example of a simple continuous piecewise linear function, $f : I\!R^2 \to I\!R$, which is not decomposable at the degenerate point $x_0 = (0,0)$:

$$f(x) = \begin{cases} -x_2 & \text{if } x_2 < 0 \text{ and } x_2 > x_1, \\ -x_1 & \text{if } x_1 < 0 \text{ and } x_2 \le x_1, \\ 0 & \text{otherwise.} \end{cases} \tag{42}$$

(The graph of f in 3-space looks like a corner of a pyramid "lying on the ground" over the third orthant.) Let us take $a_i^T x - b_i$, $i = 1,2,3$, as the ridges of f, where $a_1^T = (1,0)$, $a_2^T = (0,1)$, $a_3^T = (1,-1)$ and $b_i = 0$, $i = 1,2,3$. We can decompose f as follows at any point $\hat{x} \ne x_0$:

$$f(x) = f(\hat{x}) + g_{\hat{x}}^T(x-\hat{x}) + \sum_{i\in\mathcal{A}(\hat{x})} \nu_{\hat{x}}^i \min(0, a_i^T(x-\hat{x})),$$

where

$$g_{\hat{x}} \equiv \begin{cases} (0,-1)^T & \text{if } a_2^T\hat{x} < b_2 \text{ and } a_3^T\hat{x} < b_3, \\ (-1,0)^T & \text{if } a_1^T\hat{x} < b_1 \text{ and } a_3^T\hat{x} \ge b_3, \\ (0,0)^T & \text{otherwise,} \end{cases}$$

$$\nu_{\hat{x}}^i = \begin{cases} -1 & \text{if } a_i^T\hat{x} < b_i, \\ 0 & \text{if } a_i^T\hat{x} > b_i \end{cases}, \ i = 1,2 \quad \text{and} \quad \nu_{\hat{x}}^3 = \begin{cases} 1 & \text{if } a_1^T\hat{x} < b_1, \\ 0 & \text{if } a_1^T\hat{x} > b_1 \end{cases}.$$

The bold plain lines in Figure 9 delimit the different subdomains over which f is differentiable. The circles contain the value of the restricted gradient, $g_{\hat{x}}^T$, in each region where f is decomposable. Each square contains the ν^i value for each segment of ridge.

We can easily show that there does not exist a decomposition of f at $x_0 = (0,0)$. Suppose that $g_{x_0}, \{\nu_{x_0}^i\}_{i\in\mathcal{A}(x_0)}$ were one. Then it should be such that

$$g_{x_0} + \nu_{x_0}^1 a_1 + \nu_{x_0}^2 a_2 + \nu_{x_0}^3 a_3 = (0,-1)^T, \tag{43}$$

$$g_{x_0} + \nu_{x_0}^1 a_1 + \nu_{x_0}^2 a_2 = (-1,0)^T, \tag{44}$$

$$
\begin{aligned}
g_{x_0} &= (0,0)^T & (45)\\
\text{and} \quad g_{x_0} + \nu^3_{x_0} a_3 &= (0,0)^T & (46)
\end{aligned}
$$

(see Figure 9). But (46) and (45) imply $\nu^3_{x_0} = 0$, which, together with (43) and (44), yields a contradiction. This function is also reported in [32] when Fletcher discusses nonsmooth optimization problems which are not well modelled (even locally) by *composite functions*

$$\phi(x) \equiv g(x) + h(c(x)),$$

where $g : I\!R^n \to I\!R$, $c : I\!R^n \to I\!R^m$ are smooth functions and $h : I\!R^m \to I\!R$ is convex but nonsmooth. This function (42) is also given in [62, example (2.18)] to illustrate the limitations of Womersley's optimality conditions for piecewise smooth functions. The problem is described therein in terms of the directional derivative $f'(\hat{x}; d)$ being strictly smaller than the *generalized directional derivative* [12], $f^0(\hat{x}; d)$, for some direction d (namely, here, for a direction such that $d_2 < d_1 < 0$).

Note that in practice, when at the current iterate, x^k, a ridge i is active but x^k lies on a segment of the ridge such that $\nu^i_{x^k} = 0$ (as the dashed line part of the ridges on Figure 9), we do not consider i active. Doing so we simplify calculations (there is no advantage to preserve this "activity" (there is no discontinuity in the derivative there) and we avoid a degenerate situation by considering only "essential" activities).

Let f be a function which is not decomposable at a degenerate point x_0 (or for which we do not know a decomposition at x_0, we should say).

If x_0 was encountered in the course of the algorithm as a breakpoint along the line search (in step 9), from iterate x_k, then, contrary to the case where f is decomposable at x_0, we do not have a straightforward way [as in equation (23)] to update the directional derivative of f along the search direction d^k. Consider for example the direction d on Figure 10 with the function given by (42) (see Figure 9). As we saw earlier, the only point where f is not decomposable is x_0. One way to obtain the directional derivative of f at x_0 ($\equiv x^k + \alpha_0 d$, for some step size α_0, where x^k is the iterate from which the line search originated) in the direction d is to perturb the direction and use the decomposition of f at $x^i \equiv x^k + \alpha_i d'$, $i = 1, 2, 3$, where d' is the perturbed descent direction and $\alpha_i = -(a_i^T x^k - b_i)/a_i^T d'$, $i = 1, 2, 3$ are the step sizes corresponding to breakpoint x^i, $i = 1, 2, 3$, along the direction d' starting from x^k (note that if the perturbation is small enough, x^i will not be degenerate and hence f is decomposable at x^i, $i = 1, 2, 3$).

Thus, the update of the directional derivative of f at x_0 along d is done as follows

$$f'(x_0; d) = f'(\bar{x}; d) - \sum_{i=1}^{3} \nu^i_{x^i} |a_i^T d|, \tag{47}$$

where $\bar{x}$ is the breakpoint preceding breakpoint x_0 along d. Note that in the factor $|a_i^T d|$ we use the *unperturbed* direction d, so that (if the magnitude of the perturbation is small enough) the update in equation (47) is exact. If we consider only the ridges which are active at the non-decomposable degenerate breakpoint, we can again avoid using a *small* perturbation of the search direction. In fact, roughly speaking, we consider a perturbation of the search direction for the sole purpose of knowing which ν^i values we should use for each ridge active at x_0 (the ν^i corresponding to which segment of each ridge—the dashed line versus the bold plain line part of ridge i in Figure 10). Clearly, considering rather d'' (Figure 10) as the perturbed search direction yields the same result.

We cannot however extend this algorithm to deal with dead-point *iterates* (i.e. not encountered as breakpoint along the line search) without considering carefully the combinatorial nature of the problem of degeneracy.

Similarly, in the discontinuous case, there is no straightforward extension of our approach to the cases where the algorithm encounters a contact point. In the continuous case, the behaviour of f over two juxtaposed cells are linked (see the proof of the decomposition theorem). At contact points however, there is coincidence of the values of restrictions of f to subdomains not otherwise linked to each other.

7 Nonlinear Case

Solution algorithms for particular nonlinear, nonsmooth optimization problems have been proposed by various researchers. Fletcher derives in [31] a model algorithm for composite nondifferentiable optimization problems (which encompass exact penalty functions in nonlinear programming, nonlinear min-max problems, best nonlinear l_1, l_2, and l_∞ approximation, and finding feasible points of nonlinear inequalities). Local properties of algorithm for minimizing nonsmooth composite functions are further considered by Womersley in [63]. Womersley & Fletcher presented an algorithm for *max functions* using an active-set strategy, a nonsmooth line search, and a quasi-Newton approximation to the reduced Hessian of a Lagrangian function [65]. Structure functionals have been further used by Overton & Womersley in setting up solution algorithms for minimizing functionals of eigenvalues of square matrices with respect to parameters in the matrix specification [54, 55, 56]. Some particular nonsmooth problems have been tackled by active-set methods for a long time [3, 9, 10, 67] and more recently via interior-point methods [1, 18, 16, 24].

Having given the framework of the decomposition developed in the (discontinuous) piecewise linear case, one can consider adapting conventional techniques for nonlinear programming for the general (possibly discontinuous) piecewise differentiable case, as we did above with the projected gradient method for the (possibly discontinuous) piecewise linear case. Indeed, the essence of the concepts introduced in the previous sections does not depend upon the linear nature of the problem.

In this section, we sketch the extension of our work to the nonlinear case. We define a (possibly discontinuous) *piecewise differentiable function* $f : I\!R^n \to I\!R$ to be a function whose derivative is defined everywhere except over a subset of a finite number of sets of the form $\{x \in I\!R^n : r(x) = 0\}$, where r is a differentiable function. In the nonlinear situation, we thus extend the concept of *ridge* to be a specified set $\{x \in I\!R^n : r(x) = 0\}$ containing points where the derivative of f is not defined, where r is a differentiable function (we say also that r is a ridge).

We generalize the definition of decomposition at a point $\hat{x} \in I\!R^n$, so that it expresses the *first-order* behaviour of a piecewise differentiable function in the neighbourhood of $\hat{x}$:

Definition 3 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a (possibly discontinuous) piecewise* differentiable *function with ridges $\{r_i\}_{i\in\mathcal{R}}$, where $\mathcal{R}$ is a finite index set, and let $\hat{x} \in \mathbb{R}^n$. Let $g_{\hat{x}} \in \mathbb{R}^n$ and $\psi_{\hat{x}}$ be a function defined on $\mathbb{R}^n$ such that we have:*

For all $d \in \mathbb{R}^n$ such that $\nabla r_i(\hat{x})^T d \geq 0,\ i \in \mathcal{F}(\hat{x})$, there exists $\delta > 0$ small enough such that:

$$f(\hat{x} + \delta d) = f(\hat{x}) + \delta g_{\hat{x}}^T d + \delta \psi_{\hat{x}}(d) + O(\delta^2),$$

where

$$\psi_{\hat{x}}(d) = \sum_{i\in\mathcal{A}(\hat{x})\setminus\mathcal{F}(\hat{x})} \nu_{\hat{x}}^i \min(0, \nabla r_i(\hat{x})^T d)$$

for some scalars $\{\nu_{\hat{x}}^i\}_{i\in\mathcal{A}(\hat{x})\setminus\mathcal{F}(\hat{x})}$.

We say that $g_{\hat{x}}, \{\nu_{\hat{x}}^i\}_{i\in\mathcal{A}(\hat{x})\setminus\mathcal{F}(\hat{x})}$ is a first-order decomposition of f (into a smooth function and a sum of functions having a single ridge) at $\hat{x}$*.*

We would like to impose restrictions on a piecewise differentiable function so that the nonlinearity of its ridges does not prevent the proofs of the decomposition theorem (Theorem 4) and of the optimality conditions (which become first-order necessary conditions) from being extended. For instance, in order to be able to adapt standard techniques for nonlinear programming to the piecewise differentiable case, we must restrict our considerations to functions having only a finite number of *pieces* (subdomains over which the function is smooth) in the neighbourhood of a given point (in the piecewise linear case the finiteness of the number of pieces was implied by the finiteness of the number of ridges). Nevertheless, a whole class of functions for which the nonlinear analogues of Theorems 4 and 5 hold remains. To give an example, we would expect these results to apply to a function defined over $\mathbb{R}^2$ which is quadratic over each of the subdomains delimited by a circle intersecting an ellipse in the plane.

In the remainder of this section, we briefly describe the main issues one should take into account for the extension of the work to the nonlinear case—the general (possibly discontinuous) piecewise differentiable situation.

In Sections 3 and 4, the algorithm used descent directions attempting to decrease the smooth part of the function while maintaining the value of its nonsmooth part. A first-order algorithm for the nonlinear case could obtain these two objectives *up to first-order changes*, as in the Conn-Pietrzykowski approach to nonlinear optimization, via a penalty function [25].

In the nonlinear case, the matrix of activities, $A(\hat{x})$, has as its columns the vectors $\{\nabla r_i(\hat{x})\}_{i\in\mathcal{A}(\hat{x})}$. A first remark concerns *near-activities*. In order to avoid zigzagging (see an illustration of this in [19]) when dealing with nonlinear ridges, it is necessary to consider projecting onto the tangent hyperplanes, not only when a ridge, r, is active at the current point, $\hat{x}$, but also when $\|r(\hat{x})\|$ is "small". We consider a ridge, r, *ϵ-active* at $\hat{x}$ whenever $|r(\hat{x})| < \epsilon$, where ϵ is some tolerance (which may be reduced as we approach optimality). However, it is then necessary to ensure that near-activities are made active as we approach final convergence. The direction of search is thus made up of two components: the projected gradient direction, h^k, is called the *horizontal step* in contrast with the *vertical step*,

$$v^k \equiv -A_\epsilon(x^k)[A_\epsilon(x^k)^T A_\epsilon(x^k)]^{-1}\Phi(x^k + \alpha^k h^k),$$

which attempts to make active the relevant ridges via a linearization. Here $A_\epsilon(x^k)$ has as its columns the vectors $\{\nabla r_i(x^k)\}_{i\in\mathcal{A}_\epsilon(x^k)}$, where $\mathcal{A}_\epsilon(x^k)$ is the set of the ridges which are near-active at the current iterate, x^k; Φ is the vector of (near-)active ridge values; and α^k is the horizontal step size. In this context of near-activities, we can see that degeneracy cannot be disregarded for nonlinear problems. In nonlinear l_1 data fitting [5], for example, we may expect many more near-activities than the number of dimensions of the problem if the fit is good, particularly earlier on in many algorithms when ϵ may be relatively large. Hence, degeneracy is very likely to occur in such a problem. Moreover, in order to establish whether an iterate is degenerate, we use a notion of *linear ϵ-dependence*, where ϵ is some small tolerance. For example, we can say that a vector v is linearly ϵ-dependent upon a set of vectors $v_1, \ldots, v_k$ if $\|P(\frac{v}{\|v\|})\| < \epsilon$, where P is the orthogonal projector onto the space orthogonal to the space spanned by $v_1, \ldots, v_k$.

The nonlinear line-search algorithm presented in [14] (for the nondifferentiable exact penalty function corresponding to a nonlinear programming problem) can be used here to *estimate* the location of a possible minimum breakpoint along the search direction (in the nonlinear case, the minimum along a search direction need not be a breakpoint). Moreover, one would expect an efficient line search to only find points of sufficient decrease rather than finding the minimum along the line.

The above-mentioned considerations would yield a method converging directly to a local optimum (global convergence), but possessing (in general) only a linear convergence rate (see [25]).

In order to develop a second-order algorithm, assuming now that f is (possibly discontinuous) *piecewise twice-differentiable* (i.e. twice differentiable everywhere except over a finite number of ridges), one must first extend the definition of first-order decomposition to that of *second-order decomposition*.

One could then consider extending the strategies used by Coleman and Conn [14] on the exact penalty function approach to nonlinear programming (although the exact penalty function involves only first-order types of nondifferentiabilities—ridges). The main idea is to attempt to find a direction which minimizes the change in f (up to *second-order terms*) subject to preserving the activities (up to *second-order terms*). Specifically, second-order conditions must be derived [which are the first-order conditions plus a condition on the "definiteness" of the reduced Hessian of the *twice-differentiable part* of f (in the second-order decomposition of f)]. An analogue of the Newton step (or of a modification of Newton's method, see [39]) using a non-orthogonal projection [20] is then taken (or a single-dropping direction is used). An algorithm following these lines would be expected to possess global convergence properties (regardless of starting point) and a fast (2-step superlinear) asymptotic convergence rate as in [13].

8 Numerical Results

We now consider discontinuous optimization problems whose structure is worth exploiting by means of the approach of this paper. We remark that although such discontinuous optimization problems are (often "highly") non-convex, it is not more costly to apply the proposed nonsmooth optimization algorithm directly to the whole domain than restricting it to one particular subdomain where the objective function is continuous.

8.1 Fixed-Charge Application

In many decision-making situations, each task that can be undertaken may have associated with it both a fixed and a variable cost. Whenever the level of such a task is positive, a fixed charge is added to the usual variable cost of the task. If the level of the task, however, is zero, then no fixed charge is incurred.

A typical fixed-charge problem is to minimize a function, $f : \mathbb{R}^n \to \mathbb{R}$, of the form

$$f(x) = \sum_{i=1}^{n} f_i(x_i)$$

subject to linear constraints (including $x \geq 0$), where

$$f_i(x_i) = \begin{cases} c_i x_i + F_i & \text{if } x_i > 0, \\ 0 & \text{if } x_i = 0, \end{cases}$$

the *variable costs*, $c_i \in \mathbb{R}$, and the *fixed costs*, $F_i > 0$, being given for each $i = 1, \ldots, n$. It can be solved using a concave-programming result (a global optimum occurs at an extreme point of the polyhedron defined by the linear constraints) via a branch-and-bound strategy (see [8]).

We tested our algorithm on a fixed-charge problem which originated in determining the production planning for a mixed-model assembly line production system, and which was tackled by Hiraki in [42].

The problem is to minimize the function $f : \mathbb{R}^n \to \mathbb{R}$ defined by

$$f(x) = \sum_{i=1}^{n} f_i(x_i) + \sum_{i=1}^{n} h_i(x_i)$$

and subject to linear constraints (including $x \geq 0$), where the functions f_i and h_i, $i = 1, \ldots, n$, have the form given by the Figures 11 and 12.

Hiraki defines the variables $x_{i,l}$, $1 \leq i \leq n$, $1 \leq l \leq m$ (where m is the number of ridges of f_i and h_i, see Figures 11 and 12), satisfying $0 \leq x_{i,l} \leq 1$, $1 \leq i \leq n$, $1 \leq l \leq m$, to be such that $x_i = \sum_{l=1}^{m} x_{i,l}$, $1 \leq i \leq n$. He considers problems in which the functions f_i, $i = 1, \ldots, n$, are *convex*, so that the conditions:

1. If for some $1 \leq L \leq m$ we have $x_{i,L} > 0$, then $x_{i,l} = 1$ for all $1 \leq l < L$
2. If for some $1 \leq L \leq m$ we have $x_{i,L} = 0$, then $x_{i,l} = 0$ for all $L < l \leq m$

would automatically be satisfied at an optimum found by his algorithm. It is moreover assumed that the functions h_i, $i = 1, \ldots, n$ (defined on Figure 12), are such that

$$\delta_{i,l} - \delta_{i,l-1} < \delta_{i,l+1} - \delta_{i,l}, \quad 1 \leq l \leq m - 1. \tag{48}$$

Hiraki then uses an adaptation of the simplex method to search for an approximate solution at an extreme point (of the polyhedron defined by the linear constraints). As the objective function is not concave (neither f_i nor h_i are concave), a global optimum does not however necessarily occur at an extreme point.

In order to test our discontinuous piecewise linear optimization algorithm (DPLO), we applied it to the set of nine input data as given in [42] to which Hiraki also applied his algorithm. In each of these instances, we have $n = 5$ and $m = 5$. DPLO works in ($n =$) 5 dimensions, while Hiraki's algorithm uses ($n \times m =$) 25 dimensions, m times as many as the DPLO algorithm (where m is the number of constraints and also the number of ridges of the functions f_i's and h_i's). We thus expect our algorithm to be able to address larger problems than Hiraki's (for both methods the iterations' algebra is dominated by simplex-like [active-set] steps: $\mathcal{O}(n^2m^2)$ arithmetic operations per such step for Hiraki's method versus $\mathcal{O}(n^2)$ for the DPLO algorithm).

Due to the presence of a fault at each integer value of each component x_i in this application, the DPLO algorithm stopped in the early stages at a locally optimal solution very near the starting point. In order to prevent our algorithm stopping too early with a poor locally optimal value, in an attempt at globalization, we modified the line search so that a "jump up" is allowed, as long as the point found at the end of the line search decreased the penalty function. Similarly, we considered single-dropping soaring directions from a dead-point iterate, x^k, if it yielded, at the end of the line search, an iterate, x^{k+1}, such that $f_\gamma(x^{k+1}) < f_\gamma(x^k)$, where f_γ is the penalty function (otherwise, the algorithm recovers the previous iterate, x^k, and goes on from it without considering soaring directions any more from x^k). In the worst case, such a strategy can only double the number of line searches performed by the algorithm. Moreover, in order to obtain a local optimum with the same or a better objective function value than Hiraki's algorithm (when we could do so), we often had to try several starting points (we used sequentially the starting points $(5,5,5,5,5), (4,4,4,4,4), \ldots, (0,0,0,0,0)$ and then combinations of such integer coordinates).

Table 1 displays the optimal values obtained by Hiraki's and the ones obtained by our approach. The last column of the table displays the number of different starting points we had to try to obtain the value of the objective value shown under "DPLO" (the '*' simply indicates which algorithm achieved the best solution for each case). For five instances out of the nine considered, we could find a better objective function value than the one found by Hiraki's algorithm (trying between 1 and 7 different starting points before we could obtain a better objective function value than Hiraki's). In one case we obtained (at the second trial—starting point) the same objective function value as Hiraki's. We however tried 8 more starting points, including 3 yielding again the same local optimal value as Hiraki's, without being able to improve our locally optimal value. Finally, in three cases we could not do as well as the solutions reported in Hiraki's paper, after trying 10 different starting points (including different initial penalty parameter values). (Note that as these three cases have in common their set of constraints, a possible explanation could be that Hiraki's search through extreme points has been favoured by the shape of the corresponding polyhedron.)

The computer used by Hiraki is a HITAC 8700 with OS/7. The DPLO algorithm was programmed in Fortran, using subroutines from the NAG Fortran library, and run on a Sun 3/160. The DPLO algorithm necessitated between 0.78 and 4.24 seconds CPU time (and between 4 and 24 iterations) for *each* starting point tried. The CPU time indicated

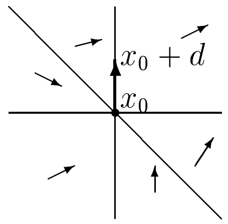

Figure 7: Degeneracy in the context of non-convexity

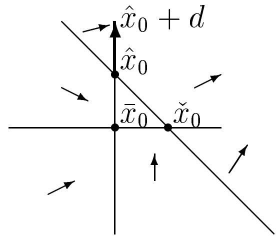

Figure 8: Associated perturbed subproblem

Data	Hiraki		DPLO			
no.	value	CPU	value	CPU	iter	start pts
case 1-1	2878.33	20.7	2870.00*	1.5	-	1
case 1-2	4095.95	34.9	4066.00*	0.9	6.7	3
case 1-3	4097.38	21.0	3967.85*	1.6	8.6	5
case 2-1	3456.67	20.3	3425.00*	1.8	11.0	7
case 2-2	3480.00*	16.2	3480.00*	1.4	9.0	2
case 2-3	3318.33	21.2	3290.00*	1.5	6.0	1
case 3-1	3275.00*	23.5	3538.67	2.1	10.4	1
case 3-2	3743.00*	57.8	3811.33	2.0	10.3	6
case 3-3	4215.00*	47.0	4218.57	2.0	11.0	1

Table 1: Numerical results (fixed charge)

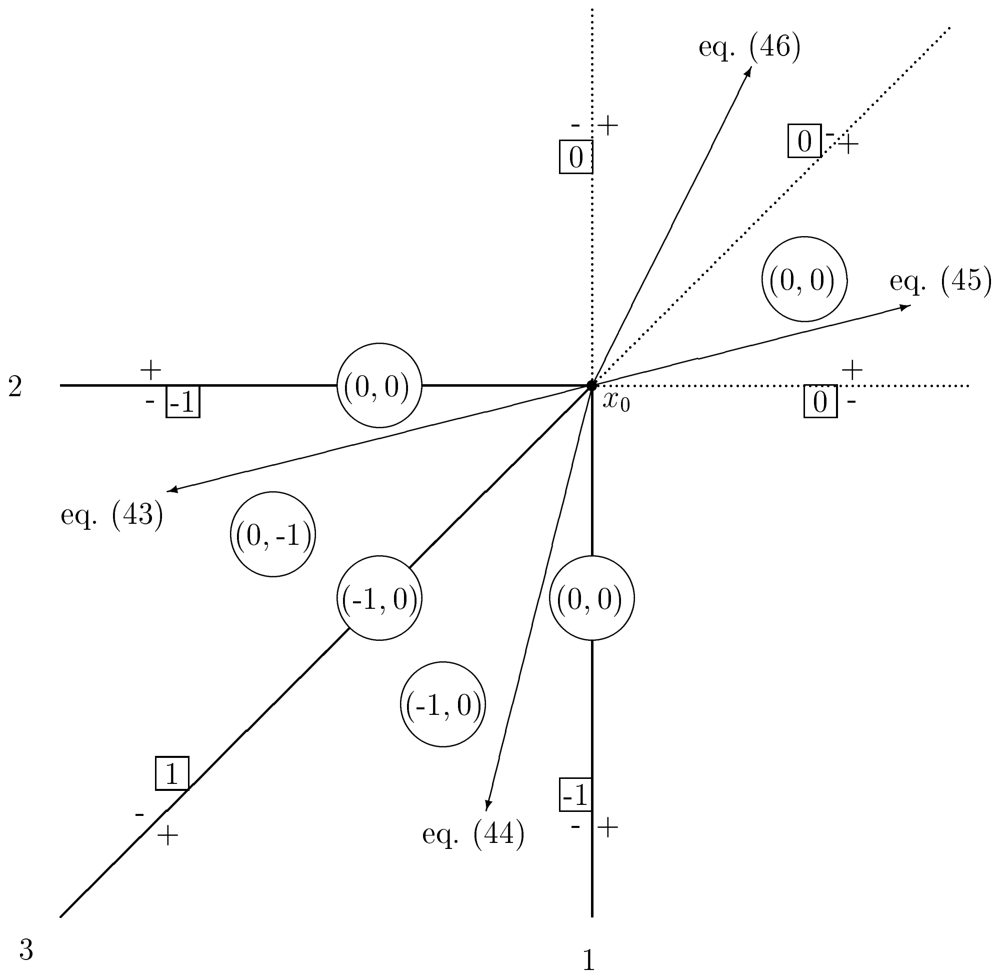

Figure 9: Function non-decomposable at a degenerate point

in table 1 for the DPLO algorithm is the average (over all runs tried for a particular instance) CPU time (in seconds) required for a *single* run of our algorithm (from *one* starting point). Also, "iter" indicates the average number of iterations required (not available for Hiraki's algorithm).

Even when taking into account the fact that we needed multiple starting points to obtain an optimal value as good as or better than that found by Hiraki's algorithm, on these examples, our algorithm compares very well with Hiraki's approach to this problem (Hiraki starts from *one* initial point, given by the solution of some associated simplex problem). The number of starting points we used is arbitrary but small: a decent global-optimization heuristic using DPLO would certainly yield a reasonable algorithm for this problem. Recall that our original algorithm is looking for *local* optima. We integrated to it here two primitive heuristics (jump-up allowing and multi-start) in order to show that the DPLO algorithm can be useful, e.g. within some global optimization scheme, to deal directly with nonsmoothnesses and discontinuities. Furthermore, the DPLO algorithm did not take advantage of any of the simplifying assumptions on properties of the functions h_i's [equation (48)], on the convexity of the functions f_i's, and on the separability of the objective function, all of which are essential to Hiraki's algorithm. It is also worth mentioning that these assumptions of Hiraki are rather artificial. Erenguc and Benson [29] notice that in several practical problems involving tasks with fixed costs, the separable model of the fixed-charge problem is inappropriate: Whenever a task in a group of tasks that share resources is undertaken at a positive level, a fixed cost is incurred. However, the value of this fixed cost may depend upon which other tasks in the group are also undertaken at a positive level. Such applications yield a *non-separable* objective function, which can be handled without difficulty by our algorithm. Erenguc and Benson present a branch-and-bound algorithm to solve such an *interactive* fixed-charge problem, as they consider only objective functions which are concave (and thus, a (global) optimum is known to exist at an extreme point of the polyhedron defined here by m linear constraints and upper and lower bounds). Our algorithm can find (although only local) optima of more general problems, as it does not rely on such a concave assumption, while Erenguc and Benson's algorithm cannot tackle non-concave problems. We are presently experimenting with the DPLO algorithm on the interactive fixed-charge transportation problems (presented in [29]) and on non-concave-objective variants of it.

8.2 Discharge Allocation for Hydro-Electric Generating Stations

For this application, we are concerned with the allocation of discharge of water among generating units of a hydro-electric generating station so as to maximize power output within operating constraints. We consider the case where there are different types of units (for the allocation problem with identical units, see [37] and [44] for example). To each type of unit corresponds a particular power-output function of the discharge.

The problem is

$$\min \quad -\sum_{i=1}^{n} P_i(x_i)$$

$$\begin{aligned}\text{subject to } \sum_{i=1}^{n} x_i &= Q \\ 0 \leq x_i &\leq M_i, \quad i = 1, \ldots, n\end{aligned}$$

where n is the number of units, x_i is the discharge allocated in unit i, $P_i(x_i)$ is the power produced by a discharge of x_i through unit i, Q is the total discharge through the station, and M_i is an upper bound on the discharge allowed through unit i. Economic and other operating considerations, such as vibration control, require that if a unit i is operated, it must discharge at least L_i meters cube per second (m^3/s). We thus consider a discontinuous model, so that for each type of unit, the output-discharge curve has the form given by Figure 13.

In [44], Klein & Sim solve a piecewise linear approximation to the discharge-allocation problem with a hybrid separable-programming/heuristic algorithm. This piecewise linear approximation is in fact the linear interpolation of data from power measurements at 20 different values of discharge through a generating unit. We experimented using the DPLO algorithm on eigthteen instances of the same piecewise linear approximation problem having two different types of units (two different output-discharge curves): type A and type B. Each instance involves $n = 16$ units, a given total discharge Q, and a number of units of type A, the remaining units being of type B.

We again tested our method by comparing the results we obtained with Klein & Sim's solutions to these 18 instances. Their hybrid separable-programming/heuristic algorithm starts from *one* initial point, obtained through the solution of an associated separable programming problem. In order to reach a solution equivalent (same value of the objective function) to that obtained by Klein & Sim, the DPLO algorithm however required us again to try between 3 and 12 different starting points (recall that our penalty approach does not require feasible initial points) for each instance, and we again took our best local optimum. These starting points were chosen following a crude heuristics estimating the total number of units that should be operating. We then tried different permutations of operating units of type A and of type B. Table 2 displays, for each of the 18 instances considered, the number of iterations required by Klein & Sim under the column "K&S" (CPU time not available) and the number of iterations required by our algorithm (under the column "PL model") for the single run that found our best local optimum, together with the corresponding CPU time in seconds. Again despite the fact that we needed multiple starting points, the number of iterations required by DPLO compares well with that of Klein & Sim. Notice moreover that each of the (simplex-like) iterations required by Klein & Sim's algorithm are in dimension $n \times G$ (where G is the number of grid points per output-discharge curve—here $G = 20$) whereas such simplex-like iterations performed by the DPLO algorithm are only in n dimensions. The same modification of the line search (described in the fixed-charge application) allowing some soaring directions was used in order to prevent our algorithm stopping too early with a poor locally optimal value.

In fact there are two instances where Klein & Sim achieved a better (by less than 0.001% in the objective value) solution than our best local optimum, and there is one instance where the DPLO algorithm found a better (again by less than 0.001%) solution than Klein & Sim. Note that in each of these three cases, our respective solutions involve

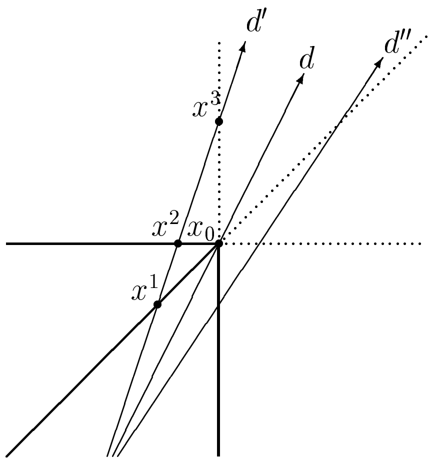

Figure 10: Perturbation of the search direction

Data no.	K&S iter	PL model iter	PL model CPU	PQ model iter	PQ model CPU
1	119	33	2	19 (6)	1
2	87	35	9	17 (0)	1
3	108	35	10	20 (4)	1
4	91	34	9	17 (0)	0.7
5	96	31	9	20 (2)	0.7
6	68	22	5	15 (2)	0.7
7	117	30	12	17 (6)	0.8
8	145	42	18	12 (0)	0.5
9	90	39	2.5	15 (6)	0.5
10	99	39	-	20 (4)	0.8
11	95	43	3	24 (4)	0.9
12	106	40	5	21 (6)	0.9
13	92	34	2	20 (6)	0.9
14	108	26	2	17 (6)	0.8
15	79	30	2	6 (0)	0.3
16	78	30	1	14 (0)	0.6
17	92	19	2	17 (6)	0.6
18	142	43	4	24 (6)	0.9

Table 2: Numerical results (discharge allocation)

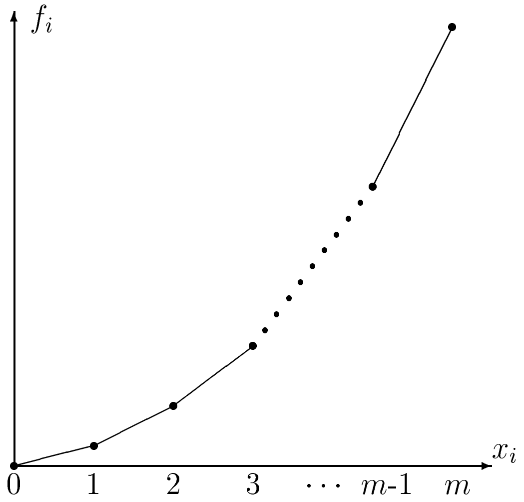

Figure 11: Form of the function f_i

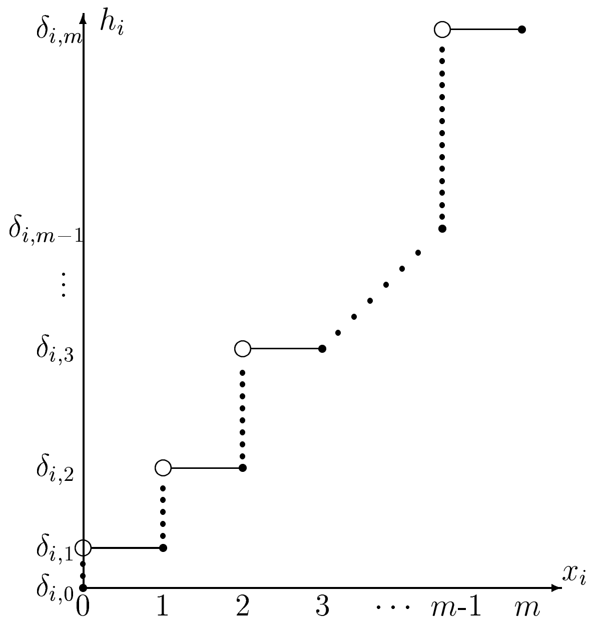

Figure 12: Form of the function h_i

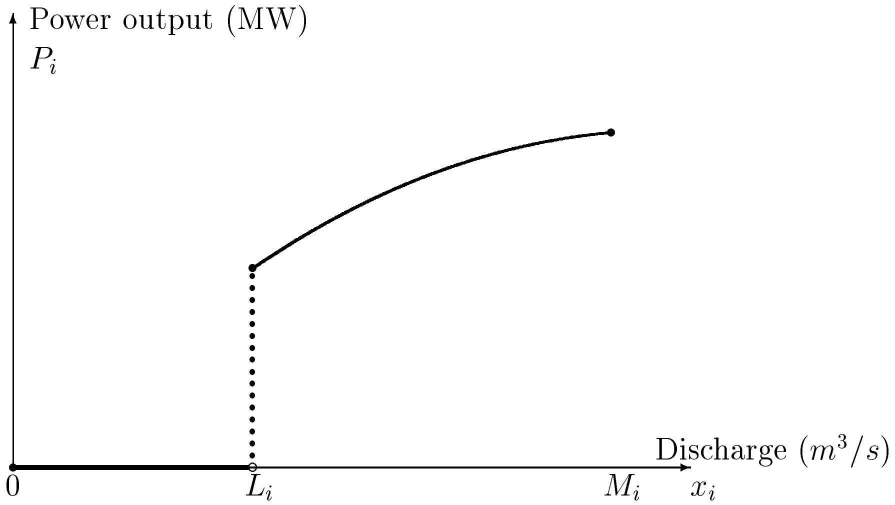

Figure 13: Output-discharge curve for a hydro-electric generating unit

the same combination of operating units, differing only with respect to the allocation of the discharge among the units that both solution agreed to be operating. We are able to verify with the DPLO algorithm that Klein & Sim's solutions were indeed local optima. These three discrepancies are thus due to some non-convexity on the subdomain corresponding to the optimal combination of operating units (and over which the objective function is continuous).

Since it is known that in the operating domain (i.e. where the discharge is at least L_i m^3/s) the quantity, P_i, of electricity generated by one particular generating unit i is a quadratic polynomial function of x_i (the water drafted through unit i), we experimented further directly on this (more realistic) nonlinear model.

To do so, we adapted the DPLO algorithm to a first-order discontinuous piecewise differentiable algorithm. This simple special case of a nonlinear discontinuous optimization problem however only involves *linear* ridges. Hence, the issues of *near-activities* (to avoid zigzagging between nonlinear ridges) and *vertical step* (which attempts to recover the nonlinear-ridge activities we wanted to preserve) described in Section 7 are irrelevant.

The main adaptation required for this nonlinear problem was in the line search used to compute a suitable step size yielding sufficient decrease (recall that in the nonlinear case, the minimum along a search direction need not be a breakpoint). Our line search combines the ideas of Coleman and Conn's nondifferentiable line-search subroutine [14] (for the nondifferentiable exact penalty function) and those of Fletcher's nonlinear line-search algorithm [32].

Another difference with the DPLO algorithm is in fact a consequence of using such an inexact line search: We attempt at first to satisfy only *approximately* the first-order necessary conditions. More precisely, we consider that the current iterate is a dead point when the projected restricted gradient is smaller than some rather loose tolerance, $\Lambda > 0$ (in our computational experiments, we initially set $\Lambda = 0.1$). This means that at a current iterate, x^k, which is found to be a dead point according to the above relaxed criterion, we rely on cruder estimates of the multipliers $\{u_i\}_{i \in \mathcal{A}(x^k)}$. As long as we are "far" from optimality, we do not wish to compute accurately these u_i's, which are used in practice as indicators of which ridges should be active at optimality and which ones should be dropped. Roughly speaking, the point is also to avoid spending too much time to obtain precisely a stationary point when we are still in the wrong subspace. Once we have identified the ridges which are active at a point satisfying the relaxed first-order necessary conditions, we then tighten the projected-restricted-gradient tolerance (dividing it sequentially by 10 until $\Lambda = 10^{-5}$, in the computational experiments we are reporting).

We applied this linearly-ridged discontinuous piecewise differentiable optimization algorithm to the same 18 instances as above, but using this time *nonlinear* output-discharge curves for each of types A and B. We in fact generated these curves, to test our nonlinear-case algorithm, by fitting a quadratic through a set of output-discharge data. Note that with this piecewise quadratic model, each function P_i now only has one ridge (which is a fault) instead of 20 (one fault plus 19 breakpoints).

The total number of iterations required by our algorithm (for the single run that found our best local optimum) and the corresponding CPU time in seconds are displayed in table 2 under the column "PQ model". The number in parentheses is the number of iterations performed with tightened projected-restricted-gradient tolerances, i.e. once the ridges expected to be active at optimality are identified. Again, we took our best local optima among between 3 and 12 different starting points—trials for each instance. On average, approximately one ninth of the CPU time (compared with the piecewise linear case) was necessitated for each trial. Note that the quadratic fits happened to be convex on each subdomain corresponding to a given combination of operating units (i.e. on each cell in this case).

The next stage we are considering is to implement a specialized version of the second-order algorithm outlined in Section 7: a Newton step is attempted to minimize the objective function restricted to the space preserving the current activities, making use of the reduced Hessian of the *twice-differentiable part* of the objective function (in the same manner the first-order algorithm relied on the reduced-projected-gradient direction).

9 Conclusions

We introduced in this paper the concepts of the *decomposition* of a function into a smooth part and a nonsmooth part, the decomposition theorem, which states that a decomposition always exists at non-degenerate points, and a proposition which *explicitly* gives this decomposition. This is in contrast with other work such as [64] which assumes that sufficient information is provided by the specification of the objective function so that the local decomposition can be written down.

Optimality conditions and a descent algorithm have then been derived from the decomposition of the function. Easy generalization of these ideas to the discontinuous situation then followed in Section 4 by restricting the decomposition of the function to *non-soaring* directions (at non-contact points). We have hence set a framework for an algorithm dealing directly with discontinuities that could be involved in a class of non-differentiable optimization problems: namely those problems involving a decomposable objective function.

The use of exact penalty methods has been twofold in this work: as a mean to reduce the constrained discontinuous piecewise differentiable problem to the unconstrained case, and as a seed for the development of our theory, since we generalized the basic ideas used by Conn [19] to deal with the nonsmoothness introduced by the use of an l_1 exact penalty function.

We described techniques to cope with problems that occur in certain cases where the hypothesis of non-degeneracy is not satisfied at points encountered in the course of the algorithm. We encountered no difficulties in our computational experiments, although serious problems can still arise at certain singular points (contact points and dead-point iterates, at which the objective function is not decomposable). The theoretical treatment of such points has to be considered in any meaningful step beyond the convex case.

Comparison of our work with recent developments in nondifferentiable optimization, for example Lemaréchal's bundle methods [46] or composite nonsmooth optimization algorithms (see [2, 32, 65]), deserves attention in future work. In particular, it would be interesting to compare Clarke's more theoretical approach stemming from convex optimization theory and nonsmooth analysis [12], with that introduced in this paper, which is based on methods more oriented to numerical implementations and on practical considerations.

The inescapable non-convexity feature of discontinuous functions gives rise to the existence of several local optima in discontinuous optimization problems. We could not expect to address fully this difficult issue of global optimization. The algorithm presented is looking for *local* optima. It can however be used as an adjunct to some heuristic or global optimization method for discontinuous optimization problems whose combinatorial nature is ultimately dominant. It gives a framework allowing the optimizer to deal directly with the nonsmoothnesses and discontinuities involved, and thereby, improve solutions found by heuristic methods, when this is possible. Results from the computational experimentation of a stochastic global optimization method integrating our algorithm are to be reported shortly [38].

The main contribution of this work is to allow one to tackle practical problems in which discontinuities are involved and the introduction of the theoretical concepts required. We generalized previous work (see for example Fourer's piecewise linear optimization algorithm [33, 34, 35]) not only to the discontinuous situation but also to the non-separable and certain (decomposable) non-convex cases. Above all, an advantage of our active-set approach over Fourer's simplex-format algorithm is that it makes our algorithm readily extendable to the nonlinear case, by adapting conventional techniques for nonlinear programming. Although developed in detail for the (discontinuous) piecewise *linear* case, the ideas we presented indicate a theoretical framework for the elaboration of practical algorithms for a more general (discontinuous) piecewise differentiable (nonlinear) problem. This will be pursued in our future research.

Acknowledgement

The authors would like to thank Adrian Lewis who simplified the initial proof of Theorem 1.

References

[1] K. D. Anderson. An efficient Newton barrier method for minimizing a sum of Euclidean norms. *SIAM Journal on Optimization*, 6(1):74–95, 1996.

[2] T. Bannert. A trust region algorithm for nonsmooth optimization. *Mathematical Programming*, 67:247–264, 1994.

[3] R. Bartels, C. Charalambous, and A. R. Conn. On Cline's direct method for solving overdetermined linear systems in the l_∞ sense. *SIAM Journal on Numerical Analysis*, 15(1):255–270, 1978.

[4] R. H. Bartels and A. R. Conn. Linearly constrained discrete l_1 problems. *ACM Transactions on Mathematical Software*, 6(4):594–608, 1980.

[5] R. H. Bartels and A. R. Conn. An approach to nonlinear l_1 data fitting. In J. P. Hennart, editor, *Proceedings of the Third Mexican Workshop on Numerical Analysis*, pages 48–58. Springer Verlag, 1981.

[6] R. H. Bartels, A. R. Conn, and Y. Li. Primal methods are better than dual methods for solving overdetermined linear systems in the l_∞ sense? *SIAM Journal on Numerical Analysis*, 26:693–726, 1989.

[7] A. Ben-Tal and M. P. Bendsoe. A new method for optimal truss topology design. *SIAM Journal on Optimization*, 3(2):322–358, 1993.

[8] H. P. Benson. A finite algorithm for concave minimization over a polyhedron. *Naval Research Logistics Quarterly*, 32:165–177, 1985.

[9] P. H. Calamai and A. R. Conn. A stable algorithm for solving the multifacility location problem involving Euclidean distances. *SIAM Journal on Scientific and Statistical Computing*, 1(4):512–525, 1980.

[10] P. H. Calamai and A. R. Conn. A projected Newton method for l_p norm location problems. *Mathematical Programming*, 38(1):75–109, 1987.

[11] C. Cheng and E. Kuh. Module placement based on resistive network optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 3:218–225, 1984.

[12] F. H. Clarke. *Optimization and Nonsmooth Analysis*. John Wiley, 1983.

[13] T. F. Coleman and A. R. Conn. Nonlinear programming via an exact penalty function: Asymptotic analysis. *Mathematical Programming*, 24:123–136, 1982.

[14] T. F. Coleman and A. R. Conn. Nonlinear programming via an exact penalty function: Global analysis. *Mathematical Programming*, 24:137–161, 1982.

[15] T. F. Coleman and L. A. Hulbert. A globally and superlinearly convergent algorithm for convex quadratic programs with simple bounds. *SIAM Journal on Optimization*, 3(2):298–321, 1993.

[16] T. F. Coleman and Y. Li. A global and quadratic affine scaling method for (augmented) linear l_1 problems. In D. Griffiths and G. Watson, editors, *Proceedings of the 13th Biennial Numerical Analysis Conference Dundee 1989*. Longmans, 1990.

[17] T. F. Coleman and Y. Li. A global and quadratically convergent method for linear l_∞ problems. *SIAM Journal on Numerical Analysis*, 29(4):1166–1186, 1992.

[18] T. F. Coleman and Y. Li. A globally and quadratically convergent affine scaling method for linear l_1 problems. *Mathematical Programming*, 56:189–222, 1992.

[19] A. R. Conn. Constrained optimization using a nondifferentiable penalty function. *SIAM Journal on Numerical Analysis*, 10:760–784, 1973.

[20] A. R. Conn. Projection matrices - A fundamental concept in optimization. In Vogt and Mickle, editors, *7th Annual Conference in Modelling and Simulation, Ed. April 26-27, 1976*, pages 599–605, 1976.

[21] A. R. Conn. Nonlinear programming, exact penalty functions and projection techniques for non-smooth functions. In P. T. Boggs, R. H. Byrd, and R. B. Schnabel, editors, *Numerical Optimization 1984 - Proceedings of the SIAM Conference on Numerical Optimization, Boulder*, pages 3–25, 1985.

[22] A. R. Conn and G. Cornuéjols. A projection method for the uncapacitated facility location problem. *Mathematical Programming*, 46:273–298, 1990.

[23] A. R. Conn and Y. Li. A structure exploiting algorithm for nonlinear minimax problems. *SIAM Journal on Optimization*, 2(2):242–263, 1992.

[24] A. R. Conn and M. L. Overton. A primal-dual interior point method for minimizing a sum of Euclidean distance, 1994. Unpublished.

[25] A. R. Conn and T. Pietrzykowski. A penalty function method converging directly to a constrained optimum. *SIAM Journal on Numerical Analysis*, 14(2):348–375, Apr. 1977.

[26] D. De Wolf, O. J. de Bisthoven, and Y. Smeers. The simplex method extended to piecewise-linearly constrained problems I: The method and an implementation. Technical report, Center for Operations Research and Econometrics, Louvain-La-Neuve, Belgium, 1991.

[27] D. De Wolf, O. J. de Bisthoven, and Y. Smeers. The simplex method extended to piecewise-linearly constrained problems II: An application to the gas transmission problem. Technical report, Center for Operations Research and Econometrics, Louvain-La-Neuve, Belgium, 1991.

[28] G. Di Pillo and L. Grippo. Exact penalty functions in constrained optimization. *SIAM Journal on Control and Optimization*, 27(6):1333–1360, 1989.

[29] S. S. Erenguc and H. P. Benson. The interactive fixed charge linear programming problem. *Naval Research Logistics Quarterly*, 33:157–177, 1986.

[30] A. V. Fiacco and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley, 1968. (Reprinted SIAM, 1990).

[31] R. Fletcher. A model algorithm for composite nondifferentiable optimization problems. *Mathematical Programming Study*, 17:67–76, 1982.

[32] R. Fletcher. *Practical Methods of Optimization.* Wiley-Interscience, second edition, 1987.

[33] R. Fourer. A simplex algorithm for piecewise-linear programming I: Derivation and proof. *Mathematical Programming*, 33:204–233, 1985.

[34] R. Fourer. A simplex algorithm for piecewise-linear programming II: Finiteness, feasibility and degeneracy. *Mathematical Programming*, 41:281–315, 1988.

[35] R. Fourer. A simplex algorithm for piecewise-linear programming III: Computational analysis and applications. *Mathematical Programming*, 53(2):213–235, 1992.

[36] A. B. Gamble, A. R. Conn, and W. R. Pulleyblank. A network penalty method. *Mathematical Programming*, 50:53–73, 1991.

[37] J. Gauvin, P. Parent, and G. Savard. Répartition optimale de la puissance dans une centrale hydraulique à réserve pompée. *RAIRO Recherche opérationnelle/Operations Research*, 20(1):1–18, 1986.

[38] M. Gendreau and M. Mongeau. General interactive fixed-charge piecewise-linear programming using tabu search. Technical report, Centre de recherche sur les transports, Université de Montréal. In preparation.

[39] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization.* Academic Press, 1981.

[40] J. Hald and K. Madsen. Combined LP and quasi-Newton methods for minimax optimization. *Mathematical Programming*, 20:49–62, 1981.

[41] S. P. Han. Variable metric methods for minimizing a class of nondifferentiable functions. *Mathematical Programming*, 20:1–13, 1981.

[42] S. Hiraki. A simplex procedure for a fixed charge problem. *Journal of the Operations Research Society of Japan*, 23(3):243–266, 1980.

[43] I. I. Imo and D. J. Leech. Discontinuous optimization in batch production using SUMT. *International Journal of Production Research*, 22(2):313–321, 1984.

[44] E. M. Klein and S. H. Sim. Discharge allocation for hydro-electric generating stations. *European Journal of Operational Research*, 73(1):132–138, 1994.

[45] D. E. Knuth. *The Art of Computer Programming, Vol. 3, Sorting and Searching.* Addison-Wesley, 1975.

[46] C. Lemaréchal. Bundle methods in nonsmooth optimization. In C. Lemaréchal and R. Mifflin, editors, *Proceedings of the IIASA Workshop, Nonsmooth Optimization, March 28-April 8, 1977*, volume 3, pages 79–102. Pergamon Press, 1978.

[47] D. M. Mates. *A projection method for the floor planning problem.* PhD thesis, Dept. of Combinatorics & Optimization, University of Waterloo, Ontario, Canada, 1993.

[48] M. Mongeau. *Discontinuous Piecewise Linear Optimization.* PhD thesis, Dept. of Combinatorics & Optimization, University of Waterloo, Ontario, Canada, 1991.

[49] B. Montreuil, H. D. Ratliff, and M. Goetschalckx. Matching based interactive facility layout. *AIIE Transactions*, 19(3):271–279, 1987.

[50] W. Murray and M. Overton. A projected Lagrangian algorithm for nonlinear l_1 optimization. *SIAM Journal on Scientific and Statistical Computing*, 2:207–224, 1981.

[51] M. R. Osborne. *Finite Algorithms in Optimization and Data Analysis.* John Wiley, 1985.

[52] M. R. Osborne, S. Pruess, and R. S. Womersley. Concise representation of generalised gradients. *Journal of the Australian Mathematical Society*, 28:57–74, 1986.

[53] M. R. Osborne and G. A. Watson. An algorithm for minimax approximation in the nonlinear case. *Computing Journal*, 12:63–68, 1968.

[54] M. L. Overton and R. S. Womersley. On minimizing the spectral radius of a nonsymmetric matrix function: Optimality conditions and duality theory. *SIAM Journal on Matrix Analysis and Applications*, 9(4):473–498, 1988.

[55] M. L. Overton and R. S. Womersley. On the sum of the largest eigenvalues of a symmetric matrix. *SIAM Journal on Matrix Analysis and Applications*, 13(1):41–45, 1992.

[56] M. L. Overton and R. S. Womersley. Optimality conditions and duality theory for minimizing sums of the largest eigenvalues of symmetric matrices. *Mathematical Programming*, 62:321–357, 1993.

[57] U. S. Palekar, M. H. Karwan, and S. Zionts. A branch-and-bound method for the fixed charge transportation problem. *Management Science*, 36(9):1092–1105, 1990.

[58] D. M. Ryan and M. R. Osborne. On the solution of highly degenerate linear programmes. *Mathematical Programming*, 41:385–392, 1988.

[59] A. Tishler and I. Zang. A switching regression method using inequality conditions. *Journal of Econometrics*, 11:259–274, 1979.

[60] G. Vijayan and R.-S. Tsay. A new method for floorplanning using topological constraint reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(12), 1991.

[61] P. Wolfe. A technique for resolving degeneracy in linear programming. *SIAM Journal*, 11:205–211, 1963.

[62] R. S. Womersley. Optimality conditions for piecewise smooth functions. *Mathematical Programming Study*, 17:13–27, 1982.

[63] R. S. Womersley. Local properties of algorithms for minimizing nonsmooth composite functions. *Mathematical Programming*, 32:69–89, 1985.

[64] R. S. Womersley. Censored discrete linear l_1 approximation. *SIAM Journal on Scientific and Statistical Computing*, 7(1):105–122, 1986.

[65] R. S. Womersley and R. Fletcher. An algorithm for composite nonsmooth optimization problems. *Journal of Optimization Theory & Applications*, 48:493–523, 1986.

[66] I. Zang. Discontinuous optimization by smoothing. *Mathematics of Operations Research*, 6(1):140–152, 1981.

[67] W. I. Zangwill. An algorithm for the Chebyshev problem - with an application to concave programming. *Management Science*, 14(1):58–78, 1967.

[68] J. Zowe. Nondifferentiable optimization. In K. Schittkowski, editor, *Computational Mathematical Programming*, pages 323–356. NATO Advanced Science Institute Series F: 15, Bad Windsheim, July 1984, Springer-Verlag, 1985.