An Update Rule and a Convergence Result for a Penalty Function Method

Regina S. Burachik* C. Yalçın Kaya†

January 29, 2007

Dedicated to the memory of Professor Alexander M. Rubinov

Abstract

We use a primal-dual scheme to devise a new update rule for a penalty function method applicable to general optimization problems, including nonsmooth and nonconvex ones. The update rule we introduce uses dual information in a simple way. Numerical test problems show that our update rule has certain advantages over the classical one. We study the relationship between exact penalty parameters and dual solutions. Under the differentiability of the dual function at the least exact penalty parameter, we establish convergence of the minimizers of the sequential penalty functions to a solution of the original problem. Numerical experiments are then used to illustrate some of the theoretical results.

Key words: Penalty function method, penalty parameter update, least exact penalty parameter, duality, nonsmooth optimization, nonconvex optimization.

Mathematical Subject Classification: 49M30; 49M29; 49M37; 90C26; 90C30.

1 Introduction

A penalty function method solves a constrained optimization problem by transforming it into a sequence of unconstrained ones. A detailed survey of penalty methods and their applications to nonlinear programming can be found in [2, 6, 4] and the references therein. In these methods, the original constrained problem is replaced by an unconstrained problem, whose objective function is the sum of a certain "merit" function (which reflects the objective function of the original problem) and a penalty term which reflects the constraint set.

*School of Mathematics and Statistics, University of South Australia, Mawson Lakes, S.A. 5095 Australia; and Centre for Informatics and Applied Optimization, University of Ballarat, Victoria, Australia. E-mail: regina.burachik@unisa.edu.au

†School of Mathematics and Statistics, University of South Australia, Mawson Lakes, S.A. 5095 Australia. E-mail: yalcin.kaya@unisa.edu.au

The merit function is chosen in general as the original objective function, while the penalty term is obtained by multiplying a suitable function, which represents the constraints, by a positive parameter d, called the *penalty parameter*. A given penalty parameter $\bar{d}$ is called an *exact* penalty parameter when every solution of the original problem can be found by solving the unconstrained optimization problem with the *penalty function* associated with $\overline{d}$.

The penalty approach showed to be a powerful tool from a theoretical point of view (see, e.g., [2] for a detailed survey of theoretical applications of penalty methods). Furthermore, all the fundamental notions of the theory of constrained optimization can be developed using the exact penalty function approach (see [4]).

Various kinds of penalty techniques have been proposed and studied in the past four decades. In most of these techniques the exact penalty parameter is found by gradually increasing the value of d, until the *penalty threshold* or *least exact penalty parameter* is reached. However, this procedure is likely to be time-consuming and to introduce numerical ill-conditioning because of too large a value of an exact penalty parameter reached, which results in inaccuracies in the solution.

In order to avoid ill-conditioning, it is proposed in [12, 13, 10] a dynamic update of the penalty parameter, based on dual information. These works analyse the case of convex and smooth problems. This poses the natural question of whether a penalty update can be designed, such that it uses dual information successfully, in the absence of convexity and/or smoothness assumptions.

In [17, 18], Rubinov and his co-workers proposed a new kind of (generalized) penalty function for nonsmooth and nonconvex problems. Their scheme possesses an exact penalty parameter which turns out to be relatively small. A specific formula for the least exact penalty parameter is also presented in these works. The availability of a specific formula for the penalty threshold is very interesting from the theoretical point of view. Moreover, a small exact penalty parameter provided by this formula opens the way to avoiding numerical instabilities. However, it is not possible (in general) to actually use the formula and compute the value of the least exact penalty parameter. Thus it is still necessary to carry out the classical procedure (of gradually increasing the value of the parameter) in order to reach (and cross over) the exact penalty threshold. This further motivates the question of whether there is a simple, direct way to use dual information for the penalty update for nonsmooth and nonconvex problems.

A major aim of our study is to propose such an update, which would avoid numerical instabilities inherent to the gradual increase of the penalty parameter. For this purpose, first, we introduce a duality scheme for the original problem, using the *sharp Lagrangian*. This duality scheme has zero duality gap thanks to [15, Theorem 11.59]. Then we use a primal-dual scheme called the Modified Subgradient (MSG) algorithm, which was recently introduced in [8, 9] for tackling nonsmooth and nonconvex optimization problems subject to equality constraints. The MSG algorithm was further studied in [3], where convergence of the dual variables to a dual solution was proved. We use the updates of the MSG algorithm for deriving a simple update formula for the penalty parameter.

Another aim of this article is to study the relationship between exact penalty parameters and dual solutions. In our setting, the set of exact penalty parameters is an interval which is unbounded above. We prove here that the infimum of this interval, which we call $\overline{d}_{\min}$, may not be an exact penalty parameter for a given problem. More precisely, we prove that $\overline{d}_{\min}$ is an exact penalty parameter if and only if the dual function is differentiable at $\overline{d}_{\min}$. Moreover, under the latter assumption we establish our main convergence result: If $d_k \uparrow \overline{d}_{\min}$ and x_k is a minimizer of the penalty function associated with the penalty parameter d_k, then every accumulation point of $\{x_k\}$ solves the original problem. To our knowledge, this convergence result is not available for nonsmooth and nonconvex problems.

For the nonsmooth and nonconvex case, it is proved in [3] that all accumulation points of an *auxiliary* sequence $\{\widetilde{x}_k\}$ are optimal. The iterates $\widetilde{x}_k$ are minimizers of the penalty problems for penalty parameter $d_k + \eta$, where $\eta > 0$ is arbitrary but fixed. In [3] it is also proved that, in general, the accumulation points of the sequence x_k (of minimizers of the penalty problems for penalty parameter d_k) may not be a solution of the original problem. However, the numerical instabilities produced when $d_k + \eta > \overline{d}_{\min}$ justify the need for a sharper result of the kind presented now.

In the numerical experiments, we compare our update rule with the classical one through two test problems, and show that our update rule may avoid the above mentioned numerical difficulties associated with penalty function methods. We also illustrate further advantages, especially when the penalty parameter can take a negative value. In both problems, we verify our theoretical results by constructing the graphs of H and its derivative.

Our paper is organized as follows. In Section 2 we state the problem, give the basic notation and describe the Lagrangian scheme. Also in this section we list some existing results which will be used in further sections. In Section 3 we present our theoretical results. In Section 4 we recall the MSG algorithm and introduce our penalty update rule. We describe our numerical experiments in Section 5, while the concluding remarks are given in Section 6.

2 Preliminaries

We consider the nonlinear programming problem:

$$(P) \qquad \text{minimize } f_0(x) \text{ over all } x \text{ in } X \text{ satisfying } f^+(x) = 0,$$

where X is a compact subset of $\mathbb{R}^n$, and the functions $f_0 : \mathbb{R}^n \to \mathbb{R}$ and $f^+ : \mathbb{R}^n \to \mathbb{R}_+$ are continuous. Note that problems with both equality and inequality constraints can be transformed into the format (P) in a standard way by using the operator $a^+ := \max\{0, a\}$. While this transformation does not preserve differentiability and/or convexity of the problem, it clearly preserves the continuity of the data.

Our duality scheme enjoys zero duality gap, as a consequence of [15, Theorem 11.59]. Before quoting this result, let us recall some definitions from [15]. Let $\varphi : \mathbb{R}^n \to \mathbb{R}_{+\infty} := \mathbb{R} \cup \{+\infty\}$ and consider the optimization problem

$$\min_{x \in \mathbb{R}^n} \varphi(x). \tag{1}$$

A function $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\pm\infty} := \mathbb{R} \cup \{\pm\infty\}$ is said to be a *duality parameterization* for problem (1) when $\varphi(\cdot) = g(\cdot, 0)$. The *perturbation function* associated with g is $p(v) := \inf_{x\in\mathbb{R}^n} g(x, v)$. Any function $\sigma : \mathbb{R}^m \to \mathbb{R}_{+\infty}$ which is proper, convex and lower semicontinuous is said to be an *augmenting function* if

$$\sigma \geq 0, \quad \min \sigma = 0, \quad \operatorname{Argmin} \sigma = \{0\}. \tag{2}$$

The *augmented Lagrangian* $\bar{l} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}_+ \to \mathbb{R}_{\pm\infty}$ corresponding to the duality parameterization g and the augmenting function σ is given by

$$\bar{l}(x, u, c) := \inf_{v\in\mathbb{R}^m} \left[g(x, v) + c\sigma(v) - \langle u, v\rangle\right]. \tag{3}$$

The *dual function* induced by the augmented Lagrangian $\bar{l}$ is

$$\widetilde{H}(u, c) := \inf_x \bar{l}(x, u, c). \tag{4}$$

So the (augmented) dual problem becomes

$$(D) \qquad \max_{u\in\mathbb{R}^m,\, c\geq 0} \widetilde{H}(u, c). \tag{5}$$

We say that a duality parameterization $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\pm\infty}$ is *level bounded in x locally uniformly in v* if for each $\bar{v} \in \mathbb{R}^m$ and each $\beta \in \mathbb{R}$, there exists a neighborhood $W \subset \mathbb{R}^m$ of $\bar{v}$ such that for all $w \in W$ we have that

$$\{x \in \mathbb{R}^n \ :\ g(x, w) \leq \beta\} \subset B,$$

where $B \subset \mathbb{R}^n$ is a bounded set.

Theorem 1 *([15, Theorem 11.59]) Consider a duality parameterization $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_{\pm\infty}$ for problem (1) and its associated augmented Lagrangian $\bar{l}$ as in (3). Assume that the following hypotheses hold.*

(i) g is level bounded in x locally uniformly in v.

(ii) $\inf_{x\in\mathbb{R}^n} \bar{l}(x, u, c) > -\infty$ for at least one $(u, c) \in \mathbb{R}^m \times \mathbb{R}_+$.

Then

(a) zero duality holds, i.e., $\inf_x \varphi(x) = \sup_{u,c} \widetilde{H}(u, c)$,

(b) Primal and (augmented) dual solutions (i.e., solutions of (D)) are characterized as saddle points of the augmented Lagrangian:

$$\begin{aligned} &\overline{x} \in \operatorname{Argmin}_x \varphi(x) \quad \text{and} \quad (\overline{u}, \bar{c}) \in \operatorname{Argmax}_{u,c} \widetilde{H}(u, c) \\ \iff \quad &\inf_x \bar{l}(x, \overline{u}, \bar{c}) = \bar{l}(\overline{x}, \overline{u}, \bar{c}) = \sup_{u,c} \bar{l}(\overline{x}, u, c). \end{aligned}$$

(c) The elements of $\operatorname{Argmax}_{u,c} \widetilde{H}(u, c)$ are the pairs $(\overline{u}, \bar{c})$ such that

$$p(v) \geq p(0) + \langle \overline{u}, v\rangle - \overline{c}\sigma(v) \quad \forall\, v.$$

Denote by $X_0 := \{x \in X \ : \ f^+(x) = 0\}$, the constraint set of problem (P). Given a set $A \subset \mathbb{R}^n$, the *indicator function* $\delta_A : \mathbb{R}^n \to \mathbb{R}_{+\infty}$ is defined as $\delta_A(x) = 0$ when $x \in A$ and $\delta_A(x) = +\infty$ otherwise. The duality properties of (P) can be obtained applying Theorem 1 for $\varphi := f_0 + \delta_{X_0}$. We use as augmenting function $\sigma(v) = \|v\|$, so our augmented Lagrangian is the *sharp augmented Lagrangian* (see [15, Example 11.58]). Let $m = 1$ and define the duality parameterization $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}_{+\infty}$ as

$$g(x,v) = \begin{cases} f_0(x), & \text{if } x \in X \text{ and } f^+(x) = v, \\ +\infty, & \text{if } x \notin X \text{ or } f^+(x) \neq v. \end{cases}$$

It is clear that $\varphi = g(\cdot, 0)$. Moreover, since X is compact, it is easy to check that g is is level bounded in x locally uniformly in v. It is also straightforward to verify that the augmented Lagrangian $\bar{l} : \mathbb{R}^n \times \mathbb{R} \times \mathbb{R}_+ \to \mathbb{R}_{\pm\infty}$ associated with these choices of g and σ is

$$\bar{l}(x,u,c) = \begin{cases} f_0(x) + (c-u)f^+(x), & \text{if } x \in X, \\ +\infty, & \text{if } x \notin X. \end{cases} \tag{6}$$

So we have that $\inf_x \bar{l}(x,u,c) = \inf_{x \in X} f_0(x) + (c-u)f^+(x) > -\infty$ for every (u,c) because the right-hand infimum is the minimization of a continuous function over a compact set. Therefore, all hypotheses of Theorem 1 hold and hence the conclusions (a), (b) and (c) of this result hold for our scheme.

By taking $d := c - u \in \mathbb{R}$ the Lagrangian in (6) becomes

$$L(x,d) := \begin{cases} f_0(x) + d\, f^+(x), & \text{if } x \in X, \\ +\infty, & \text{if } x \notin X. \end{cases} \tag{7}$$

Let the solution set and the optimal value of problem (P) be denoted by $S(P)$ and $\overline{M}$, respectively. We typically denote an element of $S(P)$ by $\overline{x}$. The dual function H associated with the Lagrangian in (7) is

$$H(d) := \min_{x \in X} \left[f_0(x) + d\, f^+(x) \right] . \tag{8}$$

This function is concave and upper semicontinuous (because it is the minimum of affine functions of d). Moreover, since X is compact, H is finite everywhere and hence continuous. The dual problem of (P) induced by H is given by

$$(P^*) : \qquad \max_{d \in \mathbb{R}} H(d) \ .$$

The solution set of Problem (P^*) and its optimal value are denoted by $S(P^*)$ and $\overline{H}$, respectively. We denote an element in $S(P^*)$ by $\overline{d}$. For a given $d \in \mathbb{R}$, consider the set

$$X(d) := \operatorname*{Argmin}_{x \in X} \left[f_0(x) + d\, f^+(x) \right] \ . \tag{9}$$

We assume that we are able to solve the minimization problem given in (8). In other words, we are able to find an element of $X(d)$ for every $d \in \mathbb{R}$.

Remark 1 Using (4), (6) and the compactness of X, the dual function becomes

$$\widetilde{H}(u,c) = \min_{x\in X} \left[f_0(x) + (c-u)\, f^+(x)\right], \tag{10}$$

so we clearly have $\widetilde{H}(u,c) = H(c-u)$. Setting $\widetilde{X}(u,c) := \text{Argmin}_{x\in X} \left[f_0(x) + (c-u)\, f^+(x)\right]$ we also get $\widetilde{X}(u,c) = X(c-u)$. As we stated earlier, the sharp Lagrangian yields zero duality gap, which in our case tantamounts to

$$\sup_{(c,u)\in \mathbf{R}_+\times\mathbf{R}} \widetilde{H}(u,c) = \overline{M}\,. \tag{11}$$

This clearly implies

$$\sup_{d\in\mathbf{R}} H(d) = \overline{M}\,. \tag{12}$$

The following is a well-known tool from convex analysis. Given a concave function $H : \mathbb{R}^m \to \mathbb{R}$ and a fixed $d \in \mathbb{R}^m$, the set

$$\partial H(d) := \{v \in \mathbb{R}^m \,:\, H(d') \le H(d) + \langle d' - d, v\rangle,\ \forall\, d' \in \mathbb{R}^m\},$$

is called the *subdifferential* of H at d, and each element of this set is called a *subgradient* of H at d (the terms "superdifferential" and "supergradient", respectively, are also used).

It is easy to check that, when $x \in X(d)$, we have

$$f^+(x) \in \partial H(d). \tag{13}$$

Recall that the one-sided derivatives of a function $H : \mathbb{R} \to \mathbb{R}$ at $d \in \mathbb{R}$ are defined as

$$H'_+(d) := \lim_{\lambda\downarrow 0} \frac{H(d+\lambda) - H(d)}{\lambda}\,, \qquad H'_-(d) := \lim_{\lambda\uparrow 0} \frac{H(d+\lambda) - H(d)}{\lambda}\,. \tag{14}$$

Since H is concave, it is well-known (see [14, page 229]) that the subdifferential and the one-sided derivatives of $H : \mathbb{R} \to \mathbb{R}$ at d are related by the equality

$$\partial H(d) := \{v \in \mathbb{R} \,:\, H'_+(d') \le v \le H'_-(d)\}\,. \tag{15}$$

Even though H as in (8) is in general nonsmooth, its one-sided derivatives are always defined. The result below follows directly from [5, Theorem 1].

Theorem 2 *Let H and $X(d)$ be as in* (8) *and* (9)*, respectively. Then*

$$H'_+(d) = \min_{x\in X(d)} f^+(x)\,, \quad H'_-(d) = \max_{x\in X(d)} f^+(x)\,.$$

The next lemma is a trivial modification of a result appearing in [8]. It will be frequently used in the next section, so we include here its simple proof.

Lemma 1 *Let $d \in \mathbb{R}$ and suppose that $\widetilde{x} \in X(d)$. Then $\widetilde{x} \in S(P)$ and $d \in S(P^*)$ if and only if $f(\widetilde{x}) = 0$.*

Proof. It is enough to prove that $\widetilde{x} \in S(P)$ and $d \in S(P^*)$, since the converse is trivial. Assume that $f(\widetilde{x}) = 0$ and $\widetilde{x} \in X(d)$, then $L(\widetilde{x}, d) = f_0(\widetilde{x}) \geq \overline{M}$. Altogether,

$$\overline{M} \leq f_0(\widetilde{x}) = L(\widetilde{x}, d) = \min_{x \in X} L(x, d) = H(d) \leq \overline{H} \leq \overline{M},$$

where we used weak duality in the last inequality. Thus we have $f_0(\widetilde{x}) = \overline{M} = H(d)$ and hence $\widetilde{x} \in S(P)$ and $d \in S(P^*)$. $\square$

3 Exact Penalty Parameters

We will view the Lagrangian $L(x, d)$ in (7) as an exact penalty function. A parameter d, for which $X(d) = S(P)$, will be called an *exact penalty parameter* (EPP) for Problem (P). Rubinov and Yang [16, Section 3.2.8] call this parameter a *strong exact parameter.* We use the alternative description, "exact penalty parameter," in order to emphasize the connection of the theory and numerical implementation we present in this study with sequential penalty methods.

Define now the sets

$$\begin{aligned} C &:= \{d \in \mathbb{R} \,:\, X(d + \eta) = S(P),\, \forall\, \eta > 0\}\,, \\ D &:= \{d \in \mathbb{R} \,:\, X(d) = S(P)\}\,. \end{aligned} \tag{16}$$

Let $\overline{d}_{\min} \in \mathbb{R}$ be defined as

$$\overline{d}_{\min} := \inf C\,, \tag{17}$$

with the convention $\inf \emptyset = +\infty$. The notation $\overline{d}_{\min}$ is justified as it will be shown in Theorem 3(b) that the infimum in (17) is attained whenever it is finite.

The following result lists the main properties of $\overline{d}_{\min}$ and the relationship between the sets C, D and $S(P^*)$.

Theorem 3 *Let C, D be given as in* (16)*, and $\overline{d}_{\min}$ be defined as in* (17)*. Then,*

(a) $(\overline{d}_{\min}, +\infty) \subset D \subset C = S(P^*)$. Hence $\overline{d}_{\min} < +\infty$ if and only if $S(P^*) \neq \emptyset$.

(b) $\overline{d}_{\min} > -\infty$ if and only if $\exists\, \widetilde{x} \in X$ with $f^+(\widetilde{x}) > 0$. If $\overline{d}_{\min}$ is finite, we have

$$C = [\overline{d}_{\min}, +\infty)\,,$$

so in this case the infimum in (17) *is attained,*

(c) Assume that $\overline{d}_{\min}$ is finite. Then $X(\overline{d}_{\min}) \supset S(P)$.

(d) Assume that $\overline{d}_{\min}$ is finite. If $d_k \uparrow \overline{d}_{\min}$ and $x_k \in X(d_k)$, then every accumulation point of $\{x_k\}$ belongs to $X(\overline{d}_{\min})$.

Proof. (a) For proving the inclusion $D \supset (\overline{d}_{\min}, +\infty)$ let us take $d > \overline{d}_{\min}$. Then there exists $d' \in C$ such that $\overline{d}_{\min} \le d' < d$. Take $\eta := d - d' > 0$. The definitions of d' and η yield $S(P) = X(d' + \eta) = X(d)$ so $d \in D$. Let us prove now the inclusion $D \subset C$. Take $d \in D$ and $\eta > 0$. We must prove that $S(P) = X(d + \eta)$.

First, we show that $S(P) \subset X(d + \eta)$. Fix $\overline{x} \in S(P)$ and suppose $\overline{x} \notin X(d + \eta)$. This means that there exists $x' \in X$ such that

$$f_0(x') + (d + \eta)\, f^+(x') < f_0(\overline{x}) + (d + \eta)\, f^+(\overline{x}) = f_0(\overline{x})\,, \tag{18}$$

where we used $f^+(\overline{x}) = 0$. It follows that $f^+(x') > 0$ (if $f^+(x') = 0$ then x' is feasible and $f_0(x') < f_0(\overline{x})$, which is a contradiction). Now we can write

$$\overline{M} = f_0(\overline{x}) = H(d) \le f_0(x') + d\, f^+(x') < f_0(x') + (d + \eta)\, f^+(x') < f_0(\overline{x}) = \overline{M}\,,$$

a contradiction, so $\overline{x} \in X(d + \eta)$.

Next, we show that $X(d + \eta) \subset S(P)$. Fix $\widetilde{x} \in X(d + \eta)$. We have

$$f_0(\widetilde{x}) + (d + \eta)\, f^+(\widetilde{x}) \le f_0(\overline{x})\,. \tag{19}$$

By Lemma 1, it is enough to show that $f^+(\widetilde{x}) = 0$ in order to conclude that $\overline{x} \in S(P)$. Indeed, the assumption $f^+(\widetilde{x}) > 0$ leads to a contradiction:

$$\overline{M} \ge H(d + \eta) = f_0(\widetilde{x}) + (d + \eta)\, f^+(\widetilde{x}) > f_0(\widetilde{x}) + d\, f^+(\widetilde{x}) \ge H(d)\,, \tag{20}$$

contradicting the fact that $d \in D$ (which entails $H(d) = \overline{M}$). So the inclusion $X(d+\eta) \subset S(P)$ also holds. As a result $X(d+\eta) = S(P)$ and thus the inclusion $D \subset C$ is established.

Next we prove that $C = S(P^*)$. The proof of the inclusion $S(P^*) \subset C$ follows the same steps as those in the proof of $D \subset C$ above, starting instead by taking $\overline{d} \in S(P^*)$ and concluding with $\overline{d} \in C$. More precisely, the proof of $D \subset C$ only uses $H(d) = \overline{M}$, which also holds when $d \in S(P^*)$. Therefore, we only need to prove that $C \subset S(P^*)$.

Take $d \in C$ and suppose that $d \notin S(P^*)$. Then $H(d) < f_0(\overline{x})$ for every $\overline{x} \in S(P)$. Because H is upper semicontinuous there exists $\eta > 0$ for which $H(d + \eta) < f_0(\overline{x})$, contradicting the fact that $d \in C$. Hence we must have $d \in S(P^*)$.

Let us now prove (b). Note that, if $f^+(x) = 0$ for every $x \in X$, then $H(d) = \overline{M}$ and $X(d) = S(P)$ for every $d \in \mathbb{R}$, which yields $\overline{d}_{\min} = -\infty$. Conversely, assume that $\overline{d}_{\min} = -\infty$ and suppose there exists $x' \in X$ with $f^+(x') > 0$. The assumption on $\overline{d}_{\min}$ implies that we can take a sequence $d_k \downarrow -\infty$ with $X(d_k) = S(P)$. We can write for every $\overline{x} \in S(P)$,

$$-\infty < f_0(\overline{x}) = H(d_k) = \min_{x \in X} f_0(x) + d_k\, f^+(x) \le f_0(x') + d_k\, f^+(x'),$$

and the assumption on x' implies that the right hand side of the expression above tends to $-\infty$, a contradiction. So we must have $f^+(x) = 0$ for every $x \in X$. This proves the first sentence in (b). Let us prove now that $C = [\overline{d}_{\min}, \infty)$ when $\overline{d}_{\min} \in \mathbb{R}$. From item (a) and the definitions of C and $\overline{d}_{\min}$ it is clear that $[\overline{d}_{\min}, \infty) \supset C \supset (\overline{d}_{\min}, \infty)$. So it is enough to show that $\overline{d}_{\min} \in C$. Fix $\eta > 0$, choose any $\eta', \eta'' > 0$ such that

$\eta' + \eta'' = \eta$ and write $\overline{d}_{\min} + \eta = (\overline{d}_{\min} + \eta') + \eta''$. Note that $d' := \overline{d}_{\min} + \eta' \in C$ so $S(P) = X(d' + \eta'') = X(\overline{d}_{\min} + \eta)$, as we wanted.

To prove (c), let $d_k \downarrow \overline{d}_{\min}$. By (b), $d_k \in C$. Using the upper semicontinuity of H and the fact that $d_k \in C = S(P^*)$ we have that $\overline{M} \geq H(\overline{d}_{\min}) \geq \limsup_{k\to\infty} H(d_k) = \overline{M}$, which yields $\overline{M} = H(\overline{d}_{\min})$ and hence $S(P) \subset X(\overline{d}_{\min})$.

Part (d) is a consequence of [15, Theorem 1.17(b)] and the continuity of H. □

Remark 2 *If $\overline{d}_{\min}$ is finite and $d_k \downarrow \overline{d}_{\min}$, then every $x_k \in X(d_k)$ belongs to $S(P)$.*

A direct consequence of Theorem 3 follows below.

Corollary 1 *The set of dual solutions $S(P^*)$ is nonempty if and only if the set D of exact penalty parameters is an interval which is unbounded above.*

Proof. If $S(P^*)$ is nonempty, then by part (a) in Theorem 3 we have that $\overline{d}_{\min} < +\infty$ and because $(\overline{d}_{\min}, \infty) \subset D \subset [\overline{d}_{\min}, \infty)$ the conclusion holds. Conversely, if D is an interval which is unbounded above, then in particular it is not empty. Therefore $\emptyset \subsetneq D \subset C = S(P^*)$. Thus $S(P^*)$ is nonempty. □

Note that $\overline{d}_{\min}$ may not be an exact penalty parameter, because Theorem 3(c) only gives the inclusion $X(\overline{d}_{\min}) \supset S(P)$. The following simple example shows that this inclusion may be strict.

Example 1 Consider the problem

$$(P) \qquad \text{minimize } |x| \text{ over all } x \text{ in } [-1,1] \text{ satisfying } f^+(x) = \max\{0, x\} = 0,$$

so $S(P) = \{0\}$ with $\overline{M} = 0$. Moreover, it can be easily checked that

$$H(d) = \begin{cases} d+1 & d \leq -1, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore,

$$X(d) = \begin{cases} \{1\} & d < -1, \\ [0,1] & d = -1, \\ \{0\} & d > -1. \end{cases}$$

One has $\overline{d}_{\min} = -1$ and clearly $X(\overline{d}_{\min}) \supsetneq S(P)$.

The above example suggests that $X(\overline{d}_{\min}) \supsetneq S(P)$ whenever H is not differentiable at $\overline{d}_{\min}$. This fact is proved next.

Theorem 4 *Let H be as in* (8) *and $\overline{d}_{\min}$ given by* (17)*. Suppose that $\overline{d}_{\min}$ is finite. The following statements are equivalent.*

(a) H is differentiable at $\overline{d}_{\min}$.

(b) $X(\overline{d}_{\min}) = S(P)$, *i.e.* $\overline{d}_{\min}$ *is an EPP.*

(c) If $d_k \uparrow \overline{d}_{\min}$ *and* $x_k \in X(d_k)$ *for all* k*, then every accumulation point of* $\{x_k\}$ *belongs to* $S(P)$*.*

If either case holds, then $\nabla H(\overline{d}_{\min}) = 0$*.*

Proof. Assume (a) holds. By Theorem 3(a) we have that $H(d) = \overline{M}$ for every $d > \overline{d}_{\min}$. So H is constant on $(\overline{d}_{\min}, \infty)$ and hence $\partial H(d) = \{0\}$ for every $d > \overline{d}_{\min}$. Since the graph of the multifunction $\partial H(\cdot)$ is closed (see Theorem 24.4 in [14]), we must have $0 \in \partial H(\overline{d}_{\min})$. Now using (a), we get

$$0 = \nabla H(\overline{d}_{\min}) = \partial H(\overline{d}_{\min}). \tag{21}$$

On the other hand, by (13), $f^+(\widetilde{x}) \in \partial H(\overline{d}_{\min})$ for every $\widetilde{x} \in X(\overline{d}_{\min})$. Combining this with (21) we get $f^+(\widetilde{x}) = 0$ for every $\widetilde{x} \in X(\overline{d}_{\min})$. By Lemma 1 we conclude that $\widetilde{x} \in S(P)$. So $X(\overline{d}_{\min}) \subset S(P)$. Now (b) follows from Theorem 3(c).

In order to prove that (b) implies (c), take a sequence $\{x_k\}$ such that $x_k \in X(d_k)$ for all k. By Theorem 3(d) and part (b) of this theorem, (c) readily follows.

Now we prove that (c) implies (a). Assume that (a) is not true, that is H is not differentiable at $\overline{d}_{\min}$. By Theorem 2 and (15) this yields the existence of some $\alpha > 0$ such that $\alpha \in \partial H(\overline{d}_{\min})$. There are two cases to consider:

Case (i): There exists a subsequence $\{d_{k_j}\}$ of $\{d_k\}$ such that there exists $a_{k_j} \in \partial H(d_{k_j})$ with $a_{k_j} \downarrow 0$. By antimonotonicity of ∂H we have

$$0 \geq (d_{k_j} - \overline{d}_{\min})\,(a_{k_j} - \alpha). \tag{22}$$

By the assumption on $\{a_{k_j}\}$ there exists a j_0 such that for all $j > j_0$ we must have $a_{k_j} < \alpha$. So (22) can be re-written for $j > j_0$ as

$$0 \leq (\overline{d}_{\min} - d_{k_j})\,(a_{k_j} - \alpha) < 0,$$

where we also used in the last inequality the fact that $\overline{d}_{\min} > d_k$ for all k. The above expression constitutes a contradiction and hence we are left with the only other case, stated below.

Case (ii): There exists $\beta > 0$ such that

$$\inf_{k\in\mathbb{N}} \inf_{v\in\partial H(d_k)} v \geq \beta > 0.$$

Using the expression above and (13) we get $f^+(x_k) \geq \beta$ for all k. The continuity of f^+ now yields $f^+(\widetilde{x}) \geq \beta > 0$ for every accumulation point of $\{x_k\}$. But this is in contradiction with the fact that assumption (c) gives $\widetilde{x} \in S(P)$ and therefore $f^+(\widetilde{x}) = 0$. Hence H must be differentiable at $\overline{d}_{\min}$. The proof of the equivalences is complete. The last sentence of the theorem follows from (21). □

When the parameter $\overline{d}_{\min}$ is an EPP, it is referred to as the *least exact penalty parameter* (LEPP) for Problem (P).

4 Dual Penalty Update

The modified subgradient (MSG) algorithm was introduced to tackle nonconvex, nonsmooth optimization problems subject to equality constraints, by solving the dual problem. It uses an epsilon-subgradient search direction in order to (strictly) improve the value of the dual function H. Given (u_k, c_k), a current iterate, the MSG algorithm, as given in [3, 8, 9], obtains the next iterate by the rule

$$u_{k+1} := u_k - s_k f^+(x_k) \ , \tag{23}$$
$$c_{k+1} := c_k + (s_k + \varepsilon_k) f^+(x_k) \ , \tag{24}$$

where $x_k \in \widetilde{X}(u_k, c_k)$ and the step-size parameter $s_k > 0$. The parameter ε_k is restricted as $0 < \varepsilon_k < s_k$ in [8, 9], while it is more relaxed as $\varepsilon_k > 0$ in [3]. We will use the setting and the results in [3], where s_k is chosen in a more general way, and convergence of the whole sequence of dual variables $\{(u_k, c_k)\}$ is established. It is further proved in [3] that all accumulation points of an auxiliary sequence of primal variables $\{\widetilde{x}_k\}$, such that $\widetilde{x}_k \in \widetilde{X}(u_k, c_k + \eta)$ for all k and for every fixed $\eta > 0$, are optimal.

In the present paper's setting, $X(d_k+\eta) = \widetilde{X}(u_k, c_k+\eta)$, with $d_k = c_k - u_k$. Theorem 4(c) improves the "auxiliary" primal convergence result in [3] in the following sense: Every accumulation point of the primal sequence $\{x_k\}$ such that $x_k \in X(d_k)$ and $d_k \uparrow \overline{d}_{\min}$, is an optimal solution when H is differentiable at $\overline{d}_{\min}$. From a numerical point of view, it is important to avoid the need for the $\eta > 0$ increment as an instrument for achieving convergence. Indeed, for large enough k we would have $d_k + \eta > \overline{d}_{\min}$, and this "stepping over the penalty threshold" may cause numerical instabilities and introduce inaccuracies in the solution of the subproblems. This unwanted situation will be witnessed particularly in Problem 2 in Section 5.

Letting $\varepsilon_k := \alpha s_k$ for a fixed $\alpha > 0$, and using (23)-(24), we obtain the following update for $d_k = c_k - u_k$:

$$d_{k+1} = d_k + (2+\alpha)\, s_k\, f^+(x_k) \ , \tag{25}$$

where $x_k \in X(d_k)$. The step-size s_k is chosen in [3, Section 4.2] as

$$s_k = \delta\, \frac{\overline{H} - H(d_k)}{[f^+(x_k)]^2} \ , \tag{26}$$

where $0 < \delta < 2$. It should be noted that $H(d_k) \le \overline{H}$ for every k, and so one always has $s_k > 0$. Combining (25) and (26) the update rule simplifies to

$$d_{k+1} = d_k + \beta\, \frac{\overline{H} - H(d_k)}{f^+(x_k)} \tag{27}$$

where $\beta > 0$. As the solution of the unconstrained problems approach the solution of the original problem, $f^+(x_k)$ becomes very small and so the update in (27) often yields very large values of d_{k+1}, resulting in numerical instabilities in obtaining the solution of the next unconstrained subproblem. To alleviate this unwanted behaviour, we consider altering the subgradient $f^+(x_k)$ as $f^+(x_k) + \varepsilon$, where $\varepsilon > 0$ is small.

One should observe from (27) that as d_k tends to a solution $\overline{d}$, $\overline{H} - H(d_k) \to 0$ and $f^+(x_k) \to 0$. If an EPP exists, then the limit of the quotient in (27) is finite, and so the numerator $\overline{H} - H(d_k)$ tends to 0 "more rapidly" than $f^+(x_k)$. This justifies the replacement of $f^+(x_k)$ by $f^+(x_k) + \varepsilon$, because the convergence of d_k to $\overline{d}$ can still be achieved, albeit possibly more slowly near a solution. This is a price worth to pay, given the numerically crippling effect of a vanishing $f^+(x_k)$. On the other hand, if ε is chosen reasonably small, then rate of convergence should not be affected much near a solution.

When the subgradient $f^+(x_k)$ in (25) and (26) is replaced by $f^+(x_k) + \varepsilon$, the update formula in (27) simply becomes

$$d_{k+1} = d_k + \beta\,\frac{\overline{H} - H(d_k)}{f^+(x_k) + \varepsilon}\ . \tag{28}$$

We will refer to this formula as the *dual penalty update* (**DPU**) rule, because it is derived from a duality scheme. The DPU rule requires the knowledge of the optimal value $\overline{H}$, which is at hand only in some special cases, for example when the problem of solving a nonlinear system of equations is reformulated as a minimization problem [7]. Choosing the unknown optimal value $\overline{H}$ in (28) has been an issue in subgradient methods. In [1], $\overline{H}$ is chosen as a convex combination of a fixed upper bound and the current best dual value; [19] proposes the so-called variable target value method which assumes no a priori knowledge regarding $\overline{H}$. As in [3], we will use an upper bound estimate, denoted by $\widehat{H}$, for $\overline{H}$. In many problems, $\widehat{H}$ can be established easily: the value of the cost f_0 at a feasible point constitutes an upper bound for the optimal value.

We now propose an algorithm for the Sequential Penalty Function (SPF) method using the DPU rule for solving Problem (P).

SPF Method with DPU Rule

Step 0 Choose $d_0 \in \mathbb{R}$. Set $k = 0$.

Step *k* Given d_k:

Step *k*.1 Find $x_k \in X(d_k)$, i.e.

$$L(x_k, d_k) = \min_{x\in X}\left[f_0(x) + d_k\, f^+(x)\right]\ . \tag{29}$$

If $f^+(x_k) = 0$, then STOP: $d_k \in S(P^*)$ and $x_k \in S(P)$.

Step *k*.2 Perform DPU:

$$d_{k+1} = d_k + \beta\,\frac{\widehat{H} - H(d_k)}{f^+(x_k) + \varepsilon}\ . \tag{30}$$

where $\widehat{H}$ is an upper bound estimate of $\overline{H}$. Set $k = k + 1$ and repeat Step k.

In the classical SPF method, traditionally the value of d is increased by a constant scalar multiple, namely

$$d_{k+1} = \gamma\, d_k \tag{31}$$

where $\gamma > 1$, with $d_0 > 0$, instead of (30). We will refer to (31) as the *classical penalty update* (**CPU**) rule. So the SPF method with the CPU rule is performed through the same algorithm given for the SPF method with the DPU rule above by simply replacing (30) by (31).

In the numerical experiments we carry out in the next section, we will implement the SPF method with both the DPU and CPU rules, for comparison purposes.

5 Numerical Experiments

To illustrate the use and advantages of the DPU rule and compare it with the CPU rule, we have chosen two test problems from the literature. For the solution of the subproblems, the MATLAB function m-file `fminsearch` has been utilized. In both test problems, the termination tolerance on the function value (TolFun) and the termination tolerance on the optimization variable (TolX) for `fminsearch` have been chosen as 10^{-10}.

Problem 1 Consider the test problem 62 (GLR-P1-1) from [11].

$$\left\{\begin{array}{l} \min\ f_0(x) = -32.174\left[255\ln\left(\dfrac{x_1+x_2+x_3+.03}{0.09x_1+x_2+x_3+.03}\right)\right. \\ \qquad\qquad + 280\ln\left(\dfrac{x_2+x_3+.03}{0.07x_2+x_3+.03}\right) \\ \qquad\qquad \left. + 290\ln\left(\dfrac{x_3+.03}{0.13x_3+.03}\right)\right] \\ \text{subject to}\ \ x_1+x_2+x_3=1 \\ \qquad\qquad 0 \le x_i \le 1\ , \quad i=1,2,3. \end{array}\right.$$

The constraints in this problem can be transformed into a single constraint as follows.

$$f^+(x) = \max\left(0,\ \max\left(\max\left(\max_i x_i - 1,\ \max_i -x_i\right),\ |x_1+x_2+x_3-1|\right)\right) = 0\ .$$

Then, the problem, together with this single constraint, takes the same form as Problem (P).

For the DPU rule in (30), we use an upperbound for the estimate of $\overline{H}$, namely we set $\widehat{H} = -20000$. We also set $\beta = 2$ and $\varepsilon = 10^{-4}$. A solution to the problem is obtained in four iterations, which are listed in Table 1 and depicted in Figure 1. In Figure 1, we also provide a graph of the dual function H, which has been generated by setting $\beta = 0.001$. By using the same β, we also found that the least exact penalty parameter for this problem is $\overline{d}_{\min} = 6387$ (approximated to four digits of accuracy).

Using the DPU rule, this time with $\beta = 1$, the same solution is obtained in five iterations. With $\beta = 0.5$, seven iterations are needed. As expected, the smaller the β is, the more iterations one would need to get a solution.

Using the CPU rule in (31) with $\gamma = 2$, the same solution is obtained in 12 iterations, which are listed in Table 2 and depicted in Figure 1. With a less gradual increase in d_k, i.e. with a larger γ, one would expect to get a solution in fewer iterations. So we set $\gamma = 10$, with which the algorithm terminates in five iterations (as in the case of the DPU rule with $\beta = 1$). However the last iterate, as seen in Table 3, is not a solution of the problem. The reason is simple: with a larger γ, because the increase in d_k is linear, d_k becomes too large in Step 4 (namely $d_4 = 50000$), causing numerical instability, resulting in premature termination, yielding a feasible but a non-optimal solution.

In this particular problem, numerical instability in the last step seems to depend on how far d_k falls from $\overline{d}_{\min} = 6387$. If the CPU rule is used again, this time with $\gamma = 5$, then the previous solution is obtained accurately in six iterations. In this case $d_5 = 15625$, which is relatively closer to $\overline{d}_{\min}$, compared to $d_4 = 50000$ in the case of $\gamma = 10$.

Loosely speaking, the DPU rule takes into account the change in the dual function relative to a reference upper bound $\widehat{H}$. When one approaches the solution the increase in d_k is adjusted by the update in such a way that it will not be too far from the least penalty parameter $\overline{d}_{\min}$.

A beneficial use of the DPU rule seems to depend on whether the upper bound $\widehat{H}$ is relatively close to the optimal dual value $\overline{H}$ or not. For example, if one takes $\widehat{H} = 0$, which is much farther from $\overline{H}$, with $\beta = 2$, numerical instability occurs; although, with $\beta = 1$, the ill-behaviour disappears, and the accurate solution is obtained in just four iterations.

In summary, the SPF method with DPU rule can achieve the solution in just a few iterations, if the two important parameters $\widehat{H}$ and β are chosen appropriately. On the other hand, the SPF method with CPU rule seems to require a larger number of iterations.

To examine the differentiability of the dual function $H(d)$ we have generated the graph of the derivative $H'(d)$ in Figure 2. We have used a forward difference approximation for the one-sided derivative $H'_+(d)$, using $d = d_k$ and $\lambda = d_{k+1} - d_k$ in (14), to plot the graph. Note that we have set β small enough to make the approximation accurate. It turns out that the graph of $H'_-(d)$, using $d = d_{k+1}$ and $\lambda = d_k - d_{k+1}$ in (14) (which amounts to a backward difference approximation), overlaps with the graph of $H'_+(d)$, pointing to the smoothness of $H'(d)$. This is independently confirmed by observing that the graph of $f^+(x_k)$ vs d overlaps with the graph displayed in Figure 2 (see Theorem 2).

Because Figure 2 points to differentiability of H, Theorem 4 ensures that $\overline{d}_{\min}$ is LEPP and any accumulation point of the sequence $\{x_k\}$ generated by the SPF is a solution.

k	x_1^k	x_2^k	x_3^k	d_k	$f^+(x_k)$	$f_0(x_k)$	$H(d_k)$
-1	0.00000000	0.00000000	0.00000000		1.0	0.00000	
0	466.93779342	19.79014883	0.53807329	5	486	-54192.98314	-51761.65306
1	28.21749161	3.20127900	0.20554160	136	31	-45085.86994	-40932.14563
2	2.93940741	0.78764517	0.09302594	1503	2.8	-34440.26759	-30202.65302
3	0.61781268	0.32820223	0.05398509	8738	1.1×10^{-16}	-26272.51449	-26272.51449

Table 1: Problem 1 – Iterations with the DPU rule, where $\beta = 2$, $\widehat{H} = -20000$, and $\varepsilon = 10^{-4}$.

k	x_1^k	x_2^k	x_3^k	d_k	$f^+(x_k)$	$f_0(x_k)$	$H(d_k)$
-1	0.00000000	0.00000000	0.00000000		1.0	0.00000	
0	466.93779342	19.79014883	0.53807329	5	486	-54192.98314	-51761.65306
1	263.88014395	13.61362375	0.44314887	10	277	-52730.11426	-49960.74510
2	147.85425931	9.32625393	0.36363958	20	157	-51047.92096	-47917.03791
3	82.02680411	6.36018952	0.29712567	40	88	-49124.27593	-45616.91116
4	44.98641604	4.31612736	0.24157137	80	49	-46938.31321	-43054.78403
5	24.34249650	2.91362423	0.19526220	160	27	-44471.60516	-40239.38390
6	12.96337776	1.95602902	0.15675260	320	14	-41709.42909	-37205.05809
7	6.77077307	1.30572890	0.12482161	640	7.2	-38641.98628	-34033.13918
8	3.45040606	0.86668240	0.09843636	1280	3.4	-35265.40767	-30893.53590
9	1.70091898	0.57208938	0.07672185	2560	1.4	-31582.37580	-28127.06646
10	0.79828731	0.37568803	0.05893657	5120	2.3×10^{-1}	-27602.23232	-26409.72331
11	0.61781270	0.32820221	0.05398509	10240	1.1×10^{-16}	-26272.51449	-26272.51449

Table 2: Problem 1 – Iterations with the CPU rule, where $\gamma = 2$.

k	x_1^k	x_2^k	x_3^k	d_k	$f^+(x_k)$	$f_0(x_k)$	$H(d_k)$
-1	0.00000000	0.00000000	0.00000000		1.00	0.00000	
0	466.93779342	19.79014883	0.53807329	5	486	-54192.98314	-51761.65306
1	67.69479504	5.61702253	0.27813340	50	72.6	-48450.13111	-44820.63356
2	8.55433209	1.50885849	0.13550789	500	9.20	-39769.81487	-35170.46563
3	0.82011791	0.38116296	0.05948727	5000	2.6×10^{-1}	-27743.17225	-26439.33156
4	0.54473298	0.39957173	0.05569529	50000	0	-26236.76865	-26236.76865

Table 3: Problem 1 – Iterations with the CPU rule, where $\gamma = 10$.

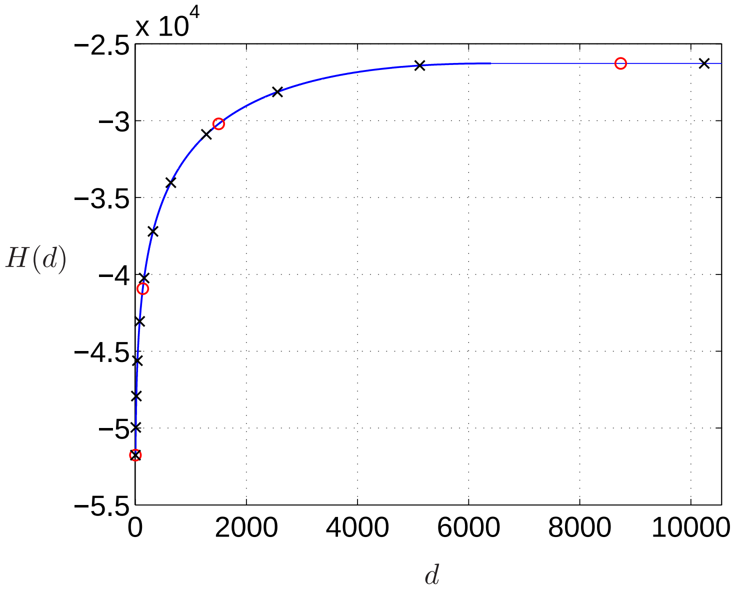

Figure 1: Problem 1 – Graph of the dual function H. Iterations with the DPU rule are shown by $\circ$, and those with the CPU rule are shown by $\times$.

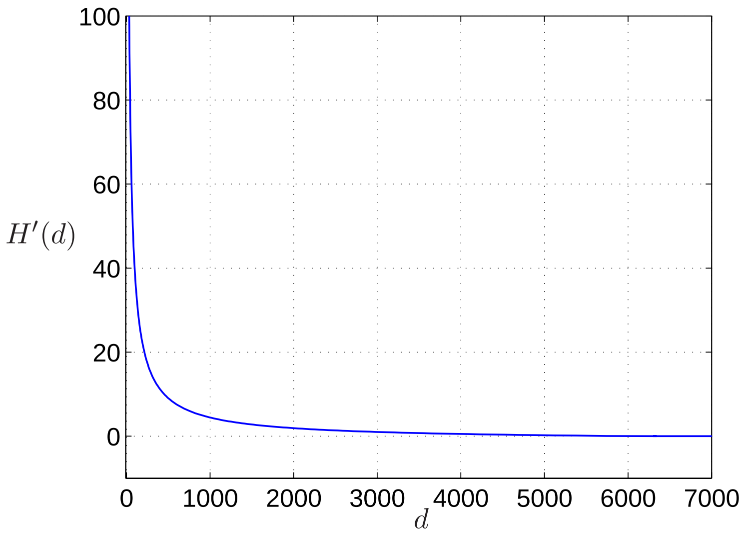

Figure 2: Problem 1 – Derivative of the dual function H.

Problem 2 We consider the test problem 77 (PGR-P1-3) from [11].

$$\begin{cases} \min\ f_0(x) = (x_1-1)^2 + (x_1-x_2)^2 + (x_3-1)^2 + (x_4-1)^4 + (x_5-1)^6 \\ \text{subject to } g_1(x) = x_1^2 x_4 + \sin(x_4 - x_5) - 2\sqrt{2} = 0 \\ g_2(x) = x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0\ . \end{cases}$$

The constraints in this problem can be transformed into a single constraint through

$$f^+(x) = \max\left(|g_1(x)|, |g_2(x)|\right) = 0\ .$$

Now the problem is of the same form as Problem (P). The solution of this problem is listed in [11] as

$$\begin{aligned} \overline{x} &= (1.166172,\ 1.182111,\ 1.380257,\ 1.506036,\ 0.6109203) \\ f_0(\overline{x}) &= 0.24150513 \\ r(\overline{x}) &= 1.2 \times 10^{-10} \end{aligned} \tag{32}$$

where $r(\overline{x}) = |g_1(\overline{x})| + |g_2(\overline{x})|$.

Iterations with the DPU rule are listed in Table 4 and graphically depicted in Figure 3. The tabulated numbers are approximations of the values obtained in the iterations. We start the iterations with a negative penalty parameter, which eventually converge to the solution. Such a start, i.e. $d_0 < 0$, may be necessary in situations where finding a solution to the subproblem with $d_0 > 0$ happens to be difficult. With $d_0 < 0$, the SPF method with CPU rule is not applicable.

The solution point $\overline{x}$ given in Table 4 is accurate only to two significant figures. This inaccurate solution is due to the numerical instability induced by the relatively large value of $d_9 = 0.826$. It turns out that the LEPP is $\overline{d}_{\min} = 0.1174184$. Any $\overline{d} > \overline{d}_{\min}$ is an exact penalty parameter and so should yield the solution. However, even values significantly smaller than $\overline{d} = 0.826$ cause numerical instability and yield an inaccurate solution. For example even $\overline{d} = 0.3$ results in

$$\begin{aligned} \overline{x} &= (1.2,\ 1.1,\ 1.4,\ 1.5,\ 0.67) \\ f_0(\overline{x}) &= 0.24581 \\ f^+(\overline{x}) &= 1.1 \times 10^{-15} \end{aligned}$$

where the second and fifth components of $\overline{x}$ do not agree with those of the solution in Table 4 (namely, $1.1 \neq 1.2$ and $0.67 \neq 0.61$). Needless to say, accuracy diminishes even more drastically with relatively larger values of $\overline{d}$.

An accurate solution of the problem can possibly be obtained by coming arbitrarily close to $\overline{d}_{\min}$. However, first of all, in general it is not possible to know $\overline{d}_{\min}$ itself accurately. For approaching $\overline{d}_{\min}$ one might use the SPF method with an update rule, for increasing the value of d_k up to $\overline{d}_{\min}$. Suppose that we can get arbitrarily close to $\overline{d}_{\min}$ from below. Then a second issue arises as to whether the sequence of iterates $\{x_k\}$ would necessarily converge to a solution or not. This issue is addressed by Theorem 4(c): Differentiability of

H at $\overline{d}_{\min}$ guarantees that every accumulation point of $\{x_k\}$ is a solution. So for problems of the kind we have here, graphing $H'(d)$ might be a useful exercise.

We have plotted the graph of the derivative of H in Figure 4 in the same way we did in the previous problem. We observe from the figure that along the interval where we carry out the iterations, H seems to be nonsmooth only at two points, namely at $d = 0$ and $d = d' \approx 0.035$; H is differentiable elsewhere, including $\overline{d}_{\min}$. By Theorem 2, there are more than one minimizer $\overline{x}$ of the Lagrangian (or the penalty function) $L(x, d)$ at $d = 0$ and $d = d'$. However this is not the contentious issue here. It is much more important that, because H is differentiable at $\overline{d}_{\min}$, any accumulation point of the SPF sequence $\{x_k\}$ is a solution. This guarantees that by getting closer to $\overline{d}_{\min}$ from below, we can obtain a more accurate solution, which we will do next.

The solution tabulated in Table 4 can be refined by using the DPU rule with a smaller constant β, and a more accurate estimate of $\overline{H}$ (again from Table 4). With $d_0 = 0.1$, $\beta = 0.01$ and $\widehat{H} = 0.25$, a solution for $\overline{x}$ accurate to three digits after the decimal point is obtained in 18 iterations. The cost found is accurate to five digits after the decimal point. If $\beta = 0.001$ is used, then the cost is found with the same accuracy as in (32), and $\overline{x}$ is obtained accurately to five digits after the decimal point, in 155 iterations.

Because $d_0 = 0.1 > 0$, one could also choose to use the CPU rule. With $\gamma = 1.001$ the accuracy we could obtain in $\overline{x}$ was three digits after the decimal point, in 161 iterations. With $\gamma = 1.00005$ the accuracy in $\overline{x}$ was improved to four digits after the decimal point, in more than 3000 iterations. The accuracy that can be achieved with the CPU rule in more than 3000 iterations is still worse than that can be achieved with the DPU rule in less than 200 iterations.

k	x_1^k	x_2^k	x_3^k	x_4^k	x_5^k	d_k	$f^+(x_k)$	$f_0(x_k)$	$H(d_k)$
-1	1.0000	1.0000	1.0000	1.0000	1.0000		7.41	0.00000	
0	0.7500	0.5000	0.8095	0.5993	1.0029	-0.500	8.8	0.18707	-4.19292
1	0.8782	0.7564	0.8675	0.6456	0.9975	-0.244	8.4	0.06300	-1.98818
2	0.9461	0.8923	0.9181	0.7007	0.9976	-0.108	8.2	0.02053	-0.85979
3	0.9816	0.9632	0.9612	0.7705	0.9978	-0.037	7.9	0.00495	-0.28706
4	1.0001	1.0002	1.0004	1.0464	0.9994	0.000	7.3	0.00000	0.00140
5	1.0103	1.0206	1.0833	1.2616	0.9981	0.021	6.2	0.01184	0.13955
6	1.0168	1.0335	1.2539	1.3858	0.9983	0.034	3.6	0.08720	0.20901
7	1.0351	1.0522	1.3824	1.4580	0.7279	0.046	6.0×10^{-1}	0.19215	0.21974
8	1.1569	1.1729	1.3804	1.5029	0.6145	0.113	4.1×10^{-2}	0.23678	0.24141
9	1.1676	1.1845	1.3809	1.5044	0.6139	0.826	6.7×10^{-16}	0.24152	0.24152

Table 4: Problem 2 – Iterations with the DPU rule, where $\beta = 0.5$, $\widehat{H} = 0.3$, and $\varepsilon = 10^{-4}$.

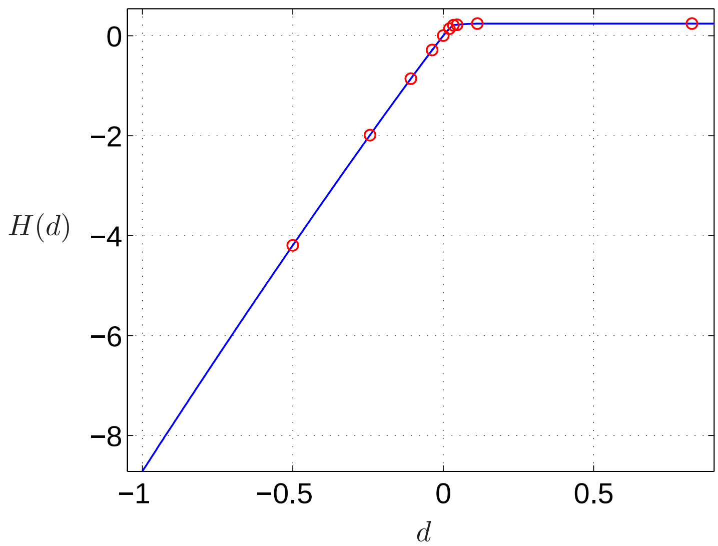

Figure 3: Problem 2 – Graph of the dual function H. Iterations with the DPU rule are shown by ○.

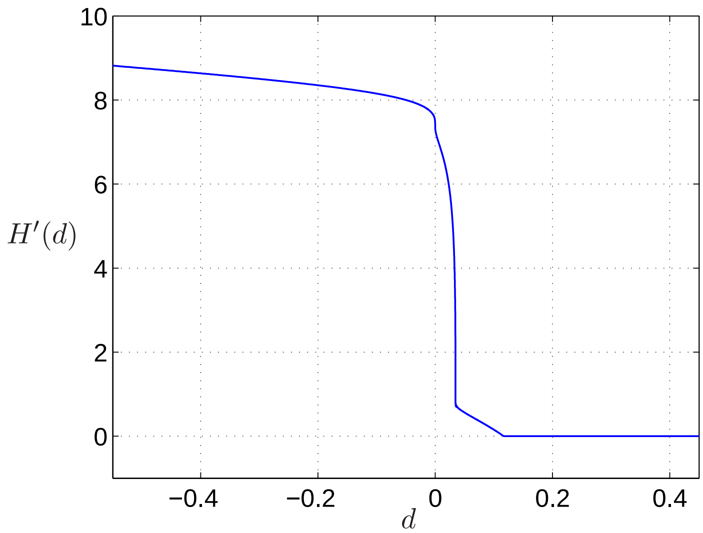

Figure 4: Problem 2 – Derivative of the dual function H.

6 Conclusion

Choosing a small enough exact penalty parameter so that numerical ill-conditioning can be avoided is an important issue in nonlinear programming. It is ideal to choose a parameter "close" to the infimum $\overline{d}_{\min}$ of all exact penalty parameters. However, because in general $\overline{d}_{\min}$ is not known, the value of the penalty parameter is increased gradually from below. We have proposed the so-called dual penalty update rule and showed through numerical experiments that our rule may have clear advantages over the use of the classical update rule.

Our setting allows nonpositive exact penalty parameters. We have established that the dual function is differentiable at $\overline{d}_{\min}$ if and only if $\overline{d}_{\min}$ is an exact penalty parameter. Under either of these conditions, we proved that all the accumulation points of the sequence generated by a sequential penalty function method are solutions of the original problem. To the best of authors' knowledge, this convergence result is not available elsewhere for nonsmooth and nonconvex problems. We have illustrated our theoretical results on two numerical test problems.

Acknowledgments

We are grateful to late Alex Rubinov, and Adil Bagirov, for motivating us to carry out this study.

References

[1] M. S. Bazaraa, and H. D. Sherali (1981). On the choice of step sizes in subgradient optimization. *European Journal of Operations Research*, **7**, 380–388.

[2] D. Boukari, and A. V. Fiacco (1995). Survey of penalty, exact penalty and multiplier methods from 1968 to 1993. *Optimization*, **32**, 301–334.

[3] R. S. Burachik, R. N. Gasimov, N. A. Ismayilova, and C. Y. Kaya (2006). On a modified subgradient algorithm for dual problems via sharp augmented Lagrangian. *Journal of Global Optimization*, **34**(1), 55–78.

[4] J. V. Burke (1991). An exact penalization viewpoint of constrained optimization. *SIAM Journal on Control and Optimization*, **29**(4), 968–998.

[5] J. M. Danskin (1966). The theory of max-min with applications. *SIAM Journal on Applied Mathematics*, **14**(4), 641–664.

[6] G. di Pillo (1994). Exact Penalty Methods. In: Algorithms for Continuous Optimization (Ed. E. Spedicato), Kluwer Academic Publishers, the Netherlands, 209–253.

[7] A. C. Floudas, P. M. Pardalos, C. S. Adjiman, W. R. Esposito, Z. H. Gümüş, S. T. Harding, J. L. Klepeis, C. A. Meyer, and C. A. Schweiger (1999). *Handbook of*

Test Problems in Local and Global Optimization. Kluwer Academic Publishers, the Netherlands.

[8] R. N. Gasimov (2002). Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming. *Journal of Global Optimization*, **24**, 187–203.

[9] R. N. Gasimov, and N. A. Ismayilova (2005). The modified subgradient method for equality constrained nonconvex optimization problems. Qi, Liqun (ed.) et al., Optimization and Control with Applications. New York, Springer. Applied Optimization **96**, 257–270.

[10] I. Griva, and R. A. Polyak (2006). Primal-dual nonlinear rescaling method with dynamic scaling parameter update. *Mathematical Programming*, Ser. A **106**, 237–259.

[11] W. Hock, and K. Schittkowski (1981). *Test Examples for Nonlinear Programming Codes*, Springer-Verlag, Germany.

[12] R. A. Polyak (2002). Nonlinear rescaling vs. smoothing technique in convex optimization. *Mathematical Programming*, Ser. A **92**, 197–235.

[13] R. A. Polyak, and M. Teboulle (1997). Nonlinear rescaling and proximal-like methods in convex optimization. *Mathematical Programming*, **76**, 265–284.

[14] R. T. Rockafellar (1970). *Convex Analysis*. Princeton University Press.

[15] R. T. Rockafellar, and R. J.-B. Wets (1998). *Variational Analysis*. Springer, Berlin.

[16] A. M. Rubinov, and X. Q. Yang (2003). *Lagrange-type functions in constrained nonconvex optimization*. Kluwer Academic Publishers, Dordrecht.

[17] A. M. Rubinov, X. Q. Yang, and A. M. Bagirov (2003). Penalty functions with a small penalty parameter: numerical experiments. *Optimization Methods and Software*, **17**(5), 931–964.

[18] A. M. Rubinov, X. Q. Yang, A. M. Bagirov, and R. N. Gasimov (2003). Lagrange-type functions in constrained optimization. *Journal of Mathematical Sciences*, **115**(4), 2437–2505.

[19] H. D. Sherali, G. Choi, and C. H. Tuncbilek (2000). A variable target value method for nondifferentiable optimization. *Operations Research Letters*, **26**, 1–8.