

Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities

Krzysztof C. Kiwiel

▶ **To cite this version:**

Krzysztof C. Kiwiel. Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. [Research Report] RR-1742, INRIA. 1992. inria-00076982

HAL Id: inria-00076982
https://inria.hal.science/inria-00076982

Submitted on 29 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNITÉ DE RECHERCHE
INRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.:(1) 39 63 55 11

Rapports de Recherche

1992

25ème anniversaire

N° 1742

Programme 5
Traitement du Signal,
Automatique et Productique

PROXIMAL LEVEL BUNDLE METHODS FOR CONVEX NONDIFFERENTIABLE OPTIMIZATION, SADDLE-POINT PROBLEMS AND VARIATIONAL INEQUALITIES

Krzysztof C. KIWIEL

Août 1992

RR-1742

PROXIMAL LEVEL BUNDLE METHODS FOR CONVEX NONDIFFERENTIABLE OPTIMIZATION, SADDLE-POINT PROBLEMS AND VARIATIONAL INEQUALITIES

MÉTHODES DE FAISCEAUX PROXIMALES À NIVEAU, POUR L'OPTIMISATION CONVEXE NON DIFFÉRENTIABLE, LES PROBLÈMES DE POINTS-SELLES ET LES INÉGALITÉS VARIATIONNELLES

Krzysztof C. KIWIEL[†]

July 29, 1992

ABSTRACT

We study proximal level methods for convex optimization that use projections onto successive approximations of level sets of the objective corresponding to estimates of the optimal value. We show that they enjoy almost optimal efficiency estimates. We give extensions for solving convex constrained problems, convex-concave saddle-point problems and variational inequalities with monotone operators. We present several variants, establish their efficiency estimates, and discuss possible implementations. In particular, our methods require bounded storage in contrast to the original level methods of Lemarechal, Nemirovskii and Nesterov.

RÉSUMÉ

Nous étudions des méthodes d'optimisation proximales qui utilisent la projection sur des approximations successives d'ensembles de niveau de la fonction-coût correspondant à des approximations de la valeur optimale. Nous montrons que ces méthodes ont une efficacité presque optimale. Nous les étendons à la résolution de problèmes convexes avec contraintes, de problèmes convexes-concaves de point-selle, et d'inégalités variationnelles avec opérateur monotone. Nous en présentons plusieurs variantes, établissant une estimation de leur efficacité, et traitant d'implémentations possibles. En particulier, nos méthodes ne requièrent qu'une mémoire limitée, par opposition à la méthode originale de Lemaréchal, Nemirovskii et Nesterov.

[†] On leave from the Systems Research Institute, Polish Academy of Sciences, Newelska 6, 01-447 Warsaw, Poland.

1 Introduction

We consider various modifications of the recently proposed proximal level method of [LNN91] for solving the convex program

$$f^* = \min\{ f(x) : x \in S \} \tag{1.1}$$

under the following assumptions. S is a nonempty compact convex subset of $\mathbb{R}^N$ with diameter $\operatorname{diam}(S) = \sup\{|x-y| : x, y \in S\}$ with respect to the Euclidean norm $|\cdot|$, f is a convex function Lipschitz continuous on S with Lipschitz constant L_f, and for each $x \in S$ we can compute $f(x)$ and a subgradient $g_f(x) \in \partial f(x)$ of f at x such that $|g_f(x)| \le L_f$. S should also be simple enough to allow finding minimizers of convex quadratics over S.

The algorithm that we propose for (1.1) is iterative. It generates trial points $x^k \in S$ at which the linearizations $f^k(\cdot) = f(x^k) + \langle g_f(x^k), \cdot - x^k \rangle$ of f are computed for $k = 1, 2, \ldots$, and lower bounds $f^k_{\text{low}} \le f^*$ such that $f^k_{\text{low}} \uparrow f^*$ as $k \to \infty$. At the kth iteration, f is approximated from below by the piecewise linear (polyhedral) cutting-plane model $\check{f}^k = \max_{j \in J^k} f^j$, where $J^k \subset \{1\!:\!k\}$ contains k and at most N other indices. We set

$$x^{k+1} = \arg\min\{ |x - x^k_c|^2/2 : x \in S,\ \check{f}^k(x) \le f^k_{\text{lev}} \}, \tag{1.2}$$

where the *prox-center* x^k_c is chosen from the points $\{x^j\}_{j=1}^k$, and the target level value

$$f^k_{\text{lev}} = f^k_{\text{up}} - \kappa\Delta^k = \kappa f^k_{\text{low}} + (1-\kappa) f^k_{\text{up}} \tag{1.3}$$

is determined by the record value $f^k_{\text{up}} = \min_{j=1:k} f(x^j)$, the optimality gap $\Delta^k = f^k_{\text{up}} - f^k_{\text{low}}$ and the level parameter $0 < \kappa < 1$. A descent step to $x^{k+1}_c = x^{k+1}$ is taken if $f(x^{k+1})$ is significantly lower than $f(x^k_c)$. Otherwise, a null step $x^{k+1}_c = x^k_c$ occurs.

We show that the method has the following efficiency for any accuracy $\epsilon > 0$:

$$k > c(\kappa)(\operatorname{diam}(S) L_f/\epsilon)^2 \quad \Rightarrow \quad f^k_{\text{up}} - f^* \le \Delta^k < \epsilon, \tag{1.4a}$$

$$c(\kappa) = 1/\kappa^2(1-\kappa^2), \tag{1.4b}$$

$$\min c(\cdot) = c(1/\sqrt{2}) = 4. \tag{1.4c}$$

We recall from [LNN91, NYu83] that such estimates are optimal in a certain sense: if S is a ball and $N \ge (\operatorname{diam}(S) L_f/\epsilon)^2/4$ then for any method that uses at most $(\operatorname{diam}(S) L_f/\epsilon)^2/4$ objective and subgradient evaluations there exists a function for which this method does not obtain an accuracy better than ϵ.

Our method stems from the pioneering algorithm of [LNN91], which also has the same efficiency (1.4). However, the latter is much more difficult to implement because it employs $J^k = \{1\!:\!k\}$, i.e., $\check{f}^k = \max_{j=1:k} f^j$ in (1.2), and replaces f^k_{low} in (1.3) by $\check{f}^k_{\min} = \min_S \check{f}^k$, requiring storage of order kN at iteration k. In contrast, we use subgradient selection as in [Kiw85] to ensure $|J^k| \le N+1$ for all k. Moreover, by extending the subgradient aggregation techniques of [Kiw90], we derive a method of the same efficiency in which $\check{f}^k$ may involve only two pieces (f^k and an aggregate linearization $\tilde{f}^{k-1} \in \operatorname{co}\{f^j\}_{j=1}^{k-1}$).

The level approach has been extended in [LNN91] to problems with additional convex constraints, convex-concave saddle-point problems and variational inequalities with monotone operators. As in (1.2), these methods project certain points onto the level sets of

polyhedral approximations to the problem functions. The methods reduce optimality gaps suitable for such problems with efficiency similar to (1.4). They are not, however, readily implementable because they require unbounded storage. We shall, therefore, show how to bound their storage via subgradient selection and aggregation without worsening their efficiency estimates. In effect, we derive algorithms for the above-mentioned problems that may be considered as implementable versions of those in [LNN91].

We may add that preliminary numerical experience with the level methods of [LNN91] has been very encouraging. We hope that our versions will work at least equally well, since they are able to suppress outdated subgradient information. In order to keep this paper reasonably short, we intend to provide numerical evidence elsewhere.

The paper is organized as follows. In §2 we introduce our first proximal level method for (1.1). Its efficiency is analyzed in §3. In §4 we give some useful modifications. Extensions to problems with additional convex constraints, saddle-point problems and variational inequalities are developed in §5, §6 and §7 respectively. Finally, we have a conclusion section.

We use the following notation. We denote by $\langle \cdot, \cdot \rangle$ and $|\cdot|$, respectively, the usual inner product and norm in $\mathbb{R}^N$. Both superscripts and subscripts are used to denote different vectors. For $\epsilon \geq 0$, the ϵ-subdifferential of f at x is defined by $\partial_\epsilon f(x) = \{p \in \mathbb{R}^N : f(y) \geq f(x) + \langle p, y - x \rangle - \epsilon \quad \forall y \in \mathbb{R}^N\}$. We denote by ∂f the ordinary subdifferential $\partial_0 f$. Given a nonempty closed convex set $C \subset \mathbb{R}^N$ and $x \in \mathbb{R}^N$, we let $P_C(x) = \arg\min\{|x - y| : y \in C\}$ denote the orthogonal projection of x on C. Recall that

$$|x - x_c|^2 \geq |P_C(x_c) - x_c|^2 + |x - P_C(x_c)|^2 \quad \forall x \in C, x_c \in \mathbb{R}^N. \tag{1.5}$$

The natural logarithm with base e is denoted by $\ln(\cdot)$. We let $1{:}k$ denote $1, 2, \ldots, k$. For brevity, we let $a/bc = a/(bc)$. The convex hull is denoted by co.

2 The proximal level algorithm

By (1.2), we have $x^{k+1} = P_{S^k}(x_c^k)$ if $S^k = \{x \in S : \check{f}^k(x) \leq f_{\text{lev}}^k\}$ is nonempty. Since $\check{f}^k \leq f$, S^k contains the optimal set $S^* = \operatorname{Arg\,min}_S f$ whenever $f_{\text{lev}}^k \geq f^*$. Therefore, by choosing f_{lev}^k large enough, we may always ensure that $S^k \neq \emptyset$. Then x^{k+1} solves the convex problem

$$\begin{aligned} &\text{minimize} && |x - x_c^k|^2/2 \quad \text{over all } x \in S && \text{(2.1a)} \\ &\text{satisfying} && f^j(x) \leq f_{\text{lev}}^k \quad \text{for} \quad j \in J^k. && \text{(2.1b)} \end{aligned}$$

Denote the Lagrange multipliers of (2.1b) by λ_j^k, $j \in J^k$. Let $\hat{J}^k = \{j \in J^k : \lambda_j^k > 0\}$. By the Kuhn-Tucker (K-T) conditions, if we select $J_s^k \subset J^k$ such that $\hat{J}^k \subset J_s^k$, then J_s^k may replace J^k in (2.1) without changing its solution (see, e.g., [Kiw85, Kiw90] for similar arguments). In other words, we see that $\check{f}_s^k = \max_{j \in J_s^k} f^j$ may replace $\check{f}^k$ in (1.2), so that $x^{k+1} = P_{S_s^k}(x_c^k)$ for $S_s^k = \{x \in S : \check{f}_s^k(x) \leq f_{\text{lev}}^k\}$. This suggests that only the linearizations f^j, $j \in J_s^k$, that have contributed to x^{k+1} need be retained for the next iteration.

We may now state our first proximal level algorithm.

Algorithm 2.1

Step 0 (*Initialization*). Select an initial point $x^1 \in S$, a final optimality tolerance $\epsilon_{\text{opt}} \geq 0$ and a level parameter $0 < \kappa < 1$. Set $x_c^1 = x^1$. Set $f_{\text{up}}^0 = \infty$ and $f_{\text{low}}^1 = \min_S f^1$, where

$f^1 = f(x^1) + < g_f(x^1), \cdot - x^1 >$. Set $J^1 = \{1\}$. Set the counters $k = 1$, $l = 0$ and $k(0) = 1$ ($k(l)$ will denote the iteration number of the lth increase of f_{low}^k).

Step 1 (*Level update*). Set $f_{\text{up}}^k = \min\{f(x^k), f_{\text{up}}^{k-1}\}$, $\Delta^k = f_{\text{up}}^k - f_{\text{low}}^k$ and f_{lev}^k by (1.3).

Step 2 (*Stopping criterion*). If $\Delta^k \leq \epsilon_{\text{opt}}$ or $g_f(x^k) = 0$, terminate; otherwise, continue.

Step 3 (*Level feasibility check*). If (2.1) is feasible, go to Step 5; otherwise, continue.

Step 4 (*Update lower bound*). Find $\check{f}_{\text{min}}^k = \min_S \check{f}^k$. Set $\hat{f}_{\text{low}}^k = \check{f}_{\text{min}}^k$, $\hat{\Delta}^k = f_{\text{up}}^k - \hat{f}_{\text{low}}^k$ and $f_{\text{low}}^k = \hat{f}_{\text{low}}^k$. Choose $x_c^k \in \{x^j : j \in J^k\}$. Set $k(l+1) = k$ and increase l by 1. Go to Step 1.

Step 5 (*Projection*). Find the solution x^{k+1} of (2.1) and its multipliers λ_j^k such that the set $\hat{J}^k = \{j \in J^k : \lambda_j^k > 0\}$ satisfies $|\hat{J}^k| \leq N$.

Step 6 (*Objective evaluation*). Calculate $f(x^{k+1})$ and $g_f(x^{k+1})$.

Step 7 (*Selection*). Select $J_s^k \subset J^k$ such that $\hat{J}^k \subset J_s^k$. Set $J^{k+1} = J_s^k \cup \{k+1\}$, $x_c^{k+1} = x_c^k$, $f_{\text{low}}^{k+1} = f_{\text{low}}^k$ and, if $k > k(l)$, $\hat{f}_{\text{low}}^k = f_{\text{low}}^k$ and $\hat{\Delta}^k = \Delta^k$. Increase k by 1 and go to Step 1.

A few comments on the method are in order.

At Step 0, one may choose any $f_{\text{low}}^1 \in [\min_S f^1, f^*]$, e.g., $f_{\text{low}}^1 = f^*$ if it is known. This will not invalidate our subsequent efficiency estimates.

Step 2 is justified by the fact that $f(x^k) = f^*$ if $g_f(x^k) = 0$, and by the optimality estimates (2.2). If we let $x_{\text{rec}}^k \in \{x^j\}_{j=1}^k$ denote the kth 'record' point such that $f(x_{\text{rec}}^k) = f_{\text{up}}^k$, then upon termination x_{rec}^k is an ϵ_{opt}-solution to (1.1).

In most applications, S will be described by finitely many linear inequalities. Then Step 3 may check if $S^k \neq \emptyset$ and Step 4 find $\min_S \check{f}^k$ via linear programming (LP). At Step 5, many quadratic programming (QP) methods for (2.1) will automatically produce $|\hat{J}^k| \leq N$ (because (2.1) involves N variables). Hence Step 7 may choose J^{k+1} such that $|J^{k+1}| \leq N+1$. Moreover, using the representation $f^j = f^j(x_c^k) + \left\langle g_f(x^j), \cdot - x_c^k \right\rangle$ in (2.1), we do not need x^j for updating $f^j(x_c^{k+1}) = f^j(x_c^k) + \left\langle g_f(x^j), x_c^{k+1} - x_c^k \right\rangle$ at Step 4. Thus the required storage is of order $(N+1)^2$ (plus the QP workspace).

At Step 4, we have $\check{f}_{\text{min}}^k \leq f^*$ (from $\check{f}^k \leq f$) and $f_{\text{lev}}^k < \check{f}_{\text{min}}^k$ (otherwise $\emptyset \neq \text{Arg}\min_S \check{f}^k \subset S^k$ at Step 3). Since Step 4 increases f_{low}^k to $\check{f}_{\text{min}}^k$, and then Step 1 sets $f_{\text{lev}}^k \geq f_{\text{low}}^k$ (cf. (1.3)), we deduce that Step 4 may be executed at most once at each iteration. In particular, $\hat{f}_{\text{low}}^k$ and $\hat{\Delta}^k$ denote the final values of f_{low}^k and Δ^k at iteration k. The steps at which these quantities and f_{lev}^k are considered should be clear from context. With this notation in mind, several simple properties of the method may by derived inductively from the following observations.

Let us split the iterations into groups $K^l = \{k(l): k(l+1) - 1\}$ for $l \geq 0$ ($K^l = \emptyset$ if $k(l) = k(l+1)$). By construction, $f_{\text{up}}^k \geq f_{\text{up}}^{k+1} \geq f^* \geq f_{\text{low}}^{k+1} \geq \hat{f}_{\text{low}}^k \geq f_{\text{low}}^k$, $\hat{\Delta}^k = f_{\text{up}}^k - \hat{f}_{\text{low}}^k$ and $\Delta^k = f_{\text{up}}^k - f_{\text{low}}^k$, so the gaps $\hat{\Delta}^k \leq \Delta^k$ over-estimate the optimality gap:

$$f_{\text{up}}^k - f^* = \min\{f(x^j) : j = 1{:}\,k\} - f^* \leq \hat{\Delta}^k \leq \Delta^k \tag{2.2}$$

and $\Delta^{k+1} \leq \hat{\Delta}^k \leq \Delta^k$ for all k. Moreover, $f_{\text{low}}^k = \hat{f}_{\text{low}}^{k(l)}$ at Step 5, so the level $f_{\text{lev}}^k = f_{\text{low}}^k + (1-\kappa)\Delta^k$ cannot increase within each group:

$$f_{\text{lev}}^{k(l)} \geq f_{\text{lev}}^j \geq f_{\text{lev}}^k \quad \text{if} \quad k(l) < j \leq k \leq k(l+1). \tag{2.3}$$

Hence $\hat{f}^k_{\rm low}$ and $\hat{\Delta}^k$ only reflect the improvement in $f^k_{\rm low}$ and Δ^k at iterations $k = k(l+1)$, $l \geq 0$. Then at Step 4, $\hat{f}^k_{\rm low} \geq f^k_{\rm lev} = f^k_{\rm up} - \kappa\Delta^k$ implies $\hat{\Delta}^k = f^k_{\rm up} - \hat{f}^k_{\rm low} \leq \kappa\Delta^k$. Thus we have the useful relation

$$\Delta^k \geq \hat{\Delta}^k \geq \hat{\Delta}^{k(l+1)}/\kappa \quad \text{if} \quad k \in K^l \text{ and } l \geq 0. \tag{2.4}$$

In words: each group K^l (except K^0) starts by discovering that the target level is unattainable. Then increasing the lower bound reduces the gap between the bounds by at least a fraction of $\kappa < 1$. The remaining level and gap decreases within each group occur only when the objective improves, with the lower bound staying fixed.

Note that the prox-center x^k_c is fixed in each group. This stabilizes the search for a better point. At Step 4, one may choose x^k_c such that $f(x^k_c) = \min_{j \in J^k} f(x^j)$.

3 Convergence

Our aim is to establish the efficiency estimate (1.4) for Algorithm 2.1. We assume, with no loss of generality, that the tolerance $\epsilon_{\rm opt} = 0$ and that the algorithm does not terminate.

We start by relating the gaps Δ^k with lower bounds for changes in x^k.

Lemma 3.1. *At Step 5 we have* $|x^{k+1} - x^k| \geq \kappa\Delta^k/L_f$ *if* $k > k(l)$, *and* $|x^{k+1} - x^k_c| \geq \kappa\Delta^k/L_f$ *if* $k = k(l)$.

Proof. Recall that $f^k_{\rm up} = \min_{j=1:k} f(x^j)$. If $k > k(l)$, let $j = k$. Then $f^j \leq \check{f}^k$ from $j \in J^k$. Since $x^{k+1} \in S^k$, we have $f^j(x^{k+1}) = f(x^j) + \left\langle g_f(x^j), x^{k+1} - x^j \right\rangle \leq \check{f}^k(x^{k+1}) \leq f^k_{\rm lev}$, so

$$L_f|x^{k+1} - x^j| \geq |g_f(x^j)||x^{k+1} - x^j| \geq f(x^j) - f^k_{\rm lev} \geq f^k_{\rm up} - (f^k_{\rm up} - \kappa\Delta^k) = \kappa\Delta^k$$

from the assumption $|g_f(x^j)| \leq L_f$, the Cauchy-Schwarz inequality and (1.3). If $k = k(l)$, let $j \in J^k$ above be such that $x^k_c = x^j$ at Step 4. □

Recall that $x^{k+1} = P_{S^k}(x^k_c) = P_{S^k_s}(x^k_c)$ and $S^k_s = \{x \in S : \check{f}^k_s(x) \leq f^k_{\rm lev}\}$.

Lemma 3.2. *If* $k > k(l)$ *at Step 5 then* $|x^{k+1} - x^k_c|^2 \geq |x^k - x^k_c|^2 + |x^{k+1} - x^k|^2$ *and* $x^k_c = x^{k-1}_c$.

Proof. By the rules of Steps 4 and 7, $x^k_c = x^{k-1}_c$, $J^k \supset J^{k-1}_s$ and $\check{f}^k \geq \check{f}^{k-1}_s$, so $S^k \subset S^{k-1}_s$ from (2.3). Apply (1.5) with $C = S^{k-1}_s$, $x_c = x^{k-1}_c$, $x = x^{k+1} \in C$ and $x^k = P_C(x^{k-1}_c)$. □

We may now estimate the rate of decrease of the gap Δ^k within each group K^l.

Lemma 3.3. *If* $k(l) \leq k < k(l+1)$ *and* $\Delta^k > 0$ *then* $k - k(l) + 1 \leq (\operatorname{diam}(S)L_f/\kappa\Delta^k)^2$.

Proof. Note that $\Delta^j \geq \Delta^k$ for $j = 1{:}k$ because Δ^j never increases. Applying Lemmas 3.1 and 3.2 with $x^j_c = x^{k(l)}_c$ for $j = k(l){:}k$ and $x^{k+1}, x^k_c \in S$, we get

$$\begin{aligned} \operatorname{diam}(S)^2 &\geq |x^{k+1} - x^k_c|^2 \geq |x^{k(l)+1} - x^{k(l)}_c|^2 + \sum_{j=k(l)+1}^{k} |x^{j+1} - x^j|^2 \\ &\geq \sum_{j=k(l)}^{k} (\kappa\Delta^j/L_f)^2 \geq (\kappa\Delta^k/L_f)^2(k - k(l) + 1). \quad \square \end{aligned}$$

Lemma 3.4. *If* $\hat{\Delta}^{k_\epsilon} \geq \epsilon > 0$ *for some* k_ϵ *then* $k_\epsilon \leq (\mathrm{diam}(S)L_f/\epsilon)^2/\kappa^2(1-\kappa^2)$.

Proof. (i) Let $K(\epsilon) = \{1{:}\,k_\epsilon\} \subset \bigcup_{l=0}^m K^l$. Since $\hat{\Delta}^{k_\epsilon} \geq \epsilon > 0$ and $\Delta^{k+1} \leq \hat{\Delta}^k \leq \Delta^k$ for all k, use (2.4) and induction to obtain $\Delta^k \geq \epsilon/\kappa^{m-l}$ for all $k \in K^l \cap K(\epsilon)$ and $l = 0{:}\,m$.

(ii) Let $b = (\mathrm{diam}(S)L_f/\kappa\epsilon)^2$. By (i) and Lemma 3.3, $|K^l \cap K(\epsilon)| \leq b\kappa^{2(m-l)}$ for $l = 0{:}\,m$. Hence $k_\epsilon = \sum_{l=0}^m |K^l \cap K(\epsilon)| \leq \sum_{l=0}^m b\kappa^{2(m-l)} \leq b/(1-\kappa^2)$. □

The next two results follow immediately from Lemma 3.4 by contradiction.

Theorem 3.5. *The efficiency estimate* (1.4) *holds for each* $\epsilon > 0$. □

Corollary 3.6. *If* $\epsilon_{\mathrm{opt}} = \epsilon > 0$ *then the algoritm will terminate with* $f(x^k_{\mathrm{rec}}) \leq f^* + \epsilon$ *in* $k = 1 + k_\epsilon$ *iterations, where* $k_\epsilon \leq (\mathrm{diam}(S)L_f/\epsilon)^2/\kappa^2(1-\kappa^2)$. □

For completeness, we include an asymptotic result.

Theorem 3.7. *If the algorithm does not terminate then* f^k_{up}, f^k_{low} *and* f^k_{lev} *converge to* f^*, *and* Δ^k *and* $\hat{\Delta}^k$ *converge to zero as* $k \to \infty$. *Moreover,* $\{x^k_{\mathrm{rec}}\}$ *converges to* $\mathrm{Arg\,min}_S f$.

Proof. Since $\hat{\Delta}^k > 0$ never increases, $\hat{\Delta}^k \downarrow 0$ from Lemma 3.4 (by contradiction). Hence the facts that $\Delta^{k+1} \leq \hat{\Delta}^k \leq \Delta^k$ and $\max\{|f^k_{\mathrm{low}} - f^*|, |f^k_{\mathrm{up}} - f^*|, |f^k_{\mathrm{lev}} - f^*|\} \leq \Delta^k$ for all k imply the first assertion. The second one follows from $f(x^k_{\mathrm{rec}}) \downarrow f^*$, the continuity of f and the compactness of S. □

4 Modifications

Algoritm 2.1 keeps the prox-center x^k_c and the lower bound f^k_{low} fixed within each group K^l. Our efficiency analysis seems to forbid the most 'natural' choices of $x^k_c = x^k_{\mathrm{rec}}$ (with $f(x^k_c) = \min_{j=1:k} f(x^j)$) and $f^k_{\mathrm{low}} = \min_S \check{f}^k$ for all k. Consider, therefore, the following modification of Step 3 that attempts to finish each group quickly in order to allow the 'natural' choices more frequently. Let $\hat{\kappa} \in [\kappa, 1)$ be fixed.

Step 3' (*Group termination test*). If $S^k = \emptyset$ or at least one of the following four tests is passed: (i) $\Delta^k \leq \hat{\kappa}\hat{\Delta}^{k-1}$ and $k > k(l)$, (ii) $f^k_{\mathrm{up}} - \check{f}^k_{\mathrm{min}} \leq \hat{\kappa}\Delta^k$, (iii) $\Delta^k \leq \hat{\kappa}\hat{\Delta}^{k(l)}$, or (iv) $f^k_{\mathrm{up}} - \check{f}^k_{\mathrm{min}} \leq \hat{\kappa}\hat{\Delta}^{k(l)}$, then go to Step 4, setting $\hat{f}^k_{\mathrm{low}} = \max\{f^k_{\mathrm{low}}, \check{f}^k_{\mathrm{min}}\}$ (or $\hat{f}^k_{\mathrm{low}} = f^k_{\mathrm{low}}$ to avoid finding $\check{f}^k_{\mathrm{min}}$ when the first or third test is passed and $S^k \neq \emptyset$). Otherwise, go to Step 5.

Clearly, (2.4) holds with κ replaced by $\hat{\kappa}$ if the last two tests of Step 3' are not used; otherwise, these tests provide the weaker relations

$$\hat{\Delta}^{k(l+1)} \leq \hat{\kappa}\hat{\Delta}^{k(l)} \quad \text{and} \quad \Delta^k \geq \hat{\Delta}^k \geq \hat{\Delta}^{k(l+1)} \quad \text{if} \quad k \in K^l \text{ and } l \geq 0. \tag{4.1}$$

Theorem 4.1. (a) *If Step* 3' *is employed then the following estimate holds for any* $\epsilon > 0$:

$$k > c(\kappa, \hat{\kappa})(\mathrm{diam}(S)L_f/\epsilon)^2 \quad \Rightarrow \quad f^k_{\mathrm{up}} - f^* \leq \hat{\Delta}^k < \epsilon, \tag{4.2a}$$

$$c(\kappa, \hat{\kappa}) = 1/\kappa^2\hat{\kappa}^2(1-\hat{\kappa}^2) \quad \text{and} \quad \min c(\cdot,\cdot) = c(\sqrt{2/3}, \sqrt{2/3}) = 6.75, \tag{4.2b}$$

but if the last two tests of Step 3' are omitted then (4.2b) *may be replaced by*

$$c(\kappa,\hat\kappa) = 1/\kappa^2(1-\hat\kappa^2) \quad \text{and} \quad \min c(\cdot,\cdot) = c(1/\sqrt{2},1/\sqrt{2}) = 4. \tag{4.2c}$$

(b) *If Step 4 is allowed to choose any* $x_c^k \in S$ *then* (4.2b) *may be substituted by*

$$c(\kappa,\hat\kappa) = 1/\kappa^2\hat\kappa^2(1-\hat\kappa^2) - 1/2e\ln(\hat\kappa), \tag{4.2d}$$

$$\min c(\cdot,\cdot) = c(0.796718\ldots,0.796718\ldots) \approx 7.60461. \tag{4.2e}$$

If additionally the last two tests of Step 3' are omitted then (4.2b) *may be replaced by*

$$c(\kappa,\hat\kappa) = 1/\kappa^2(1-\hat\kappa^2) - 1/2e\ln(\hat\kappa), \tag{4.2f}$$

$$\min c(\cdot,\cdot) = c(0.677653\ldots,0.677653\ldots) \approx 4.49950. \tag{4.2g}$$

Proof. (i) We start with the second part of assertion (b). Since the second bound of Lemma 3.1 need not be true for arbitrary $x_c^k \in S$ and $l > 0$, Lemma 3.3 now reads: $k(l) \le k < k(l+1)$ and $\Delta^k > 0$ imply $k - k(l) \le (\mathrm{diam}(S)L_f/\kappa\Delta^k)^2$, where $k(l)$ may be dropped if $l = 0$ because $x_c^1 = x^1$ and $1 \in J^1$.

(ii) Suppose $\hat\Delta^{k_\epsilon} \ge \epsilon > 0$ for some $k_\epsilon \in K^m$ and $m \ge 0$. As in the proof of Lemma 3.4, (2.4) with κ replaced by $\hat\kappa$ implies $\Delta^k \ge \epsilon/\hat\kappa^{m-l}$ for all $k \in K^l \cap K(\epsilon)$ and $l = 0{:}m$. Therefore, with $b = (\mathrm{diam}(S)L_f/\kappa\epsilon)^2$, (i) yields $|K^l \cap K(\epsilon)| \le 1 + b\hat\kappa^{2(m-l)}/\epsilon^2$ for $l = 1{:}m$ and $|K^0 \cap K(\epsilon)| \le b\hat\kappa^{2m}/\epsilon^2$. Hence $k_\epsilon = \sum_{l=0}^m |K^l \cap K(\epsilon)| \le m + \sum_{l=0}^m b\hat\kappa^{2(m-l)} \le m + b/(1-\hat\kappa^2)$, i.e., $k_\epsilon \le (\mathrm{diam}(S)L_f/\epsilon)^2/\kappa^2(1-\hat\kappa^2) + m$. Therefore, (4.2a,f) will be proved (by contradiction) if we show that $m \le -(\mathrm{diam}(S)L_f/\epsilon)^2/2e\ln(\hat\kappa)$. If $m > 0$ then (2.4) with κ replaced by $\hat\kappa$ yields $\epsilon \le \hat\Delta^{k(m)} \le \hat\kappa^m\Delta^1$, where $\Delta^1 = f(x^1) - f(x^1) - \min_S \langle g_f(x^1), \cdot - x^1\rangle \le L_f\,\mathrm{diam}(S)$ by the Cauchy-Schwarz inequality, so $m \le -\ln(\mathrm{diam}(S)L_f/\epsilon)/\ln(\hat\kappa)$. Thus, to get the required bound, it suffices to prove that $-\ln(t)/\ln(\hat\kappa) \le -t^2/2e\ln(\hat\kappa)$ for all $t > 0$. Indeed, $t^2 - 2e\ln(t) \ge 0$ for all $t > 0$ (minimize it!).

(iii) For the first part of assertion (b), in (ii) refer to (4.1) instead of (2.4) to get $\Delta^k \ge \epsilon/\hat\kappa^{\max\{m-l-1,0\}}$ for all $k \in K^l \cap K(\epsilon)$ and $l = 0{:}m$, $k_\epsilon \le b/\hat\kappa^2(1-\hat\kappa^2) + m$, and $\epsilon \le \hat\Delta^{k(m)} \le \hat\kappa^m\Delta^1$ if $m > 0$. Hence the desired bounds may be obtained as in (ii).

(iv) To verify assertion (a), proceed as in (ii) and (iii) but invoke Lemma 3.2 instead of (i) to drop m from the bounds for k_ϵ. □

The worst-case efficiency estimates of Theorem 4.1 deteriorate if $\hat\kappa \ne \kappa$, but such modifications may be useful in practice anyway. In particular, if $\check f_{\min}^k$ is calculated for each k, then the corresponding test of Step 3' may be employed to ensure that $f_{\mathrm{lev}}^k > \check f_{\min}^k + (\hat\kappa - \kappa)\Delta^k$ at Step 5 due to $f_{\mathrm{up}}^k - \check f_{\min}^k > \hat\kappa\Delta^k$, i.e., f_{lev}^k is 'significantly' larger than $\check f_{\min}^k$ if $\hat\kappa > \kappa$.

We may add that the estimate (4.2a,b) (which is slightly weaker than (1.4)) is derived in [LNN91] for the original proximal level method that employs a test similar to $\Delta^k \le \kappa\hat\Delta^{k(l)}$. That method keeps $x_c^k = x_{\mathrm{rec}}^k$ (i.e., $f(x_c^k) = f_{\mathrm{up}}^k$) by setting $x_c^k = x^k$ if $f(x^k) < f(x_c^{k-1})$, and $x_c^k = x_c^{k-1}$ otherwise. With Step 3', our method may change x_c^k to the best point available at Step 4 if $S^k = \emptyset$, $f_{\mathrm{up}}^k - \check f_{\min}^k \le \hat\kappa\hat\Delta^{k(l)}$, $\Delta^k \le \hat\kappa\hat\Delta^{k-1}$ or $\Delta^k \le \hat\kappa\hat\Delta^{k(l)}$. Observe that the last two inequalities may be rewritten as

$$f(x^k) \le f_{\mathrm{up}}^{k-1} - (1-\hat\kappa)\hat\Delta^{k-1}, \tag{4.3a}$$

$$f_{\mathrm{up}}^k \le f_{\mathrm{up}}^{k-1} - (1-\hat\kappa)\hat\Delta^{k(l)}. \tag{4.3b}$$

In other words, for stability reasons, our method replaces the more 'natural' test $f(x^k) < f(x_c^{k-1})$ of [LNN91] with the Armijo-like tests (4.3) for a 'substantial' descent. (One may employ $\hat{\kappa} = 0.9$ to emulate the first test by (4.3a) or (4.3b) without blowing up the efficiency estimates.) Now, if $x_{\rm rec}^k = x^j$ with $j = j_{\rm rec}^k$ say, then the original version of Step 4 requires $j_{\rm rec}^k \in J^k$ to set $x_c^k = x_{\rm rec}^k$. We may, of course, ensure that $j_{\rm rec}^k \in J^k$ by letting Step 7 select the corresponding linearization. However, Theorem 4.1(b) shows that it is not really necessary to increase storage in this way.

Since subgradient selection may require excessive storage for large N, we now turn to *subgradient aggregation*, in which aggregate linearizations are produced recursively by taking convex combinations of the 'ordinary' linearizations. Aggregation results from the observation that the solution to (2.1) does not change when we replace the constraints (2.1b) by their convex combination with 'normalized' weights determined by the multipliers λ_j^k. This may be formalized as follows (see [Kiw85, Kiw90] for more details).

Let δ_S denote the indicator of S ($\delta_S(x) = 0$ if $x \in S$, ∞ otherwise). By the K-T conditions, (1.2) holds iff $x^{k+1} \in S^k$ and there exist $\tilde{g}_f^k \in \partial \check{f}^k(x^{k+1})$ and $\lambda_f^k \geq 0$ such that $x_c^k - x^{k+1} - \lambda_f^k \tilde{g}_f^k \in \partial \delta_S(x^{k+1})$ and $\lambda_f^k[\check{f}^k(x^{k+1}) - f_{\rm lev}^k] = 0$. Constructing the aggregate linearization $\tilde{f}^k(x) = \check{f}^k(x^{k+1}) + \langle \tilde{g}_f^k, x - x^{k+1} \rangle$ and redefining $\check{f}_s^k = \max\{\tilde{f}^k, \max_{j \in J_s^k} f^j\}$ with *any* $J_s^k \subset J^k$ (possibly $J_s^k = \emptyset$), it is easy to check that $\check{f}_s^k$ may replace $\check{f}^k$ in the conditions above, and hence also in (1.2). Therefore, $\check{f}^{k+1} = \max\{\tilde{f}^k, \max_{j \in J^{k+1}} f^j\}$ with $J^{k+1} = J_s^k \cup \{k+1\}$ would suffice for the proof of Lemma 3.2 (for k increased by 1). Notice that $\tilde{f}^k \leq \check{f}^k \leq f$ and $\tilde{f}^k(x^{k+1}) = \check{f}^k(x^{k+1})$, so that $\tilde{f}^k$ might be treated like any 'ordinary' f^j at later iterations (except when we need $f^j(x^j) = f(x^j)$ at Step 4; see the proof of Lemma 3.1). Another useful characterization of $\tilde{f}^k$ is derived from the form of $\check{f}^k = \max_{j \in J^k} f^j$ as follows. We have $\tilde{g}_f^k \in \partial \check{f}^k(x^{k+1})$ iff there exist multipliers $\tilde{\lambda}_j^k \geq 0$ satisfying $\tilde{\lambda}_j^k[\check{f}^k(x^{k+1}) - f^j(x^{k+1})] = 0$ for $j \in J^k$ and $(\tilde{g}_f^k, 1) = \sum_{j \in J^k} \tilde{\lambda}_j^k (g_f(x^j), 1)$, in which case $\tilde{f}^k = \sum_{j \in J^k} \tilde{\lambda}_j^k f^j$. By the K-T conditions for (2.1), we have $x_c^k - x^{k+1} - \sum_{j \in J^k} \lambda_j^k g_f(x^j) \in \partial \delta_S(x^{k+1})$, $\lambda_j^k \geq 0$ and $\lambda_j^k[f^j(x^{k+1}) - f_{\rm lev}^k] = 0$ for $j \in J^k$. Comparing these conditions with those for (1.2), we may identify $\tilde{\lambda}_j^k = \lambda_j^k / \lambda_f^k$ for $j \in J^k$ if $\lambda_f^k = \sum_{j \in J^k} \lambda_j^k$ is positive. Otherwise, i.e., if all $\lambda_j^k = 0$, we may take any $\tilde{\lambda}_j^k \geq 0$ such that $\sum_{j \in J^k} \tilde{\lambda}_j^k = 1$ to define $\tilde{f}^k = \sum_{j \in J^k} \tilde{\lambda}_j^k f^j$, again obtaining $\tilde{f}^k \leq \check{f}^k$ and $x^{k+1} = P_{S_s^k}(x_c^k)$. Indeed, if all λ_j^k vanish, then the K-T conditions yield $x^{k+1} = P_S(x_c^k)$ ($= x_c^k$ from $x_c^k \in S$), whereas we have $x^{k+1} = P_{S^k}(x_c^k)$ and $S^k \subset S_s^k$.

In view of the above remarks, consider the following use of aggregation in Algorithm 2.1. At Step 0 let $\tilde{f}^0 = f^1$. At Steps 1 through 4 use $\check{f}^k = \max\{\tilde{f}^{k-1}, \max_{j \in J^k} f^j\}$, where $\tilde{f}^{k-1}(\cdot) = \tilde{f}^{k-1}(x_c^k) + \langle \tilde{g}_f^{k-1}, \cdot - x_c^k \rangle$. At Step 5, append to (2.1) the constraint $\tilde{f}^{k-1}(x) \leq f_{\rm lev}^k$, finding its Lagrange multiplier λ_f^k. Set $\lambda_f^k = \sum_{j \in J^k} \lambda_j^k + \lambda_f^k$. If $\lambda_f^k > 0$, set $\tilde{\lambda}_j^k = \lambda_j^k / \lambda_f^k$, $j \in J^k$, and $\tilde{\lambda}_f^k = \lambda_f^k / \lambda_f^k$; otherwise, choose any convex combination, e.g., $\tilde{\lambda}_j^k = 0$ for $j \in J^k$ and $\tilde{\lambda}_f^k = 1$. Set

$$(\tilde{f}^k, \tilde{g}_f^k) = \sum_{j \in J^k} \tilde{\lambda}_j^k (f^j, g_f(x^j)) + \tilde{\lambda}_f^k (\tilde{f}^{k-1}, \tilde{g}_f^{k-1}). \tag{4.4}$$

(Thus $\tilde{f}^{k-1}$ is treated in the same way as any other linear piece f^j of $\check{f}^k$ in constructing $\tilde{f}^k$ as in the preceding paragraph.) At Step 7 choose any $J_s^k \subset J^k$ (possibly $J_s^k = \emptyset$).

It is easy to verify all the preceding convergence results for the above version with ag-

gregation. If $J_s^k \equiv \emptyset$, then only two linearizations f^k and $\tilde{f}^{k-1}$ need be stored. In practice convergence can be slow if J_s^k is too small.

5 Constrained minimization

We shall give an extension of our method for solving the convex constrained program

$$f^* = \min\{ f(x) : x \in S,\ F(x) \le 0 \} \tag{5.1}$$

under the following assumptions. S is a nonempty compact convex subset of $\mathbb{R}^N$, f and F are convex functions Lipschitzian on S with constants L_f and L_F respectively, for each $x \in S$ we can compute $f(x)$, $F(x)$ and subgradients $g_f(x) \in \partial f(x)$ and $g_F(x) \in \partial F(x)$ of f and F at x such that $|g_f(x)| \le L_f$ and $|g_F(x)| \le L_F$, and $F(x) \le 0$ for some $x \in S$.

We associate with (5.1) the improvement functions $H(x;t) = \max\{f(x) - t, F(x)\}$ and $h(t) = \min_S H(\cdot;t)$ defined for for $t \in \mathbb{R}$ and $x \in S$ (see, e.g., [Kiw85, LNN91]).

Lemma 5.1. *For any fixed $t \in \mathbb{R}$, $H(\cdot;t)$ is convex and Lipschitzian on S with constant $L_H = \max\{L_f, L_F\}$. For any fixed $x \in S$, $H(x;\cdot)$ and h are nonincreasing, convex and Lipschitzian on $\mathbb{R}$ with constant 1. Moreover, $h(f^*) = 0$ and $h(t) > 0$ for all $t < f^*$.*

Proof. The function $\max\{\cdot,\cdot\}$ is monotone, convex and Lipschitzian with constant 1 on $\mathbb{R}^2$ in the norm $\|z\|_\infty = \max\{|z_1|, |z_2|\}$. Therefore, the assertions concerning H follow from our assumptions on f and F. For each t, since $H(\cdot;t)$ is continuous on the compact S, there exists $x(t) \in S$ such that $h(t) = H(x(t);t)$, i.e., h is well-defined. Clearly, $\min_S H(\cdot;t)$ is nonincreasing, convex and Lipschitzian in t with constant 1, since $h(t') - h(t) \le H(x(t);t') - H(x(t);t) \le |t' - t|$ for all $t, t' \in \mathbb{R}$. If $h(t) < 0$ then $f(x) < t$ and $F(x) < 0$ for some $x \in S$, so (5.1) yields $t > f^*$ and $h(f^*) = 0$. Similarly, $h(t) \le 0$ implies $t \ge f^*$. □

Notice that if $x \in S$, $t \le f^*$ and $H(x;t) \le \epsilon$ then $f(x) \le f^* + \epsilon$ and $F(x) \le \epsilon$, i.e., x is an ϵ-solution to (5.1). This suggests the following procedure, which requires an initial lower bound $t^{(1)} < f^*$ and a parameter $0 < \mu < 1$. At iteration $n \ge 1$, having some $t^{(n)} < f^*$, minimize $H(\cdot;t^{(n)})$ over S approximately with Algoritm 2.1, stopping when it discovers that $h(t^{(n)}) > \mu H(x^{(n)};t^{(n)})$ for its best point $x^{(n)} \in S$. (In view of Theorem 3.7, a stopping criterion of the form $\Delta^k \le (1-\mu)H^k_{\rm up}$ will work, since $\min_S H(\cdot;t^{(n)}) = h(t^{(n)}) > 0$ due to $t^{(n)} < f^*$ by Lemma 5.1.) Choose any $t^{(n+1)} \in [t^{(n)} + \mu H(x^{(n)};t^{(n)}), f^*)$ (e.g., $t^{(n+1)} = t^{(n)} + \mu H(x^{(n)};t^{(n)})$, since $h(t^{(n)}) = h(t^{(n)}) - h(f^*) \le f^* - t^{(n)}$ from Lemma 5.1). Increase n by 1 and perform the next iteration. This procedure produces $H(x^{(n)};t^{(n)}) \to 0$ and $t^{(n)} \uparrow f^*$, since we have $t^{(n)} + \mu H(x^{(n)};t^{(n)}) \le t^{(n+1)} < f^*$ and $0 < \mu H(x^{(n)};t^{(n)}) < h(t^{(n)}) \le H(x^{(n)};t^{(n)})$ for all n, whereas h is continuous and $h(t) > 0$ for $t < f^*$ by Lemma 5.1. Moreover, for any $\epsilon > 0$, we must have $H(x^{(n)};t^{(n)}) \le \epsilon$ for some $n \le 1 + (f^* - t^{(1)})/\mu\epsilon$. To obtain better bounds, we shall modify a much more refined scheme for updating $t^{(n)}$ due to [LNN91].

Our algorithm will generate trial points $x^k \in S$ at which the linearizations $f^k(\cdot) = f(x^k) + \langle g_f(x^k), \cdot - x^k \rangle$ and $F^k(\cdot) = F(x^k) + \langle g_F(x^k), \cdot - x^k \rangle$ of f and F are computed, and lower bounds $f^k_{\rm low} \le f^*$ such that $f^k_{\rm low} \uparrow f^*$ as $k \to \infty$. At the kth iteration, $H(\cdot; f^k_{\rm low})$ and h are approximated from below by $\check{H}^k(\cdot; f^k_{\rm low}) = \max\{\check{f}^k - f^k_{\rm low}, \check{F}^k\}$ and $\check{h}^k(t) = \min_S \check{H}^k(\cdot;t)$,

where $\check{f}^k = \max_{j\in J_f^k} f^j$, $\check{F}^k = \max_{j\in J_F^k} F^j$ and $J_f^k \cup J_F^k \subset \{1{:}\,k\}$. These models correspond to the following relaxation of (5.1):

$$\check{f}^k_{\min} = \min\{\, \check{f}^k(x) : x \in S,\ \check{F}^k(x) \le 0 \,\}, \tag{5.2}$$

i.e., we have $\check{f}^k_{\min} \le f^*$ and $\check{h}^k(\check{f}^k_{\min}) = 0$ by Lemma 5.1 applied to (5.2). We set $x^{k+1} = \arg\min\{|x - x_c^k| : x \in S,\ \check{H}^k(x; f^k_{\text{low}}) \le H^k_{\text{lev}}\}$, where $x_c^k \in \{x^j\}_{j=1}^k$ and H^k_{lev} depends on the current bounds $H^k_{\text{low}} \le h(f^k_{\text{low}}) = \min_S H(\cdot; f^k_{\text{low}}) \le H^k_{\text{up}}$. Thus x^{k+1} solves the problem

$$\begin{aligned} &\text{minimize} && |x - x_c^k|^2/2 \quad \text{over all } x \in S && \text{(5.3a)}\\ &\text{satisfying} && f^j(x) \le H^k_{\text{lev}} + f^k_{\text{low}} \quad \text{for} \quad j \in J_f^k, && \text{(5.3b)}\\ & && F^j(x) \le H^k_{\text{lev}} \quad \text{for} \quad j \in J_F^k. && \text{(5.3c)} \end{aligned}$$

We may now state our extension of Algorithm 2.1.

Algorithm 5.2

Step 0 (*Initialization*). Select an initial point $x^1 \in S$, a final optimality tolerance $\epsilon_{\text{opt}} \ge 0$, a level parameter $0 < \kappa < 1$ and $\mu \in (1/2, 1)$. Set $k = 1$, $x_c^1 = x^1_{\text{rec}} = x^1$ and $J_f^1 = J_F^1 = \{1\}$. Set $f^1_{\text{low}} = \check{f}^1_{\min}$ by (5.2), $H^1_{\text{low}} = 0$ and $H^k_{\text{up}} = H(x^1; f^1_{\text{low}})$. Set the counters $l = 0$, $k(0) = 1$ and $k'(0) = 1$ ($k(l)$ will denote the iteration number of the lth increase of f^k_{low} to $\check{f}^{k'(l-1)}_{\min}$).

Step 1 (*Level update*). Set $\Delta^k = H^k_{\text{up}} - H^k_{\text{low}}$ and $H^k_{\text{lev}} = H^k_{\text{up}} - \kappa\Delta^k$.

Step 2 (*Stopping criterion*). If $H(x^k_{\text{rec}}; f^k_{\text{low}}) \le \epsilon_{\text{opt}}$, terminate; otherwise, continue.

Step 3 (*Gap test*). If $\Delta^k \ge (1 - \mu)H^k_{\text{up}}$, go to Step 5; otherwise, continue.

Step 4 (*Group start*). Set $f^k_{\text{low}} = \check{f}^{k'(l)}_{\min}$, $H^k_{\text{up}} = H(x^k_{\text{rec}}; f^k_{\text{low}})$ and $H^k_{\text{low}} = 0$. Set $k(l+1) = k$ and increase l by 1. Go to Step 1.

Step 5 (*Level feasibility check*). If $\check{h}^k(f^k_{\text{low}}) \le H^k_{\text{lev}}$, go to Step 7; otherwise, continue.

Step 6 (*Update lower bound*). Set $H^k_{\text{low}} = \check{h}^k(f^k_{\text{low}})$, $\check{f}^k_{\min}$ by (5.2) and $k'(l) = k$. Choose $x_c^k \in \{x^j : j \in J_f^k \cap J_F^k\}$. Go to Step 1.

Step 7 (*Projection*). Find the solution x^{k+1} of (5.3) and its multipliers λ^k_{fj} and λ^k_{Fj} such that the sets $\hat{J}_f^k = \{j \in J_f^k : \lambda^k_{fj} > 0\}$ and $\hat{J}_F^k = \{j \in J_F^k : \lambda^k_{Fj} > 0\}$ satisfy $|\hat{J}_f^k| + |\hat{J}_F^k| \le N$.

Step 8 (*Record updates*). If $H(x^{k+1}; f^k_{\text{low}}) < H^k_{\text{up}}$, set $H^{k+1}_{\text{up}} = H(x^{k+1}; f^k_{\text{low}})$ and $x^{k+1}_{\text{rec}} = x^{k+1}$; otherwise, set $H^{k+1}_{\text{up}} = H^k_{\text{up}}$ and $x^{k+1}_{\text{rec}} = x^k_{\text{rec}}$.

Step 9 (*Selection*). Select $J^k_{fs} \subset J^k_f$ and $J^k_{Fs} \subset J^k_F$ such that $\hat{J}^k_f \subset J^k_{fs}$ and $\hat{J}^k_F \subset J^k_{Fs}$. Set $J_f^{k+1} = J^k_{fs} \cup \{k+1\}$, $J_F^{k+1} = J^k_{Fs} \cup \{k+1\}$, $x_c^{k+1} = x_c^k$, $H^{k+1}_{\text{low}} = H^k_{\text{low}}$ and $f^{k+1}_{\text{low}} = f^k_{\text{low}}$. Increase k by 1 and go to Step 1.

A few comments on the method are in order.

Let us split the iterations into groups $K^l = \{k(l){:}\,k(l+1) - 1\}$ for $l \ge 0$ (which differ from those in §2). For $k = k(l)$, Step 4 (or 0) gives $f^k_{\text{low}} = \check{f}^{k'(l)}_{\min} \le f^*$ (since $\check{f}^k_{\min} \le f^*\ \forall k$), and hence $H^k_{\text{low}} = 0 \le h(f^k_{\text{low}}) \le H^k_{\text{up}}$ by Lemma 5.1, whereas Step 9 yields $f^k_{\text{low}} = f^{k(l)}_{\text{low}}$ for $k \in K^l$, $k > k(l)$. Thus inside each group K^l our method behaves like a version of Algorithm

2.1 for minimizing $H(\cdot; f_{\rm low}^{k(l)})$ over S that uses more refined models $\check{H}^k(\cdot; f_{\rm low}^{k(l)})$, which do not impair the efficiency estimates of §3. In particular, since $H_{\rm low}^k$ cannot decrease within K^l, we have $0 \le H_{\rm low}^k \le H_{\rm lev}^k$ for all $k \in K^l$. It is worth observing that $f_{\rm low}^{k(l+1)} > f_{\rm low}^{k(l)}$ for the next group. Indeed, for $k = k(l+1)$ Step 4 is entered if $H_{\rm low}^k > \mu H_{\rm up}^k \ge 0$, so $H_{\rm low}^k$ must have been increased at iteration $k = k'(l) \in K^l$ at Step 6 to $H_{\rm low}^k = \check{h}^k(f_{\rm low}^k) > H_{\rm lev}^k \ge 0$, whereas $\check{h}^k(\check{f}_{\rm min}^k) = 0$, and hence $f_{\rm low}^k < \check{f}_{\rm min}^k$ by Lemma 5.1 applied to (5.2). Since $H(x_{\rm rec}^k; \cdot)$ is nonincreasing, Steps 4 and 9 produce $H_{\rm up}^{k+1} \le H_{\rm up}^k = H(x_{\rm rec}^k; f_{\rm low}^k)$ for all k. We also note that $\check{h}^{k'(l)}(f_{\rm low}^{k(l)}) = H_{\rm low}^k > \mu H_{\rm up}^k \ge \mu h(f_{\rm low}^{k(l)})$ at Step 3 for $k = k(l+1)$, i.e.,

$$\check{h}^{k'(l)}(f_{\rm low}^{k(l)}) > \mu H(x_{\rm rec}^{k(l+1)}; f_{\rm low}^{k(l)}) \ge \mu h(f_{\rm low}^{k(l)}). \tag{5.4}$$

In other words, $f_{\rm low}^{k(l)}$ is increased to the smallest root of $\check{h}^{k'(l)}$, and (5.4) with $\mu > 1/2$ ensures that $\check{h}^{k'(l)}$ is close enough to h at $f_{\rm low}^{k(l)}$ in order for $\check{f}_{\rm min}^{k'(l)}$ to be significantly closer to the required root f^* of h.

Steps 4 and 6 may be entered at most twice at each iteration. Indeed, if Step 4 gives $\check{h}^k(f_{\rm low}^k) > \max\{(1-\kappa), \mu\} H_{\rm up}^k$, then Step 6 will set $k'(l) = k$, and the next $\check{h}^k(f_{\rm low}^k) = 0$ at Step 4; this is the only possibility. Such empty groups are harmless, but one may wish to avoid them as follows. First, at Step 4 one may check if the new $\check{h}^k(f_{\rm low}^k) > \mu H_{\rm up}^k$, in which case $f_{\rm low}^k$ may be replaced by $\check{f}_{\rm min}^k$ and $k'(l)$ increased to k. Second, one may use this test at Step 3 on all (or some) iterations, i.e., if $\check{h}^k(f_{\rm low}^k) > \mu H_{\rm up}^k$ then Step 4 may be entered with $k'(l) = k$. We may add that such modifications are in fact covered by our efficiency estimate given below, because the crucial property (5.4) is not impaired.

Upon termination at Step 2, $x_{\rm rec}^k$ is an $\epsilon_{\rm opt}$-solution to (5.1), since $f_{\rm low}^k \le f^*$.

Theorem 5.3. *The following efficiency estimate holds for any* $0 < \epsilon \le \operatorname{diam}(S) L_H$*:*

$$k > c(\kappa,\mu)(\operatorname{diam}(S) L_H/\epsilon)^2 \ln(8 \operatorname{diam}(S) L_H/\epsilon) \quad \Rightarrow \quad H(x_{\rm rec}^k; f_{\rm low}^k) < \epsilon, \tag{5.5a}$$

$$c(\kappa,\mu) = 1/\kappa^2(1-\kappa^2)\ln(2\mu)(1-\mu)^2, \tag{5.5b}$$

$$\min c(\cdot,\cdot) = c(1/\sqrt{2}, 0.652530\ldots) \approx 124.434. \tag{5.5c}$$

Proof. (i) Suppose $H_{\rm up}^{k_\epsilon} \ge \epsilon$ for some $0 < \epsilon \le \operatorname{diam}(S) L_H$, $k_\epsilon \in K^m$ and $m \ge 0$. Let $K(\epsilon) = \{1{:}\, k_\epsilon\}$. Since $H_{\rm up}^k$ is nonincreasing, we have $H_{\rm up}^k \ge \epsilon$ for all $k \in K(\epsilon)$.

(ii) Our aim is to bound m from above, so suppose $m > 1$. To unburden notation, let $\phi_l = \check{h}^{k'(l)}$ for $l = 0{:}\, m-1$ and $t_l = f_{\rm low}^{k(l)}$ for $l = 0{:}\, m$. Recall from our remarks that we have $t_0 < \ldots < t_m \le f^*$, $\phi_l \le h$, $\phi_l(t_l) > \mu h(t_l) > 0$ and $\phi_l(t_{l+1}) = 0$ for $l = 0{:}\, m-1$. By Lemma 5.1, each ϕ_l is nonincreasing, convex and Lipschitzian with constant 1, and $\phi_l(t) > 0$ for all $t < t_{l+1}$; hence its right derivative ϕ_l' is negative for such t, e.g., $t = t_l$.

(iii) Let $\delta_l = -\phi_l(t_l)\phi_l'(t_l)$ for $l = 0{:}\, m-1$. Since $\phi_l(t_l) + \phi_l'(t_l)(t_{l+1} - t_l) \le \phi_l(t_{l+1}) = 0$, we have $\delta_l \ge \phi_l^2(t_l)/(t_{l+1} - t_l) > 0$. Fix $l < m-1$. Then $0 < -\phi_{l+1}'(t_{l+1}) \le (\phi_{l+1}(t_l) - \phi_{l+1}(t_{l+1}))/(t_{l+1} - t_l)$ by a secant property of convex functions on $\mathbb{R}$. Moreover, $\phi_{l+1}(t_l) \le h(t_l)$ and $\phi_l(t_l) > \mu h(t_l)$ imply $\phi_l(t_l) > \mu \phi_{l+1}(t_l)$. Hence

$$\frac{\delta_{l+1}}{\delta_l} \le \frac{(\phi_{l+1}(t_l) - \phi_{l+1}(t_{l+1}))\phi_{l+1}(t_{l+1})}{\phi_l^2(t_l)} \le \left(1 - \frac{\phi_{l+1}(t_{l+1})}{\phi_{l+1}(t_l)}\right) \frac{\phi_{l+1}(t_{l+1})}{\phi_{l+1}(t_l)} \frac{1}{\mu^2} \le \frac{1}{4\mu^2}.$$

We conclude that $\delta_{l+1} \le \delta_l/4\mu^2$ for $l = 0{:}\, m-2$.

(iv) We have $f^* - f^1_{\rm low} \le 2L_f \operatorname{diam}(S)$ and $h(f^1_{\rm low}) \le H(x^1; f^1_{\rm low}) \le L_H \operatorname{diam}(S)$. Indeed, let x^* solve (5.1). Then $f^* = f(x^*) \le f(x^1) + L_f|x^* - x^1|$, $|x^* - x^1| \le \operatorname{diam}(S)$ and $F(x^1) \le F(x^*) + L_F|x^* - x^1| \le L_F \operatorname{diam}(S)$. Similarly, if $\check{x}$ solves (5.2) for $k = 1$ then $f^1_{\rm low} = \check{f}^1_{\rm min} = f^1(\check{x}) = f(x^1) + \langle g_f(x^1), \check{x} - x^1\rangle \ge f(x^1) - L_f \operatorname{diam}(S)$ by the Cauchy-Schwarz inequality. Recall that $H(x^1; f^1_{\rm low}) = \max\{f(x^1) - f^1_{\rm low}; F(x^1)\}$ and $L_H = \max\{L_f, L_F\}$, and combine the preceding inequalities.

(v) Let $V = L_H \operatorname{diam}(S)$. By (ii) and (iv), $\phi_0(t_0) = \check{h}^1(f^1_{\rm low}) \le h(f^1_{\rm low}) \le V$. Clearly, $|\phi_0'(t_0)| \le 1$ from Lemma 5.1. Therefore (ii) yields $\delta_{m-1} \le V/(4\mu^2)^{m-1}$, so that either $\phi_{m-1}(t_{m-1}) \le (2\mu)^{1-m}V$ or $|\phi'_{m-1}(t_{m-1})| \le (2\mu)^{1-m}$. Since $\phi_{m-1}(t_m) = 0$, ϕ_{m-1} is nonincreasing and convex, and $f^* - 2V \le t_0 < \ldots < t_m \le f^*$ by (ii) and (iv), in the second case we have $\phi_{m-1}(t_{m-1}) \le |\phi'_{m-1}(t_{m-1})|(t_m - t_{m-1}) \le (2\mu)^{1-m}2V$. Thus in both cases $\phi_{m-1}(t_{m-1}) \le 4\mu V/(2\mu)^m$. But (5.4) gives $\phi_{m-1}(t_{m-1}) > \mu H^{k(m)}_{\rm up} \ge \mu\epsilon$, so

$$m \le \ln(4V/\epsilon)/\ln(2\mu). \tag{5.6}$$

This bound also holds if $m \le 1$, since $0 < \epsilon \le V$ and $1/2 < \mu < 1$.

(vi) Fix $l \le m$. Let $\hat{k} = k(l+1) - 1$ if $l < m$, $\hat{k} = k_\epsilon$ if $l = m$. After Step 4 sets $f^{k(l)}_{\rm low}$, we have $\Delta^k \ge (1-\mu)H^k_{\rm up} \ge (1-\mu)\epsilon$ for iterations $k = k(l){:}\,\hat{k}$, which may be treated as the first $\hat{k} - k(l) + 1$ iterations of Algorithm 2.1 minimizing $H(\cdot; f^{k(l)}_{\rm low})$ over S. Therefore, since $\Delta^{\hat{k}} \ge (1-\mu)\epsilon$, L_H is the Lipschitz constant of $H(\cdot; f^{k(l)}_{\rm low})$, $V = L_H \operatorname{diam}(S)$ and $\hat{k} - k(l) + 1 = |K^l \cap K(\epsilon)|$, we deduce from Lemma 3.4 that $|K^l \cap K(\epsilon)| \le (V/(1-\mu)\epsilon)^2/\kappa^2(1-\kappa^2)$ for $l = 0{:}\, m$. Combining these estimates with (5.6), we get $k_\epsilon = \sum_{l=0}^m |K^l \cap K(\epsilon)| \le [1 + \ln(4V/\epsilon)/\ln(2\mu)](V/(1-\mu)\epsilon)^2/\kappa^2(1-\kappa^2)$. Since $1/2 < \mu < 1$, we have $k_\epsilon \le c(\kappa,\mu)(V/\epsilon)^2 \ln(8V/\epsilon)$, and this implies the required estimate. □

We should add that the above proof is a modification of one given in [LNN91] for another level method (that enjoys (5.5a) with $c(\kappa,\mu)$ multiplied by 2).

Theorem 5.4. *If the algorithm does not terminate then $f(x^k_{\rm rec}) \to f^*$, $\max\{F(x^k_{\rm rec}), 0\} \to 0$ and $f^k_{\rm low} \uparrow f^*$ as $k \to \infty$. Moreover, the sequence $\{x^k_{\rm rec}\}$ converges to the optimal set of (5.1).*

Proof. Since $f^k_{\rm low} \le f^{k+1}_{\rm low} \le f^*$, $0 \le H^{k+1}_{\rm up} \le H^k_{\rm up} = H(x^k_{\rm rec}; f^k_{\rm low})$ and $h(f^k_{\rm low}) \le H(x^k_{\rm rec}; f^k_{\rm low})$ for all k, we have $f^k_{\rm low} \uparrow f^\infty_{\rm low} \le f^*$, $H(x^k_{\rm rec}; f^k_{\rm low}) \downarrow 0$ from Theorem 5.3, $h(f^\infty_{\rm low}) = 0$ and $f^\infty_{\rm low} = f^*$ by Lemma 5.1. Let $x^\infty \in S$ be any accumulation point of $\{x^k_{\rm rec}\}$ (which lies in the compact S). Then $H(x^\infty; f^*) = 0$ by the continuity of H, i.e., x^∞ solves (5.1). □

Step 5 may incorporate the additional tests of Step 3' from §4 for entering Step 6, e.g., $\Delta^k \le \hat{\kappa}\Delta^{k''(l')}$ or $H^k_{\rm up} - \check{h}^k(f^k_{\rm low}) \le \hat{\kappa}\Delta^{k''(l')}$ for $k''(l')$ generated like $k(l)$ in Algorithm 2.1 (i.e., Step 0 sets $k''(0) = 1$ and $l' = 0$, Step 4 sets $k''(l') = k$, whereas Step 6 sets $k''(l'+1) = k$ and increases l' by 1). Invoking Theorem 4.1 instead of Lemma 3.4 in the proof of Theorem 5.3, one may obtain (5.5) with $c(\kappa,\mu)$ replaced by $c(\kappa,\hat{\kappa},\mu) = c_1(\kappa,\hat{\kappa})c_2(\mu)$, where $c_2(\mu) = 1/\ln(2\mu)(1-\mu)^2$ and $c_1(\kappa,\hat{\kappa})$ is given by the various cases of (5.5). Incidentally, although $H^k_{\rm low} = 0$ need not equal $\check{h}^k(f^k_{\rm low})$ at Step 6, we always have $\Delta^k \le H^k_{\rm up} \le H(x^1_{\rm rec}; f^1_{\rm low}) \le L_H \operatorname{diam}(S)$ (see (iv) in the proof of Theorem 5.3), as required in the proof of Theorem 4.1.

Let us now consider the case of $F = \max_{i=1:n} F_i$, i.e., n inequality constraints in

$$f^* = \min\{\, f(x) : x \in S,\ F_i(x) \le 0,\ i = 1{:}n \,\},$$

where F_i are convex functions with subgradients $g_{F_i}(x) \in \partial F_i(x)$ satisfying $|g_{F_i}(x)| \le L_F$ for all $x \in S$. Then we may employ more refined models $\check{F}^k = \max_{i=1:n} \max_{j \in J_i^k} F_i^j$ with $F_i^j = F_i(x^j) + \langle g_{F_i}(x^j), \cdot - x^j \rangle$ and $J_i^k \subset \{1{:}k\}$, $i = 1{:}n$. Replacing (5.3c) by the constraints $F_i^j(x) \le H_{\text{lev}}^k$ with Lagrange multipliers λ_{ij}^k, $j \in J_i^k$, $i = 1{:}n$, at Step 9 we may choose $J_i^{k+1} = J_{is}^k \cup \{k+1\}$, where $\{j \in J_i^k : \lambda_{ij}^k > 0\} \subset J_{is}^k \subset J_i^k$. Step 6 may set $x_c^k = x^j$ for some $j \in J_f^k$ or $j \in J_i^k$ and i such that $\check{H}^k(x^j; f_{\text{low}}^k) \ge H_{\text{up}}^k$. Since obvious extensions of Lemmas 3.1 and 3.2 are easy to prove, we deduce that all the preceding efficiency results remain true.

To save storage and work per iteration, our algoritm may include separate aggregate linearizations $\tilde{f}^{k-1}$ and $\tilde{F}^{k-1}$ in $\check{f}^k$ and $\check{F}^k$ as in §4. Thus we may augment (5.3) with the constraints $\tilde{f}^{k-1}(x) \le H_{\text{lev}}^k + f_{\text{low}}^k$ and $\tilde{F}^{k-1}(x) \le H_{\text{lev}}^k$, find their Lagrange multipliers $\lambda_{\tilde{f}}^k$ and $\lambda_{\tilde{F}}^k$, and normalize each group of multipliers separately to define $\hat{f}^k$ and $\hat{F}^k$ as in (4.4) (e.g., with $\tilde{\lambda}_{fj}^k \ge 0$, $\tilde{\lambda}_f^k \ge 0$, $\sum_{j \in J_f^k} \tilde{\lambda}_{fj}^k + \tilde{\lambda}_f^k = 1$). Again, Step 9 may choose any $J_{fs}^k \subset J_f^k$ and $J_{Fs}^k \subset J_F^k$ (possibly $J_{fs}^k = J_{Fs}^k = \emptyset$), without influencing the efficiency estimates.

6 A method for saddle-point seeking

We shall give an extension of our method for solving the convex-concave saddle-point problem

$$\text{find} \quad (y^*, z^*) \in S_y \times S_z \quad \text{such that} \quad \max_{S_z} \Phi(y^*, \cdot) = \Phi(y^*, z^*) = \min_{S_y} \Phi(\cdot, z^*) \tag{6.1}$$

under the following assumptions. S_y and S_z are nonempty compact convex subsets of $\mathbb{R}^{N_y}$ and $\mathbb{R}^{N_z}$ respectively, $\Phi(\cdot, z)$ is a convex function Lipschitzian on S_y with constant L_y for each $z \in S_z$, $\Phi(y, \cdot)$ is a concave function Lipschitzian on S_z with constant L_z for each $y \in S_y$, and for each $(y, z) \in S_y \times S_z$ we can compute $\Phi(y, z)$ and a subgradient $\Phi'_y(y, z) \in \partial_y \Phi(y, z)$ of $\Phi(\cdot, z)$ at y and a supergradient $\Phi'_z(y, z) \in \partial_z \Phi(y, z)$ of $\Phi(y, \cdot)$ at z such that $|\Phi'_y(y, z)| \le L_y$ and $|\Phi'_z(y, z)| \le L_z$. (Here $-\Phi'_z(y, z)$ is a subgradient of the convex function $-\Phi(y, \cdot)$ at z.)

It will be convenient to identify the product $\mathbb{R}^{N_y} \times \mathbb{R}^{N_z}$ with $\mathbb{R}^N$ for $N = N_y + N_z$. Letting $x = (y, z)$ denote a generic point of $\mathbb{R}^N$ with components $y \in \mathbb{R}^{N_y}$ and $z \in \mathbb{R}^{N_z}$, we have $|x|^2 = |y|^2 + |z|^2$ (all norms being Euclidean, there is no need for extra notation). In particular, let $S = S_y \times S_z$ denote the feasible set of (6.1).

Let $\phi(y) = \max_{S_z} \Phi(y, \cdot)$, $\psi(z) = \min_{S_y} \Phi(\cdot, z)$ and $f(x) = \phi(y) - \psi(z)$ for all $x = (y, z) \in S$. Clearly, ϕ is convex on S_y, ψ is concave on S_z, and $f \ge 0$ is convex on S. Since $\phi(y) \ge \Phi(y, z) \ge \psi(z)$ if $(y, z) \in S$, we see that $S^* = \operatorname{Arg\,min}_S f = \{x \in S : f(x) = 0\}$ is the set of saddle-points of (6.1) (we shall prove constructively that $S^* \ne \emptyset$). Given an $\epsilon \ge 0$, we say that $x = (y, z)$ is an ϵ-saddle-point for (6.1) if $x \in S$ and $f(x) \le \epsilon$. Thus ϵ-minimizers for f on S are of interest, but they cannot be found by Algorithm 2.1 when we cannot compute the usual linearizations of f. Owing to the special structure of f, we may, however, employ 'inexact' linearizations as follows.

Our algorithm will generate trial points $x^k = (y^k, z^k) \in S$ at which the 'inexact' linearizations $\phi^k(y) = \Phi(x^k) + \left\langle \Phi'_y(x^k), y - y^k \right\rangle$ and $\psi^k(z) = \Phi(x^k) + \left\langle \Phi'_z(x^k), z - z^k \right\rangle$ of ϕ

and ψ are computed. By construction, $\phi^k \le \Phi(\cdot, z^k) \le \phi$ and $\psi^k \ge \Phi(y^k, \cdot) \ge \psi$, although we need not have equalities at y^k and z^k as for the 'exact' linearizations considered before. At the kth iteration, f is approximated from below by $\check{f}^k(x) = \check{\phi}^k(y) - \check{\psi}^k(z)$, where $\check{\phi}^k = \max_{j \in J_y^k} \phi^j$ underestimates ϕ, $\check{\psi}^k = \min_{j \in J_z^k} \psi^j$ overestimates ψ, $k \in J_y^k \subset \{1\!:\!k\}$ and $k \in J_z^k \subset \{1\!:\!k\}$. Clearly, $\check{f}^k$ is convex and $\check{f}^k(x^k) \ge 0$, since $\check{\phi}^k(y^k) \ge \phi^k(y^k) = \Phi(x^k)$ and $\check{\psi}^k(z^k) \le \psi^k(z^k) = \Phi(x^k)$. Other useful properties of these approximations are summarized in the following result from [LNN91].

Lemma 6.1. *Let* $\check{\phi}^k_{\min} = \min_{S_y} \check{\phi}^k$, $\check{\psi}^k_{\max} = \max_{S_z} \check{\psi}^k$, *and let* μ^k_{yj}, $j \in J_y^k$, *and* μ^k_{zj}, $j \in J_z^k$, *denote the associated multipliers satisfying* $\mu^k_{yj} \ge 0$, $\sum_{j \in J_y^k} \mu^k_{yj} = 1$, $\min_{y \in S_y} \sum_{j \in J_y^k} \mu^k_{yj} \phi^j(y) = \check{\phi}^k_{\min}$, $\mu^k_{zj} \ge 0$, $\sum_{j \in J_z^k} \mu^k_{zj} = 1$ *and* $\max_{z \in S_z} \sum_{j \in J_z^k} \mu^k_{zj} \psi^j(z) = \check{\psi}^k_{\max}$. *Define* $x^k_{\rm rec} = (y^k_{\rm rec}, z^k_{\rm rec})$ *by* $y^k_{\rm rec} = \sum_{j \in J_y^k} \mu^k_{yj} y^j$ *and* $z^k_{\rm rec} = \sum_{j \in J_z^k} \mu^k_{zj} y^j$, *and let* $\check{f}^k_{\min} = \min_S \check{f}^k$. *Then* $\check{f}^k_{\min} = \check{\phi}^k_{\min} - \check{\psi}^k_{\max} \le 0$, $x^k_{\rm rec} \in S$ *and* $f(x^k_{\rm rec}) \le -\check{f}^k_{\min}$.

Proof. We have $\Phi(y, z^k_{\rm rec}) \ge \sum_j \mu^k_{zj} \Phi(y, z^j) \ge \sum_j \mu^k_{zj} \phi^j(y)$ for all $y \in S_y$, since $\Phi(y, \cdot)$ is concave and $\Phi(\cdot, z^j)$ are convex. Taking minima over S_y, we get $\psi(z^k_{\rm rec}) \ge \check{\phi}^k_{\min}$. Similarly, by the convexity of $\Phi(\cdot, z)$ and the concavity of $\Phi(y^j, \cdot)$, $\Phi(y^k_{\rm rec}, z) \le \sum_j \mu^k_{yj} \Phi(y^j, z) \le \sum_j \mu^k_{yj} \psi^j(z)$ for all $z \in S_z$ implies $\phi(y^k_{\rm rec}) \le \check{\psi}^k_{\max}$. Hence $f(x^k_{\rm rec}) = \phi(y^k_{\rm rec}) - \psi(z^k_{\rm rec}) \le \check{\psi}^k_{\max} - \check{\phi}^k_{\min} = -\check{f}^k_{\min}$. Next, $y^k_{\rm rec} \in S_y$, $z^k_{\rm rec} \in S_z$ and $x^k_{\rm rec} \in S$ by convexity, so $f(x^k_{\rm rec}) \ge 0$. □

In view of Lemma 6.1, we wish to find a model $\check{f}^k$ with a small $\check{f}^k_{\min}$. To this end, the algorithm sets $x^{k+1} = \arg\min\{|x - x^k_c| : x \in S,\ \check{f}^k(x) \le f^k_{\rm lev}\}$, where $x^k_c \in \{x^j\}_{j=1}^k$ and $f^k_{\rm lev} = -\kappa \Delta^k$, with $\Delta^k = -\check{f}^{k(l)}_{\min}$ for some $k(l) \le k$ and $0 < \kappa < 1$. Thus $x^{k+1} = (y^{k+1}, z^{k+1})$ solves the problem

$$\begin{aligned} &\text{minimize} && |x - x^k_c|^2/2 \quad \text{over all } x \in S && \text{(6.2a)} \\ &\text{satisfying} && f^{ij}(x) \le f^k_{\rm lev} \quad \text{for} \quad (i,j) \in J^k, && \text{(6.2b)} \end{aligned}$$

where $f^{ij}(x) = \phi^i(y) - \psi^j(z) = \Phi(x^i) - \Phi(x^j) + \left\langle \Phi'_y(x^i), y - y^i \right\rangle - \langle \Phi'_z(x^j), z - z^j \rangle$ and $J^k = J_y^k \times J_z^k$ correspond to the representation $\check{f}^k = \max_{(i,j) \in J^k} f^{ij} = \max_{j \in J_y^k} \phi^j - \min_{j \in J_z^k} \psi^j$.

We may now state our proximal level algorithm.

Algorithm 6.2

Step 0 (*Initialization*). Select an initial point $x^1 \in S$, a final optimality tolerance $\epsilon_{\rm opt} \ge 0$ and a level parameter $0 < \kappa < 1$. Set $k = 1$, $x^1_c = x^1$ and $J^1 = J^1_y \times J^1_z = \{1\} \times \{1\}$. Set $\Delta^1 = -\check{f}^1_{\min}$ and find the corresponding $x^1_{\rm rec} = (y^1_{\rm rec}, z^1_{\rm rec})$ as in Lemma 6.1. Set the counters $l = 0$ and $k(0) = 1$ ($k(l)$ will denote the iteration number of the lth decrease of Δ^k).

Step 1 (*Level update*). Set $f^k_{\rm lev} = -\kappa \Delta^k$.

Step 2 (*Stopping criterion*). If $\Delta^k \le \epsilon_{\rm opt}$, terminate; otherwise, continue.

Step 3 (*Level feasibility check*). If (6.2) is feasible, go to Step 5; otherwise, continue.

Step 4 (*Update lower bound*). Find $\check{f}^k_{\min}$ and the corresponding $x^k_{\rm rec}$ as in Lemma 6.1. Set $\hat{\Delta}^k = -\check{f}^k_{\min}$ and $\Delta^k = \hat{\Delta}^k$. Choose $x^k_c \in \{x^j : (j,j) \in J^k\}$ (e.g., $x^k_c = x^k$). Set $k(l+1) = k$ and increase l by 1. Go to Step 1.

Step 5 (*Projection*). Find the solution x^{k+1} of (6.2) and its multipliers λ_{ij}^k such that the set $\hat{J}^k = \{(i,j) \in J^k : \lambda_{ij}^k > 0\}$ satisfies $|\hat{J}^k| \le N$.

Step 6 (*Objective evaluation*). Calculate $\Phi(x^{k+1})$, $\Phi_y'(x^k)$ and $\Phi_z'(x^k)$.

Step 7 (*Selection*). Select $J_s^k = J_{ys}^k \times J_{zs}^k \subset J^k$ such that $\hat{J}^k \subset J_s^k$. Set $J^{k+1} = J_y^{k+1} \times J_z^{k+1} = (J_{ys}^k \cup \{k+1\}) \times (J_{zs}^k \cup \{k+1\})$, $x_c^{k+1} = x_c^k$, $x_{\rm rec}^{k+1} = x_{\rm rec}^k$, $\Delta^{k+1} = \Delta^k$ and, if $k > k(l)$, $\hat{\Delta}^k = \Delta^k$. Increase k by 1 and go to Step 1.

Clearly, our method may be considered as a modification of Algorithm 2.1 obtained by setting $f_{\rm up}^k = f^* = \min_S f = 0$ and $f_{\rm low}^k = \check{f}_{\rm min}^{k(l)}$ at Step 1. With this notation, the various remarks and relations of §2 apply to the present case. For instance, (1.3) and (2.4) remain true, whereas (2.2) is replaced by the fact that $f(x_{\rm rec}^k) \le \hat{\Delta}^k$ (from Steps 0 and 4), i.e., $x_{\rm rec}^k$ is a $\hat{\Delta}^k$-saddle-point for (6.1). Hence the efficiency of our method will be analyzed below as in §3. To this end, observe that the gradients $\nabla f^{ij} = (\Phi_y'(x^i), -\Phi_z'(x^j))$ of the linearizations f^{ij} satisfy $|\nabla f^{ij}| \le L_f$ with $L_f = (L_y^2 + L_z^2)^{1/2}$; hence L_f may serve as the Lipschitz constant for f^{ij} (as well as for $\check{f}^k$ and f itself, although we do not really need this property). We also note that $\operatorname{diam}(S) = [\operatorname{diam}(S_y)^2 + \operatorname{diam}(S_z)^2]^{1/2}$, and recall that $S^k = \{x \in S : \check{f}^k(x) \le f_{\rm lev}^k\}$.

Theorem 6.3. *The following efficiency estimate holds for any $\epsilon > 0$:*

$$k > c(\kappa)(\operatorname{diam}(S) L_f/\epsilon)^2 \quad \Rightarrow \quad f(x_{\rm rec}^k) \le \hat{\Delta}^k < \epsilon, \tag{6.3a}$$

$$c(\kappa) = 1/\kappa^2(1-\kappa^2) \quad \text{and} \quad \min c(\cdot) = c(1/\sqrt{2}) = 4. \tag{6.3b}$$

Proof. We claim that Lemma 3.1 remains true. Indeed, if $k > k(l)$, let $j = k$. Then $f^{jj} \le \check{f}^k$ from $(j,j) \in J^k$, and $f^{jj}(x^j) = \Phi(x^j) - \Phi(x^j) = 0$. Since $x^{k+1} \in S^k$, we have $f^{jj}(x^{k+1}) = f^{jj}(x^j) + \left\langle \nabla f^{jj}(x^j), x^{k+1} - x^j \right\rangle \le \check{f}^k(x^{k+1}) \le f_{\rm lev}^k$, so $L_f |x^{k+1} - x^j| \ge |\nabla f^{jj}(x^j)||x^{k+1} - x^j| \ge \kappa \Delta^k$ due to $|\nabla f^{jj}| \le L_f$, the Cauchy-Schwarz inequality and $f_{\rm lev}^k = -\kappa \Delta^k$. If $k = k(l)$, let $(j,j) \in J^k$ above be such that $x_c^k = x^j$ at Step 4. Having established Lemma 3.1, it is easy to verify the remaining Lemmas 3.2–3.4, which imply the desired result. □

We observe that Step 3 may be modified like Step 3' of §4, and Theorem 4.1 may be verified if one notes that $\Delta^1 \le L_f \operatorname{diam}(S)$. (Indeed, $\Delta^1 = -\min_S f^{11} = f^{11}(x^1) - \min_S \langle \nabla f^{11}, \cdot - x^1 \rangle \le |\nabla f^{11}| \operatorname{diam}(S)$ due to $f^{11}(x^1) = 0$ and the Cauchy-Schwarz inequality.) Moreover, the following analogue of Theorem 3.7 holds: $\hat{\Delta}^k \downarrow 0$ and $f(x_{\rm rec}^k) \downarrow 0$ as $k \to \infty$ and every accumulation point x^∞ of $\{x_{\rm rec}^k\}$ solves (6.1), i.e., $x^\infty \in S$ and $f(x^\infty) = 0$, since $f \ge 0$ is continuous (at least one such point exists because $\{x_{\rm rec}^k\} \subset S$, a compact set).

Instead of (6.2), we may solve the following problem with artificial variables y_0 and z_0:

$$\text{minimize} \quad |y - y_c^k|^2/2 + |z - z_c^k|^2/2 \quad \text{over all } (y,z) \in S,\ y_0 \in \mathbb{R},\ z_0 \in \mathbb{R} \tag{6.4a}$$

$$\text{satisfying} \quad y_0 + z_0 \le f_{\rm lev}^k,\ \phi^j(y) \le y_0 \text{ for } j \in J_y^k,\ -\psi^j(z) \le z_0 \text{ for } j \in J_z^k. \tag{6.4b}$$

Denote the Lagrange multipliers of (6.4) by λ_f^k, λ_{yj}^k, $j \in J_y^k$, λ_{zj}^k, $j \in J_z^k$. Due to the structure of (6.4), there exist multipliers such that the sets $\hat{J}_y^k = \{j \in J_y^k : \lambda_{yj}^k > 0\}$ and $\hat{J}_z^k = \{j \in J_z^k : \lambda_{zj}^k > 0\}$ satisfy $|\hat{J}_y^k| \le N_y + 1$ and $|\hat{J}_z^k| \le N_z + 1$. (Such multipliers may be found by QP methods when S is described by finitely many linear inequalities.) Hence selection may require storage of order $2N_y^2 + 2N_z^2$ (for ϕ^j, y^j, ψ^j and z^j).

To save storage, we may aggregate linearizations and corresponding points as follows. Suppose $(\tilde\phi^{k-1},\tilde y^{k-1})\in \mathrm{co}\{(\phi^j,y^j)\}_{j=1}^{k-1}$ and $(\tilde\psi^{k-1},\tilde z^{k-1})\in \mathrm{co}\{(\psi^j,z^j)\}_{j=1}^{k-1}$. Append to (6.4) the constraints $\tilde\phi^{k-1}(y)\le y_0$ and $-\tilde\psi^{k-1}(z)\le z_0$ with multipliers $\lambda_{\tilde\phi}^k$ and $\lambda_{\tilde\psi}^k$. As in §4, we may find 'normalized' nonnegative multipliers satisfying $\sum_{j\in J_y^k}\tilde\lambda_{yj}^k+\tilde\lambda_{\tilde\phi}^k=1$, $\sum_{j\in J_z^k}\tilde\lambda_{zj}^k+\tilde\lambda_{\tilde\psi}^k=1$, and set $(\tilde\phi^k,\tilde y^k)=\sum_{j\in J_y^k}\tilde\lambda_{yj}^k(\phi^j,y^j)+\tilde\lambda_{\tilde\phi}^k(\tilde\phi^{k-1},\tilde y^{k-1})$, $(\tilde\psi^k,\tilde z^k)=\sum_{j\in J_z^k}\tilde\lambda_{zj}^k(\psi^j,z^j)+\tilde\lambda_{\tilde\psi}^k(\tilde\psi^{k-1},\tilde z^{k-1})$. Clearly, $(\tilde\phi^k,\tilde y^k)\in\mathrm{co}\{(\phi^j,y^j)\}_{j=1}^k$ and $(\tilde\psi^k,\tilde z^k)\in\mathrm{co}\{(\psi^j,z^j)\}_{j=1}^k$ (as convex combinations of convex combinations). In other words, we use $\check\phi^k=\max\{\tilde\phi^{k-1},\max_{j\in J_y^k}\phi^j\}$ and $\check\psi^k=\min\{\tilde\psi^{k-1},\min_{j\in J_z^k}\psi^j\}$, so an extension of Lemma 6.1 is required, in which we represent $(\tilde\phi^{k-1},\tilde y^{k-1},1)=\sum_{i=1}^{k-1}\hat\lambda_{yi}^{k-1}(\phi^i,y^i,1)$ and $(\tilde\psi^{k-1},\tilde z^{k-1},1)=\sum_{i=1}^{k-1}\hat\lambda_{zi}^{k-1}(\psi^i,z^i,1)$ for some $\hat\lambda_{yi}^{k-1}\ge 0$ and $\hat\lambda_{zi}^{k-1}\ge 0$.

Lemma 6.4. *Let $\check\phi_{\min}^k=\min_{S_y}\check\phi^k$, $\check\psi_{\max}^k=\max_{S_z}\check\psi^k$, and let μ_{yj}^k, $j\in J_y^k$, $\tilde\mu_y^k$ and μ_{zj}^k, $j\in J_z^k$, $\tilde\mu_z^k$ denote the associated multipliers satisfying $\mu_{yj}^k\ge 0$, $\tilde\mu_y^k\ge 0$, $\sum_{j\in J_y^k}\mu_{yj}^k+\tilde\mu_y^k=1$, $\min_{y\in S_y}\sum_{j\in J_y^k}\mu_{yj}^k\phi^j(y)+\tilde\mu_y^k\tilde\phi^{k-1}(y)=\check\phi_{\min}^k$, $\mu_{zj}^k\ge 0$, $\tilde\mu_z^k\ge 0$, $\sum_{j\in J_z^k}\mu_{zj}^k+\tilde\mu_z^k=1$, $\max_{z\in S_z}\sum_{j\in J_z^k}\mu_{zj}^k\psi^j(z)+\tilde\mu_z^k\tilde\psi^{k-1}(z)=\check\psi_{\max}^k$. Define $x_{\mathrm{rec}}^k=(y_{\mathrm{rec}}^k,z_{\mathrm{rec}}^k)$ by $y_{\mathrm{rec}}^k=\sum_{j\in J_y^k}\mu_{yj}^k y^j+\tilde\mu_y^k\tilde y^{k-1}$ and $z_{\mathrm{rec}}^k=\sum_{j\in J_z^k}\mu_{zj}^k z^j+\tilde\mu_z^k\tilde z^{k-1}$, and let $\check f_{\min}^k=\min_S\check f^k$. Then $\check f_{\min}^k=\check\phi_{\min}^k-\check\psi_{\max}^k\le 0$, $x_{\mathrm{rec}}^k\in S$ and $f(x_{\mathrm{rec}}^k)\le-\check f_{\min}^k$.*

Proof. As in the proof of Lemma 6.1, $\Phi(y,\tilde z^{k-1})\ge\sum_i\hat\lambda_{zi}^{k-1}\Phi(y,z^i)\ge\sum_i\hat\lambda_{zi}^{k-1}\phi^i(y)=\tilde\phi^{k-1}(y)$ and $\Phi(y,z_{\mathrm{rec}}^k)\ge\sum_j\mu_{zj}^k\Phi(y,z^j)+\tilde\mu_z^k\Phi(y,\tilde z^{k-1})\ge\sum_j\mu_{zj}^k\phi^j(y)+\tilde\mu_z^k\tilde\phi^{k-1}(y)$ for all $y\in S_y$. Getting a similar relation for $\Phi(y_{\mathrm{rec}}^k,z)$, one may complete the proof as before. □

We conclude that, owing to aggregation, our method will retain its efficiency estimates even when only $\tilde\phi^{k-1}$, $\tilde\psi^{k-1}$, $\tilde y^{k-1}$ and $\tilde z^{k-1}$ are stored and updated recursively (i.e., $J_{ys}^k=J_{zs}^k=\emptyset$).

7 A method for variational inequalities

We give a modification of our method for the monotone variational inequality problem

$$\text{find}\quad x^*\in S\quad\text{such that}\quad\langle\Psi(x),x-x^*\rangle\ge 0\quad\forall x\in S \tag{7.1}$$

under the following assumptions. S is a nonempty compact convex subset of $\mathbb{R}^N$ and Ψ is a monotone bounded-valued operator on S, i.e., $\langle\Psi(x)-\Psi(x'),x-x'\rangle\ge 0$ if $x,x'\in S$, and $\sup_S|\Psi(\cdot)|<\infty$. (See, e.g., [LNN91] for conditions, such as maximal monotonicity or continuity of Ψ, under which (7.1) is equivalent to the classical variational inequality problem of finding $x^*\in S$ such that $\langle\Psi(x^*),x-x^*\rangle\ge 0\ \forall x\in S$.)

We associate with (7.1) the improvement function $f(x)=\sup_S\langle\Psi(\cdot),x-\cdot\rangle$ defined for $x\in S$. Clearly, $f\ge 0$ is convex, and $S^*=\{x\in S: f(x)=0\}$ is the set of solutions to (7.1) (we shall prove constructively that $S^*\ne\emptyset$). Given an $\epsilon\ge 0$, we say that x is an ϵ-solution to (7.1) if $x\in S$ and $f(x)\le\epsilon$. Thus ϵ-minimizers for f on S are of interest, but they cannot be found by Algorithm 2.1 when we cannot compute the usual linearizations of

f. The situation is similar to that of §6, so we shall again exploit the special structure of f that allows 'inexact' linearizations.

Our algorithm will generate trial points $x^k \in S$ at which the 'inexact' linearizations $f^k = \langle \Psi(x^k), \cdot - x^k \rangle$ of f are computed. By construction, $f^k \leq f$, although $f^k(x^k) = 0$ need not equal $f(x^k)$. At the kth iteration, f is approximated from below by $\check{f}^k(x) = \max_{j \in J^k} f^j$, where $k \in J^k \subset \{1\colon k\}$. Clearly, $\check{f}^k$ is convex and $\check{f}^k(x^k) \geq 0$, since $f^k(x^k) = 0$. Another useful property of $\check{f}^k$ has been established in [LNN91].

Lemma 7.1. *Let* $\check{f}^k_{\min} = \min_S \check{f}^k$, *and let* μ^k_j, $j \in J^k$, *denote the associated multipliers satisfying* $\mu^k_j \geq 0$, $\sum_{j \in J^k} \mu^k_j = 1$, *and* $\min_{x \in S} \sum_{j \in J^k} \mu^k_j f^j(x) = \check{f}^k_{\min}$. *Let* $x^k_{\rm rec} = \sum_{j \in J^k} \mu^k_j x^j$. *Then* $\check{f}^k_{\min} \leq 0$, $x^k_{\rm rec} \in S$ *and* $f(x^k_{\rm rec}) \leq -\check{f}^k_{\min}$.

Proof. Let $x \in S$. We have $\langle \Psi(x), x - x^j \rangle \geq \langle \Psi(x^j), x - x^j \rangle$ by the monotonicity of Ψ, so $\langle \Psi(x), x - x^k_{\rm rec} \rangle = \sum_j \mu^k_j \langle \Psi(x), x - x^j \rangle \geq \sum_j \mu^k_j \langle \Psi(x^j), x - x^j \rangle = \sum_j \mu^k_j f^j(x) \geq f^k_{\min}$. Hence $0 \leq f(x^k_{\rm rec}) = \sup_{x \in S} \langle \Psi(x), x^k_{\rm rec} - x \rangle \leq -f^k_{\min}$, where $x^k_{\rm rec} \in S$ by convexity. □

Since Lemmas 6.1 and 7.1 are analogous, it is not suprising that our method is similar to Algorithm 6.2.

Algorithm 7.2

Step 0 (*Initialization*). Select an initial point $x^1 \in S$, a final optimality tolerance $\epsilon_{\rm opt} \geq 0$ and a level parameter $0 < \kappa < 1$. Set $k = 1$, $x^1_c = x^1$ and $J^1 = \{1\}$. Set $\Delta^1 = -\check{f}^1_{\min}$ and find the corresponding $x^1_{\rm rec}$ as in Lemma 7.1. Set the counters $l = 0$ and $k(0) = 1$ ($k(l)$ will denote the iteration number of the lth decrease of Δ^k).

Step 1 (*Level update*). Set $f^k_{\rm lev} = -\kappa \Delta^k$.

Step 2 (*Stopping criterion*). If $\Delta^k \leq \epsilon_{\rm opt}$, terminate; otherwise, continue.

Step 3 (*Level feasibility check*). If (2.1) is feasible, go to Step 5; otherwise, continue.

Step 4 (*Update lower bound*). Find $\check{f}^k_{\min}$ and the corresponding $x^k_{\rm rec}$ as in Lemma 7.1. Set $\hat{\Delta}^k = -\check{f}^k_{\min}$ and $\Delta^k = \hat{\Delta}^k$. Choose $x^k_c \in \{x^j : j \in J^k\}$ (e.g., $x^k_c = x^k$). Set $k(l+1) = k$ and increase l by 1. Go to Step 1.

Step 5 (*Projection*). Find the solution x^{k+1} of (2.1) and its multipliers λ^k_j such that the set $\hat{J}^k = \{j \in J^k : \lambda^k_j > 0\}$ satisfies $|\hat{J}^k| \leq N$.

Step 6 (*Objective evaluation*). Calculate $\Psi(x^{k+1})$.

Step 7 (*Selection*). Select $J^k_s \subset J^k$ such that $\hat{J}^k \subset J^k_s$. Set $J^{k+1} = J^k_s \cup \{k+1\}$, $x^{k+1}_c = x^k_c$, $x^{k+1}_{\rm rec} = x^k_{\rm rec}$, $\Delta^{k+1} = \Delta^k$ and, if $k > k(l)$, $\hat{\Delta}^k = \Delta^k$. Increase k by 1 and go to Step 1.

For the reader's convenience, we repeat our remarks from §6. Again, our method may be considered as a modification of Algorithm 2.1 obtained by setting $f^k_{\rm up} = 0$ and $f^k_{\rm low} = \check{f}^{k(l)}_{\min}$ at Step 1. Therefore, (1.3) and (2.4) remain true, whereas (2.2) is replaced by the fact that $f(x^k_{\rm rec}) \leq \hat{\Delta}^k$ (from Steps 0 and 4), i.e., $x^k_{\rm rec}$ is a $\hat{\Delta}^k$-solution for (7.1). We also see that $L_f = \sup_S |\Psi(\cdot)|$ may serve as the Lipschitz constant for our linearizations f^j that satisfy $f^j(x^j) = 0$. These properties enable us to verify Lemma 3.1 as in the proof of Theorem 6.3. Therefore, it is easy to check that Theorem 6.3 holds for Algorithm 7.2 as well. Moreover, as

in §6, we observe that if Step 3 is replaced by Step 3' of §4 then Theorem 4.1 holds. Finally, as in Theorem 3.7, we have $\hat{\Delta}^k \downarrow 0$ and $f(x^k_{\rm rec}) \downarrow 0$ as $k \to \infty$ and every accumulation point x^∞ of $\{x^k_{\rm rec}\}$ solves (7.1), i.e., $x^\infty \in S$ and $f(x^\infty) = 0$, since $f \geq 0$ is closed (at least one such point exists because $\{x^k_{\rm rec}\} \subset S$, a compact set).

Let us now consider aggregation. Thus suppose, as in §4 and §6, that iteration k uses $\check{f}^k = \max\{\tilde{f}^{k-1}, \max_{j\in J^k} f^j\}$ and $\tilde{x}^{k-1} \in S$ such that $(\tilde{f}^{k-1}, \tilde{x}^{k-1}, 1) = \sum_{i=1}^{k-1} \hat{\lambda}_i^{k-1}(f^i, x^i, 1)$ for some $\hat{\lambda}_i^{k-1} \geq 0$. (One may let $(\tilde{f}^{k-1}, \tilde{x}^{k-1}) = (f^1, x^1)$ for $k = 1$.) Augmenting (4.4) with $\tilde{x}^k = \sum_{j\in J^k} \tilde{\lambda}_j^k x^j + \tilde{\lambda}_f^k \tilde{x}^{k-1}$, we see that $(\tilde{f}^k, \tilde{x}^k) \in {\rm co}\{(f^j, x^j)\}_{j=1}^k$, as required for the recursion. It remains to provide an extension of Lemma 7.1.

Lemma 7.3. *Let $\check{f}^k_{\rm min} = \min_S \check{f}^k$, and let μ_j^k, $j \in J^k$, and $\tilde{\mu}_f^k$ denote the associated multipliers satisfying $\mu_j^k \geq 0$, $\tilde{\mu}_f^k \geq 0$, $\sum_{j\in J^k} \mu_j^k + \tilde{\mu}_f^k = 1$, and $\min_{x\in S} \sum_{j\in J^k} \mu_j^k f^j(x) + \tilde{\mu}_f^k \tilde{f}^{k-1}(x) = \check{f}^k_{\rm min}$. Let $x^k_{\rm rec} = \sum_{j\in J^k} \mu_j^k x^j + \tilde{\mu}_f^k \tilde{x}^{k-1}$. Then $\check{f}^k_{\rm min} \leq 0$, $x^k_{\rm rec} \in S$ and $f(x^k_{\rm rec}) \leq -\check{f}^k_{\rm min}$.*

Proof. We have $\left\langle \Psi(x), x - \tilde{x}^{k-1} \right\rangle = \sum_i \hat{\lambda}_i^{k-1} \langle \Psi(x), x - x^i \rangle \geq \sum_i \hat{\lambda}_i^{k-1} \langle \Psi(x^i), x - x^i \rangle = \sum_i \hat{\lambda}_i^{k-1} f^i(x) = \tilde{f}^{k-1}(x)$ and $\left\langle \Psi(x), x - x^k_{\rm rec} \right\rangle = \sum_j \mu_j^k \langle \Psi(x), x - x^j \rangle + \tilde{\mu}_f^k \left\langle \Psi(x), x - \tilde{x}^{k-1} \right\rangle \geq \sum_j \mu_j^k f^j(x) + \tilde{\mu}_f^k \tilde{f}^{k-1}(x) \geq f^k_{\rm min}$ as in the proof of Lemma 7.1, which goes on as before. □

8 Conclusions

We have presented several implementable versions of the level methods of [LNN91] for nondifferentiable convex minimization, constrained optimization, convex-concave saddle-point problems and variational inequalities with monotone operators. We have shown that selection and aggregation of linearizations ensures bounded storage without worsening efficiency estimates. Preliminary numerical experience with the original methods of [LNN91] has been encouraging, although these methods suffer when their storage requirements become too large during calculations. We hope, therefore, that our approach will alleviate such difficulties without worsening practical efficiency.

Acknowledgment. This work was inspired by C. Lemarechal. It was done during my six month stay at INRIA, Rocquencourt, made possible by a grant from the French Ministry of Research and Technology.

References

[Kiw85] K.C. Kiwiel, Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics 1133, Springer-Verlag, Berlin, 1985.

[Kiw90] K.C. Kiwiel, Proximity control in bundle methods for convex nondifferentiable minimization, Math. Programming 46 (1990) 105–122.

[LNN91] C. Lemarechal, A.S. Nemirovskii and Yu. Nesterov, New variants of bundle methods, Research Report No. 1508, INRIA, Rocquencourt, 1991.

[NYu83] A.S. Nemirovskii and D.B. Yudin, Problem Complexity and Method Efficiency in Optimization. Wiley-Interscience, New York 1983.

ISSN 0249-6399