Preliminary computational experience with a descent level method for convex nondifferentiable optimization

U. Brännlund
Royal Institute of Technology, Stockholm, Sweden.
e-mail: uffe@math.kth.se

K.C. Kiwiel
System Research Institute, Warsaw, Poland.
e-mail: kiwiel@ibspan.waw.pl

P.O. Lindberg
Royal Institute of Technology, Stockholm, Sweden.
e-mail: pol@math.kth.se

Abstract

We report on numerical tests with our recently introduced descent level bundle method for convex minimization. The test problems include standard nonsmooth problems, eigenvalue problems, and Lagrangian duals of traveling salesman, capacitated lotsizing, hierarchical production planning and unit commitment problems.

Keywords

Nondifferentiable optimization, proximal bundle methods, level methods.

1 INTRODUCTION

We have recently introduced a descent bundle method Brännlund *et al.* (1995) for minimizing a (possibly nondifferentiable) convex function $f : \mathbb{R}^N \to \mathbb{R}$ over a nonempty closed convex set $S \subset \mathbb{R}^N$. We assume that at each $x \in S$ we can compute $f(x)$ and an arbitrary subgradient $g(x) \in \partial f(x)$.

At the kth iteration, having generated *linearizations* $f^j(\cdot) = f(y^j) + \langle g(y^j), \cdot - y^j \rangle$ of f at *trial points* $y^j \in S$ for $j = 1{:}\,k$, we approximate f from below by the piecewise linear

Research supported by the Swedish Research Council for Engineering Sciences, the Göran Gustafsson Foundation and the Polish State Committee for Scientific Research under Grant 8S50502206.

J. Doležal et al. (eds.), *System Modelling and Optimization*

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

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

where the *prox-center* x^k usually has the best f-value among the points $\{y^j\}_{j=1}^k$, and the *target level* $f^k_{\text{lev}} < f(x^k)$ is chosen to ensure $f^k_{\text{lev}} \to f^* := \inf_S f$ as $k \to \infty$. If a finite *lower bound* $f^k_{\text{low}} \leq f^*$ is known, then usually $f^k_{\text{lev}} = f(x^k) - \kappa_l \Delta^k$, where $0 < \kappa_l < 1$ and the *optimality gap* $\Delta^k = f(x^k) - f^k_{\text{low}}$ provides the *optimality estimate* $f(x^k) - f^* \leq \Delta^k$. Since $\check{f}^k \leq f$, if the feasible set $S^k = \{x \in S : \check{f}^k(x) \leq f^k_{\text{lev}}\}$ of (1) is empty then $f^k_{\text{lev}} < f^*$, so setting $f^k_{\text{low}} = f^k_{\text{lev}}$ reduces Δ^k by at least κ_l. Also f^k_{lev} is increased and y^{k+1} is recomputed if the *direction* $d^k = y^{k+1} - x^k$ is 'too large' relative to the *desired descent* $\delta^k = f(x^k) - f^k_{\text{lev}}$. A *descent* step to $x^{k+1} = y^{k+1}$ is taken if $f(y^{k+1}) \leq f(x^k) - \kappa_d \delta^k$, where $0 < \kappa_d < 1$ (i.e., if the actual descent is at least a fraction of the desired one). Otherwise, a *null* step $x^{k+1} = x^k$ provides a new linearization f^{k+1}.

The original level methods of Lemaréchal *et al.* (1995) require S to be *compact*. They are hardly implementable because they employ $J^k = \{1:k\}$ and $f^k_{\text{low}} = \min_S \check{f}^k$. In contrast, our method needs bounded storage and is globally convergent without any compactness assumptions.

2 THE DESCENT PROXIMAL LEVEL ALGORITHM

The trial point finding subproblem (1) may be formulated as the QP problem

$$\begin{array}{lll} \text{minimize} & |x - x^k|^2/2 & \text{over all } x \in S \\ \text{satisfying} & f^j(x) \leq f^k_{\text{lev}} & \text{for } j \in J^k. \end{array} \tag{2}$$

Algorithm 1

Step 0 (*Initiation*). Select an initial point $x^1 \in S$, a final optimality tolerance $\epsilon_{\text{opt}} \geq 0$, a multiplier bound $t_{\max} > 0$ and parameters $\kappa_d, \kappa_l, \kappa_\delta \in (0,1)$. Choose $f^1_{\text{low}} \leq f^*$ (*e.g.* , $f^1_{\text{low}} = -\infty$). Set $\Delta^1 = f(x^1) - f^1_{\text{low}}$. Set $\delta^1 = \kappa_l \Delta^1$ if $\Delta^1 < \infty$; otherwise choose $\delta^1 > 0$. Set $J^1 = \{1\}$. Set the counters $k = 1$, $l = 0$ and $k(0) = 1$.

Step 1 (*Level feasibility check*). Set $f^k_{\text{lev}} = f(x^k) - \delta^k$. If (2) is feasible, go to Step 3.

Step 2 (*Update lower bound*). Choose $f^k_{\text{low}} \in [f^k_{\text{lev}}, f^*]$ (*e.g.* , $f^k_{\text{low}} = f^k_{\text{lev}}$ or $\inf_S \check{f}^k$). Set $\Delta^k = f(x^k) - f^k_{\text{low}}$, $\delta^k = \kappa_l \Delta^k$ and go to Step 1.

Step 3 (*Projection*). Find the solution y^{k+1} of (2) and its multipliers λ^k_j such that the set $\hat{J}^k = \{j \in J^k : \lambda^k_j > 0\}$ satisfies $|\hat{J}^k| \leq N$. Set $t^k = \sum_{j \in J^k} \lambda^k_j$, $d^k = y^{k+1} - x^k$, $p^k = -d^k/t^k$, $\tilde{f}^k_S = \check{f}^k(y^{k+1}) + \langle p^k, \cdot - y^{k+1} \rangle$ and $\tilde{\alpha}^k_p = f(x^k) - \tilde{f}^k_S(x^k)$.

Step 4 (*Stopping criterion*). If $\sigma^k := \min(\Delta^k, \max\{|p^k|, \tilde{\alpha}^k_p\}) \leq \epsilon_{\text{opt}}$, terminate.

Step 5 (*Multiplier check*). If $t^k > t_{\max}$, replace δ^k by $\kappa_\delta \delta^k$ and go to Step 1.

Step 6 (*Descent test*). If $f(y^{k+1}) \leq f(x^k) - \kappa_d \delta^k$, set $t^k_L = 1$, $k(l+1) = k+1$ and increase the counter of descent steps l by 1; otherwise, set $t^k_L = 0$ (null step). Set $x^{k+1} = x^k + t^k_L d^k$.

Step 7 (*Selection*). Select $J_s^k \subset J^k$ such that $\hat{J}^k \subset J_s^k$. Set $J^{k+1} = J_s^k \cup \{k+1\}$.

Step 8 (*Gap update*). Set $f_{\text{low}}^{k+1} = f_{\text{low}}^k$ and $\Delta^{k+1} = f(x^{k+1}) - f_{\text{low}}^k$. If $t_L^k = 0$, set $\delta^{k+1} = \delta^k$; otherwise, choose $\delta^{k+1} \in [\min\{\delta^k, \kappa_l \Delta^{k+1}\}, \Delta^{k+1}]$. Increase k by 1 and go to Step 1.

3 MODIFIED LEVEL CONTROLS

The level control of Algorithm 1 can be modified in order to increase its efficiency without destroying convergence. We list a few (implemented) possibilities below.

Suppose Step 1 finds another lower bound $\hat{f}_{\text{low}}^k$ of f^* by computing $\inf_S \check{f}^k$, or from a feasible point to the primal problem in Lagrangian relaxation. If $\hat{f}_{\text{low}}^k \geq f(x^k) - \hat{\kappa}_l \Delta^k$ for some fixed $\hat{\kappa}_l \in [\kappa_l, 1)$ (i.e., $\hat{f}_{\text{low}}^k$ is significantly better than f_{low}^k), then Step 2 may be entered to set $f_{\text{low}}^k = \hat{f}_{\text{low}}^k$ and reduce Δ^k by $\hat{\kappa}_l$.

If $\inf_S \check{f}^k$ is not computed, then $f_{\text{lev}}^k \approx \inf_S \check{f}^k$ may be detected by the test $|p^k| \leq m_\alpha \tilde{\alpha}_p^k$ with a small $m_\alpha > 0$. Hence Step 5 may use this additional test for decreasing δ^k.

We may use the optimality measure $\tilde{\sigma}^k = \min_{j=1}^k \{|p^k| + \tilde{\alpha}_p^k\}$ for decreasing δ^k to $\min\{\kappa_l \Delta^k, \tilde{\sigma}^k\}$ and $\min\{\kappa_\delta \delta^k, \tilde{\sigma}^k\}$ at Steps 2 and 5 respectively, and for letting Step 8 choose $\delta^{k+1} \in [\min\{\delta^k, \kappa_l \Delta^{k+1}, \tilde{\sigma}^k\}, \Delta^{k+1}]$. This allows δ^k to decrease when $f(x^k)$ approaches f^*, as indicated by small $|p^k|$ and $\tilde{\alpha}_p^k$.

If the stepsize bound $t_{\max}$ is too small, the algorithm may crawl towards the solution. Hence our implementation sets $t_{\max} = 100t^1$, possibly increasing it to $\min\{10t_{\max}, 10^{10}\}$ at Step 5 if $t^k > t_{\max}$ and more than two consecutive descent steps occured.

4 SUBGRADIENT AGGREGATION

To trade off storage and work per iteration for speed of convergence, one may replace subgradient selection with aggregation as in Kiwiel (1995b). Alternatively, one may employ *selective aggregation* Kiwiel (1995c) as follows. If we pick $\hat{\imath}, \hat{\jmath} \in J^k$ with $\lambda_{\hat{\imath}}^k, \lambda_{\hat{\jmath}}^k > 0$, replace $f^{\hat{\jmath}}$ by $(\lambda_{\hat{\imath}}^k f^{\hat{\imath}} + \lambda_{\hat{\jmath}}^k f^{\hat{\jmath}})/(\lambda_{\hat{\imath}}^k + \lambda_{\hat{\jmath}}^k)$ and drop $\hat{\imath}$ from J^k, then the solution of (2) does not change.

5 SOLVING QP AND LP SUBPROBLEMS

Instead of using a separate LP solver for checking feasibility of (2) and possibly finding $\min_S \check{f}^k$, we employ the QP solver of Kiwiel (1989) within the exact penalty approach of Kiwiel (1995a). In this approach, given a penalty parameter $t > 0$, one solves

$$\text{minimize} \quad |x - x^k|^2/2 + t\max\{\check{f}^k(x), f_{\text{lev}}^k\} \quad \text{over all } x \in S. \tag{3}$$

Kiwiel (1995a) shows how to choose a sequence of penalty parameters so as to solve (2) or determine that (2) is infeasible and then deliver $\min_S \check{f}^k$. For handling large-scale problems, we intend to replace the QP solver of Kiwiel (1989) by that of Kiwiel (1994).

6 NUMERICAL EXPERIENCE

We present comparisons of Algorithm 1 with our implementation of the simplest level method of Lemaréchal *et al.* (1995). In our notation it can be stated as follows:

Algorithm 2
Step 0 (*Initiation*). Select an initial point $x^1 \in S$ and a final optimality tolerance $\epsilon_{\rm opt} \geq 0$. If S is unbounded, choose $f_{\rm low}^1 \in (-\infty, f^*]$. Set $J^1 = \{1\}$.

Step 1 (*Call oracle*). Compute $f(x^k)$ and $g(x^k)$.

Step 2 (*Compute lower bound*). Compute $f_{\rm low}^k = \inf_S \check{f}^k$.

Step 3 (*Optimality test*). Set $\Delta^k = f(x^k) - f_{\rm low}^k$. If $\Delta^k < \epsilon_{\rm opt}$ then stop.

Step 4 (*Projection*). Set $f_{\rm lev}^k = f_{\rm low}^k + \Delta^k/2$. Solve (2) to obtain x^{k+1}. Set $J^{k+1} = J^k \cup \{k+1\}$. Increase k by one and return to Step 1.

Our implementations of Algorithms 1 and 2, called DPLM and LNN respectively, were programmed in MATLAB, using mex-interfaces to Fortran QP routines. LNN calls CPLEX for its LP subproblems. Our results for LNN deviate slightly from those of Lemaréchal *et al.* (1995), perhaps because a different LP solver was employed.

DPLM stores at most $N+10$ subgradients, whereas LNN stores all of them. For problems where S is not compact we used a known lower bound on f^* for both methods. We tested two versions of DPLM: one with $f_{\rm low}^k = \max\{\inf_S \check{f}^k, f_{\rm low}^{k-1}\}$ calculated by the QP solver only when (1) is infeasible, and one in which $f_{\rm low}^k$ is calculated at every iteration using a separate LP solver. The latter version, which in the modified Step 1 used $\hat{\kappa}_l = 0.9999$, is denoted by DPLM(LP). We used the parameters $\kappa_d = 0.05$, $\kappa_l = 0.8$, $\kappa_\delta = 0.1$. The values of $f_{\rm low}^1$ are specified for each example. Step 4 of DPLM used the stopping criterion $\Delta^k \leq \epsilon_{\rm opt}$, unless stated otherwise.

6.1 Standard test problems

Table 1 gives results for standard test problems Kiwiel (1990), reporting iteration numbers at which the methods find solutions optimal to about six digits.

6.2 Traveling salesman problems

Table 2 gives results for Lagrangian minimum spanning tree relaxations of traveling salesman problems with N cities, starting from the origin. Neither method solved the 442 node problem to the required accuracy; LNN had reached a value of -50434 after 600 iterations and DPLM had reached a value of -50499 after the same number of iterations.

6.3 Eigenvalue problems

Table 3 gives results for eigenvalue problems stated as $\min \lambda_1(M + P^TXQ + Q^TX^TP)$, where M, P and Q are constant matrices, X is a matrix variable, and $\lambda_1(A)$ is the largest eigenvalue of a symmetric matrix A. In these five examples X is 4×5 and M is 7×7, and we used $f_{\rm low}^1 = -0.1$, $S = [-1, 1]^N$ and $\epsilon_{\rm opt} = 10^{-5}$. These problems are extremely

Table 1 Standard test problems

Problem	N	ϵ_{opt}	f^1_{low}	LNN	DPLM	DPLM(LP)
MXQUAD	10	10^{-5}	-10	74	60	60
GOFFIN	50	10^{-6}	-10	73	53	53
HILB	50	10^{-4}	-100	437	48	48
TR48	48	10^{-0}	-700000	209	183	183
SHUR	5	10^{-4}	0	40	33	36
L1HILN	10	10^{-6}	-10	32	22	22
MINSUM	6	10^{-4}	0	54	50	44
LOCATN	4	10^{-5}	0	24	31	32
POLAK2	8	10^{-5}	-10	110	86	86

Table 2 Travelling salesman problems

N	f^*	ϵ_{opt}	f^1_{low}	LNN	DPLM	DPLM(LP)
6	-617.000	10^{-3}	-1000	15	32	33
14	-3322.000	10^{-2}	-4000	38	42	42
29	-2013.500	10^{-2}	-3000	90	78	77
100	-20937.950	10^{-1}	-30000	312	125	130
120	-6911.250	10^{-2}	-8000	434	211	210
442	-50505.675	10^{-1}	-51000			

ill-conditionend. For termination we had to use the weaker criterion of Step 4 in DPLM and the corresponding one for LNN. DPLM(LP) performed exactly as DPLM.

Table 3 Eigenvalue problems

	LNN		DPLM	
Problem	$f(x^k)$	k	$f(x^k)$	k
EIG1	-2.03590866e-04	234	-2.035181426e-04	445
EIG2	-1.14572767e-05	204	-1.015554397e-05	366
EIG3	-3.89598493e-03	296	-3.898228667e-03	457
EIG4	-2.80409331e-03	326	-2.846003463e-03	520
EIG5	-1.51148612e-04	223	-1.806084067e-04	487
Mean		257		455

Table 4 Lotsizing problems

Test problem			LNN	DPLM	DPLM(LP)
Cap.	Cost	f^*			
Tight	High	27906	60	54	54
Tight	Med.	7997	52	49	58
Tight	Low	2893.3	20	15	15
Med. T	High	24363	54	36	39
Med. T	Med.	7722	27	28	35
Med. T	Low	2893.3	13	15	19
Med. L	High	20293	36	32	29
Med. L	Med.	7534	21	15	19
Med. L	Low	2865	1	1	1
Loose	High	18872	20	14	21
Loose	Med.	7464	13	13	12
Loose	Low	2865	1	1	1
		Mean	26.5	22.75	25.25

6.4 Lotsizing problems

The capacitated multi-item lotsizing problem is a scheduling model, which aims at scheduling production of several products over a planning horizon, while minimizing production costs, inventory holding costs and setup costs subject to demand and capacity constraints. The dual problem which is considered here arises from relaxing the capacity constraints using Lagragian multipliers. In each iteration we attempt to find a primal feasible solution in order to get a lower bound on the objective. The test problems have 8 variables which are constrained to the nonnegative orthant. For details, see Thizy and van Wassenhove (1985) and Brännlund (1993). We used $f^1_{\text{low}} = f^*$ and $\epsilon_{\text{opt}} = 10^{-6+\lceil \log_{10}(|f^*|+1) \rceil}$, i.e. 6 digits of accuracy.

6.5 Hierarchical Production Planning Problems

A variation of the multi-item lotsizing problem was introduced by Graves (1982). The test problems, which are specified in the appendix II of Graves (1982), are randomly generated. They fall into 3 different test sets. The test problems have different level of capacity limits (Tight, Medium, and Loose) and different levels of setup costs (High, Medium, Low). Each test problem has 36 dual variables. We generate one random problem for each test set, each capacity limit and each setup cost. In each iteration we attempt to find a primal feasible solution in order to get a lower bound on the objective. For details, see Graves (1982) and Brännlund (1993). We used $f^1_{\text{low}} = f^*$ and $\epsilon_{\text{opt}} = 10^{-6+\lceil \log_{10}(|f^*|+1) \rceil}$, i.e. 6 digits of accuracy.

Table 5 Hierarchical production planning problems

Test problem				LNN	DPLM	DPLM(LP)
Set	Cap.	Cost	f^*			
1	Tight	High	67435.0	143	130	143
1	Tight	Med.	27560.3	129	184	152
1	Tight	Low	13769.7	104	93	84
1	Med.	High	60163.2	133	130	100
1	Med.	Med.	21803.8	167	133	131
1	Med.	Low	5666.1	101	68	68
1	Loose	High	59118.5	121	112	124
1	Loose	Med.	21792.0	99	64	63
1	Loose	Low	5654.0	61	27	44
2	Tight	High	67334.0	128	146	178
2	Tight	Med.	28336.4	148	190	176
2	Tight	Low	14541.6	94	77	79
2	Med.	High	59698.9	138	116	150
2	Med.	Med.	22015.2	149	180	132
2	Med.	Low	7148.4	138	128	152
2	Loose	High	58116.9	126	129	169
2	Loose	Med.	21479.1	144	153	136
2	Loose	Low	5597.8	69	56	55
3	Tight	High	62346.1	88	110	129
3	Tight	Med.	32719.1	133	143	154
3	Tight	Low	22114.0	61	57	59
3	Med.	High	55385.9	122	109	123
3	Med.	Med.	26112.1	147	115	105
3	Med.	Low	15911.3	79	66	83
3	Loose	High	50000.5	102	99	98
3	Loose	Med.	20712.6	142	125	124
3	Loose	Low	10408.3	77	103	70
			Mean	116.4	112.7	114.1

6.6 Unit Commitment Problems

The thermal unit commitment problem is a large mixed-integer non-linear mathematical programming problem which arises in short-term power production planning. The problem consists of minimizing production costs over a planning period, satisfying system load and reserve constraints as well as technical constraints on each production unit.

We refer to Wood and Wollenberg (1984) for a description of the unit commitment problem. Thorough descriptions of the BARD test problem can be found in Bard (1988), and for the EPRI50 in Zeminger *et al.* (1977). BARD2 is a relaxed modification of BARD. ABB and ABB2 can be obtained from the authors. We used $f^1_{\text{low}} = f^*$.

Table 6 Unit commitment problems

Test problem	N	f^*	ϵ_{opt}	LNN	DPLM	DPLM(LP)
BARD	20	540952	1	399	709	386
BARD2	20	539923	1	334	751	260
ABB	40	105973	1	97	77	84
ABB2	40	143696	1	138	72	93
EPRI50	96	2843720	10	309	156	330
			Mean	255	353	231

REFERENCES

Bard, J.F. (1988) Short-term scheduling of thermal-electric generators using Lagrangian relaxation. *Operations Research*, **36**, 756–66.

Brännlund, U. (1993) *On relaxation methods for nonsmooth convex optimization.* Ph.D. thesis, Department of Mathematics, Royal Institute of Technology, Stockholm.

Brännlund, U., Kiwiel, K.C. and Lindberg, P.O. (1995) A descent proximal level bundle method for convex nondifferentiable optimization. *Operations Research Letters* (to appear).

Graves, S.C. (1982) Using Lagrangian techniques to solve hierarchical production planning problems. *Management Science*, **28**, 260–75.

Kiwiel, K.C. (1989) A dual method for certain positive semidefinite quadratic programming problems. *SIAM Journal on Scientific and Statistical Computing*, **10**, 175–86.

Kiwiel, K.C. (1990) Proximity control in bundle methods for convex nondifferentiable minimization. *Mathematical Programming*, **46**, 105–22.

Kiwiel, K.C. (1994) A Cholesky dual method for proximal piecewise linear programming. *Numerische Mathematik*, **68**, 325–40.

Kiwiel, K.C. (1995a) Finding normal solutions in piecewise linear programming. *Applied Mathematics and Optimization*, **32**, (to appear).

Kiwiel, K.C. (1995b) Proximal level bundle methods for convex nondifferentiable optimization, saddle-point problems and variational inequalities. *Mathematical Programming*, (to appear).

Kiwiel, K.C. (1995c) The efficiency of subgradient projection methods for convex optimization, part II: Implementations and extensions. *SIAM Journal on Control and Optimization*, (to appear).

Lemaréchal, C., Nemirovskii, A.S. and Nesterov, Yu.E. (1995) New variants of bundle methods. *Mathematical Programming*, (to appear).

Thizy, J.M. and van Wassenhove, L.N. (1985) Lagrangean relaxation for the multi-item capacitated lot-sizing problem: A heuristic implementation. *IIE Transactions*, **17**, 308–13.

Wood, A.J. and Wollenberg, B.F. (1984) *Power Generation Operation and Control.* John Wiley and Sons, New York.

Zeminger, H.W., Wood, A.J., Clark, H.K., Laskowski, T.F. and Burns, J.D. (1977) Synthetic electric utility systems for evaluating advanced technologies. **EM-285.** Electrical Power Research Institute (EPRI).