SIAG/OPT Views-and-News

A Forum for the SIAM Activity Group on Optimization

Volume 22 Number 2 May 2011

Contents

SIAM OP11

This issue of SIAG/OPT Views-and-News is dedicated to the 2011 SIAM Conference on Optimization. Claudia Sagastizábal has written a beautiful and elegant survey of nonsmooth optimization showing how traditional methods can be cleverly improved by exploiting structure beyond the black box. Don't miss her plenary presentation on Wednesday at 1pm!

Other highlights of OP11 include the SIAG/OPT business meeting on Wednesday (6:45-7:15pm) and a mini-tutorial on *Convex Relaxation and Applications* (Thu. 9:30–11:30am) presented by Michael Friedlander, Maryam Fazel, and Etienne de Klerk.

I am grateful to two "Heiners" (or local heroes), Stefan Ulbrich and Kai Habermehl, for providing at very short notice a great guide of things-to-do in Darmstadt. I am sure their ideas will make the already excellent OP11 meeting even more enjoyable!

Sven Leyffer, Editor

leyffer@mcs.anl.gov

http://www.mcs.anl.gov/~leyffer/

Nonsmooth Optimization: Thinking Outside of the Black Box

Claudia Sagastizábal

CEPEL, Electric Energy Research Center, Eletrobrás Group.
On leave from INRIA, France. sagastiz@impa.br

1. Introduction

In many optimization problems nonsmoothness appears in a structured manner, because the objective function has some special form. In compressed sensing, for example, regularized least squares problems have composite objective functions. Likewise, separable functions arise in large-scale stochastic or mixed-integer programming problems solved by certain decomposition technique.

The last decade has seen the advent of a new generation of bundle methods, capable of fully exploiting structured objective functions. Such information, transmitted via an oracle or black box, can be handled in various ways, depending on how much data is given by the black box. If certain first-order information is missing, it is possible to deal with inexactness very efficiently. But if some second-order information is available, it is possible to mimic a Newton algorithm and converge rapidly.

We outline basic ideas and computational questions, highlighting the main features and challenges in the area on simple examples.

2. A Structured Function

For convex optimization, certain complexity results establish that oracle based methods have at best a linear rate of convergence. This is why recent nonsmooth optimization research has focused on exploiting structure in the objective function, as a way to speed up numerical methods.

Bundle methods, [HULL93, vol. II], in particular, have evolved from the early "conjugate subgradient" methods in [Lem75], [Wol75] to a class of specialized structured methods with improved performance.

In this new generation of bundle methods, the word "structure" can mean many different things. For example, consider the problem of minimizing

$$f(x) = \sqrt{x^\top Ax} + x^\top Bx \qquad (1)$$

over $\Re^n$, where A, B are two symmetric positive semidefinite matrices.Suppose for convenience that B is positive definite, so that the origin is the unique minimizer.

We now explore different structures for this function from [LO08], that can be exploited in an algorithmic setting.

2.1 Sum Structure

The most obvious structure in (1) is that the function is defined by two terms:

$$f(x) = f_1(x) + f_2(x) \text{ with } \begin{cases} f_1(x) = \sqrt{x^\top Ax} \\ f_2(x) = x^\top Bx\,. \end{cases}$$

This type of basic structure appears often when decomposing a hard-to-deal-with problem, by Lagrangian relaxation or in a Benders decomposition scheme, see Sec. 5. below.

2.2 $\mathcal{VU}$-Structure

Theoretical tools such as the $\mathcal{U}$-Lagrangian [LS97a], [LOS00], $\mathcal{VU}$-space decomposition [MS00], and partly smooth functions [Lew02], can be seen as tools to "extract" smooth structure from nondifferentiable functions. For our example, the function is not differentiable on $\mathcal{N}(A)$, the null space of the matrix A. In this region, the first term in (1) vanishes and f appears as if smooth. This is the $\mathcal{U}$-space; all nondifferentiability of f is concentrated on the orthogonal complement: for any $x \in \mathcal{N}(A)$,

$$\mathcal{V}(x) = \mathcal{U}(x)^\perp = \mathcal{R}(A), \text{ the range of } A.$$

The convex partly smooth function (1) is an n-dimensional extension of the bivariate function $f(x_1, x_2) = |x_1| + x_2^2$. Figure 1 represents the corresponding two views, from the $\mathcal{V}$ and $\mathcal{U}$ subspaces at $x = (0, x_2)$.

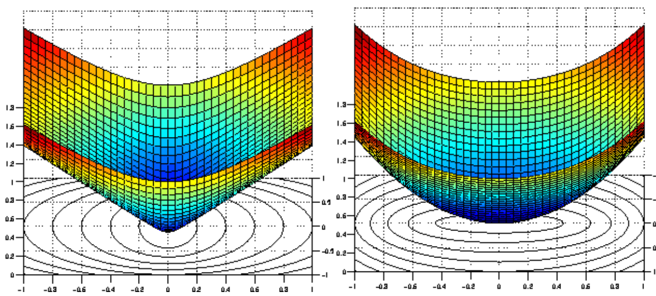


Figure 1: $\mathcal{V}$ and $\mathcal{U}$ views (left and right).

2.3 Composite Structure

Sometimes it is also possible to "split" smoothness and nonsmoothness by writing the function as the composition of a smooth mapping $c : \Re^n \to \Re^m$ with a positively homogeneous (of degree 1) convex function $h : \Re^m \to \Re$, so that $f = h \circ c$. Note in particular that the outer function is real-valued on $\Re^m$, an assumption that excludes indicator functions, but still covers a rich enough family of functions for interesting problems (max-functions, sum of Euclidean norms, eigenvalue optimization, regularized minimization maps, ℓ_1-penalization of nonlinear programming problems). The function f may be nonconvex, but with our assumptions the following chain-rule holds: $\partial f(x) = Dc(x)^\top \partial h(C)$ where $Dc(x)$ is the mapping Jacobian and $C = c(x)$.

For the function in (1), one can let $m = n+1$ and define the inner mapping as

$$c_j(x) = x_j, \text{ for } j = 1, \ldots, n \text{ and } c_{n+1}(x) = x^\top Bx\,.$$

The outer positively homogeneous function is

$$h(C) = \sqrt{C_{1:n}^\top AC_{1:n}} + C_{n+1}\,,$$

where the notation $C_{i:j}$ corresponds to the subvector formed by the i^{th} to j^{th} components of the vector $C \in \Re^{n+1} = \Re^m$. Composite functions were considered early on in [Fle87, Ch. 14], then in [BF95] and [Sha03], and have been more recently re-examined in [Nes07], [LW08], and [Sag10].

3. Knowing More or Knowing Less: The Oracle Information

Typically, a bundle method defines iterates by minimizing a *model* of the black box function f, denoted by φ, which is made up of pieces that together approximate f in some manner. The set of information needed to define the model is called the *bundle* $\mathcal{B}$. The prototypical example is the *classical cutting-plane model*

$$\varphi(x) = \max_{i\in\mathcal{B}}\{f^i + {g^i}^\top(x - x^i)\}\,,$$

where the x^i's are past iterates, and

$$\mathcal{B} = \bigcup_i \left\{ \left(x^i, f^i = f(x^i)\,, g^i \in \partial f(x^i) \right) \right\}\,.$$

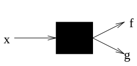
In the model definition, we write $i \in \mathcal{B}$ to mean that there exists an element in the set $\mathcal{B}$ indexed by i.

In order to prevent oscillations, a stabilization term is added to the model function to cause the next iterate to be near a "good" point, the last *serious* iterate in the bundle terminology, denoted by $\hat{x}$. A possible stabilization device is to define iterates by solving the proximal quadratic programming problem (QP)

$$\min_x\{\varphi(x) + \frac{1}{2}\mu|x - \hat{x}|^2\}\,,$$

where the prox-parameter μ is positive. When the new iterate gives sufficient decrease in f, the step is declared serious and replaces $\hat{x}$. Otherwise, the step is declared *null*. In both cases, the bundle $\mathcal{B}$ is enriched with the function and subgradient information from the last generated point.

Each linearization in the cutting-plane model is defined with information produced by the *oracle* or *black box*. In a real-life application, this is a piece of code written by the user to compute function values and one subgradient for any given vector x. For the example in (1), the black box provides



$$f(x) = \sqrt{x^\top Ax} + x^\top Bx$$
$$\nabla f(x) = \frac{Ax}{\sqrt{x^\top Ax}} + 2Bx \qquad \text{if } 0 \neq x \in \mathcal{R}(A)$$
$$g(x) \in A^{\frac{1}{2}}(B(0;1)) + 2Bx \quad \text{if } x \in \mathcal{N}(A)$$

(here $B(0;1)$ denotes the n-dimensional unit ball.)

The classical cutting-plane model, based on the bundle $\mathcal{B}$ (defined in turn via the black box), is not the only possibility. For some problems, the user may be able to provide additional information, while for other ones, it may be impossible to make exact black box calculations. Bundle methods can handle well the availability or unavailability of such information. We now review some alternative models that can be built in such situations.

3.1 Sum Black Box

Consider the sum-structure, $f(x) = f_1(x) + f_2(x)$. If for each $j = 1, 2$ individual subgradients $g_j(x) \in \partial f_j(x)$ are known, the *disaggregate* cutting-plane model has the form

$$\varphi(x) = \max_{i\in\mathcal{B}_1}\{f_1^i + {g_1^i}^\top(x - x^i)\} + \max_{i\in\mathcal{B}_2}\{f_2^i + {g_2^i}^\top(x - x^i)\}\,,$$

where we defined the individual bundles

$$\mathcal{B}_j = \bigcup_i \left\{ \left(x^i, f_j^i = f_j(x^i)\,, g_j^i \in \partial f_j(x^i) \right) \right\}$$

for $j = 1, 2$. Being the sum of maxima, this model is in principle better than the classical one, given by the maximum of sums. Note, however, that the bundle of information is also disaggregate and, hence, larger. This makes each QP larger, and can substantially increase the CPU time spent in each iteration. For a mid-term optimal generation management problem briefly described in Sec. 5., the computational study in [BLRS01] shows that it is advisable to keep the number of terms in the sum low, by regrouping terms, if necessary.

Incidentally, note that for (1), the second term $f_2(x) = x^\top Bx$ is smooth. It is then possible to build the hybrid model

$$\varphi(x) = \max_{i\in\mathcal{B}_1}\{f_1^i + {g_1^i}^\top(x - x^i)\} + x^\top Bx\,,$$

which takes advantage of the additional smoothness in one of the terms (and uses a smaller bundle). We refer to [LOP11] for an example of the hybrid model in the area of telecommunications.

3.2 Clear Black Box

For the composite structure, the function and subgradient information can be given separately for the inner smooth mapping and the outer function. More precisely, instead of simply having $f(x) = (h \circ c)(x)$ and $g(x) \in \partial(h \circ c)(x)$, we suppose the oracle has the ability to make separate computations:

$$\begin{cases} \forall x \in \Re^n & \text{an inner black box computes} \\ & c(x) \text{ and its Jacobian } Dc(x) \\ \forall C \in \Re^m & \text{an outer black box computes} \\ & h(C) \text{ and a subgradient } G \in \partial h(C). \end{cases}$$

Having this separate information makes it possible to define the *composite* model

$$\varphi(x) = \max_{i \in \mathcal{B}_c} \left\{ G^{i\top}\Big(c(\hat{x}) + Dc(\hat{x})^\top (x - \hat{x})\Big)\right\}$$

where

$$\mathcal{B}_c = \bigcup_i \{G^i \in \partial h(C^i) \text{ for } C^i = c(x^i)\}$$

and where each x^i was generated at some past iteration, as before. To understand how this model is built, replace the smooth mapping by its Taylor linearization around the serious point: $c(\hat{x}) + Dc(\hat{x})^\top(x - \hat{x})$ and consider the approximation

$$h(c(\hat{x}) + Dc(\hat{x})^\top(x - \hat{x})) \approx (h \circ c)(x)\,, \qquad (2)$$

introduced in [LW08]. The composite model is nothing but the classical cutting-plane model for the outer function in this approximation, exploiting the fact that h is positively homogeneous. We refer to [Sag10] for details, including intensive numerical results showing the effectiveness of the approach.

Since for the function (1) the approximation in (2) is

$$\sqrt{x^\top A x} + \hat{x}^\top B \hat{x} + 2(B\hat{x})^\top (x - \hat{x})\,,$$

with smooth rightmost terms, the composite model could approximate only the first term by cutting planes. Moreover, if the matrix B is not too large or too dense, it can be used in the QP for defining the metric in the quadratic term, instead of using the prox-parameter (this is comparable to the clear black box providing also second-order objects.)

3.3 Noisy Black Box

The previous oracles take advantage of *additional knowledge* of the function structure. In some situations, however, the black box has *less knowledge*, because computing exact values of the function and one subgradient is too costly, or just impossible. Bundle methods in their *inexact* variants can be used to deal with inaccurate linearizations.

For the function (1) one could think of a situation in which the matrix B is too large and dense, and only some random diagonal elements are known at any given call.

The corresponding noisy black box will only output *approximate* values:

$$\left[\begin{array}{l}\text{a function estimate } f_x \in [f(x) - \epsilon_f, f(x) + \epsilon_g] \\ \text{a subgradient estimate } g_x \in \partial_{\epsilon_f + \epsilon_g} f(x)\,,\end{array}\right.$$

where the unknown errors $\epsilon_f\,, \epsilon_g \geq 0$ are bounded, and where $\partial_\epsilon f$ denotes the ϵ-subdifferential in convex analysis. As illustrated in Figures 2 and 3, inexact linearizations no longer define cutting planes.

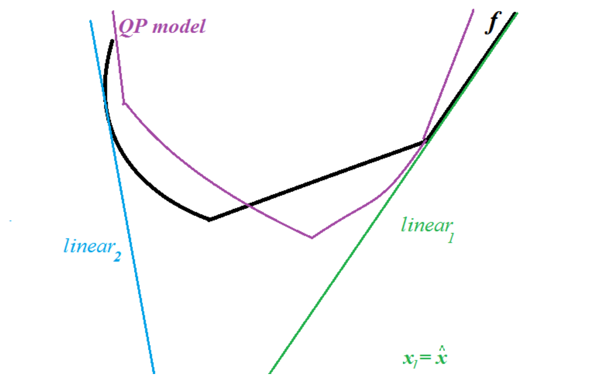


Figure 2: Exact linearizations.

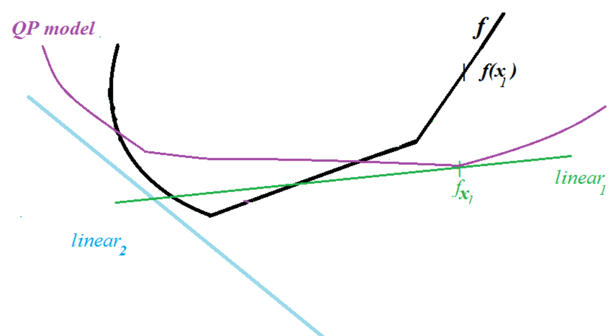


Figure 3: Inexact linearizations.

The oracle noise needs to be handled carefully, to detect when inaccuracy becomes too cumbersome. The detection mechanism is based on whether or not the optimal QP value is below the best value function, $f_{\hat{x}}$. Since the function is convex, a QP value greater than $f_{\hat{x}}$ can only be due to linearization inaccuracy. In such a case, the prox-parameter μ is decreased, to decrease the next optimal QP value and bring it closer to $f_{\hat{x}}$.

This simple noise attenuation step, introduced in [Kiw06], ensures convergence of the serious inexact functional iterates to a value that differs from the exact minimum value by less than $2(\epsilon_f + \epsilon_g)$.

For the classical cutting-plane model, the noisy bundle of information is

$$\mathcal{B} = \left\{ i : \left(x^i, \begin{array}{l} f^i = f_{x^i} \in [f(x^i) - \epsilon_f, f(x^i) + \epsilon_g], \\ g^i = g_{x^i} \in \partial_{\epsilon_f + \epsilon_g} f(x^i) \end{array} \right) \right\}.$$

Any of the previous models (disaggregate, composite, hybrid) can be considered in the inexact setting, by introducing the noisy information into the corresponding bundle.

4. $\mathcal{VU}$ Primal-Dual Models

The bundle methodology we just presented can be considered of *primal* form, in the sense that a model for the objective function, φ, is being built along iterations. Early bundle methods were created using a *dual* view, trying to iteratively build an approximate subdifferential of the function at an optimum, using $\partial\varphi$.

The $\mathcal{VU}$ algorithm [MS05] is an innovation that builds a *primal-dual model*, trying to identify the $\mathcal{V}$-subspace in order to make a $\mathcal{U}$-Newton move on its orthogonal complement. The initial step comes from noticing that bundle QP solution gives the proximal point of the model φ at the serious iterate, $\hat{x}$. Near an optimal solution, [MS02] shows that the proximal point of the function f at $\hat{x}$ is on a "ridge" of nondifferentiability, the *activity manifold* in [Lew02]. So a typical bundle iteration amounts to making an approximate $\mathcal{V}$-projection. But if we knew the $\mathcal{V}$-subspace, we would also know its orthogonal complement, the $\mathcal{U}$-subspace, on which the function looks smooth. Then, moving in a Newton-like direction tangent to the manifold would give fast convergence.

After solving a proximal QP as a classical bundle method, the $\mathcal{VU}$-bundle algorithm solves a *second* QP per iteration, to identify the $\mathcal{V}$-subspace and define a Newton-like direction. Along this direction, the function behaves nicely, because it coincides with certain smooth $\mathcal{U}$-Lagrangian. The second QP uses the *dual model* built along iterations, which approximates the subdifferential of the $\mathcal{U}$-Lagrangian. This extra step makes the method provably superlinearly convergent for the subsequence of serious iterates, under reasonable assumptions.

5. Some Applications

Our survey so far is by no means exhaustive. At OP11, there is a range of sessions related to new variants or applications of bundle methods such as MS20, CP21, MS24, CP27, and MS64.

We finish with some examples demonstrating the versatility and power of the bundle methodology.

5.1 Price Decomposition on a Scenario Tree

We consider a mid-term generation planning problem over a power mix with 85 classic thermal power plants, 58 nuclear reservoirs, and one reservoir modeling the spot market, posed by Electricité de France (EDF) and studied in [BLRS01], [ES10]. The two year optimization horizon has a daily time discretization. Each day is further divided into three periods, representing peak hours of high demand, base demand, and offpeak hours, yielding more than 1,000 time steps. In this mid-term horizon, the level of demand is the main source of uncertainty, particularly during winter periods. Uncertainty is modeled with a scenario tree that typically has more than 50,000 nodes. On each node, an optimization problem with more than 100 variables and 100 constraints has to be solved. So, even with a rather simplistic description of the power mix, the resulting stochastic linear program is large-scale and needs to be solved by some decomposition technique. Currently, EDF applies Lagrangian relaxation of the demand constraints and solves the dual problem by the variable metric proximal bundle method [LS97b].

For a given multiplier x, the relaxation uncouples the generation of different power plants, yielding separate subproblems shown in Figure 4. As a result, the dual function has the sum structure in Sec. 2.1, with a number of terms equal to the number of "units" (a term used to refer to both plants and reservoirs).

Computing the pairs $(f_j(x), g_j(x) \in \partial f_j(x))$ is very costly for some units j. In particular, for a scenario tree with 50,000 nodes, evaluating the term for one nuclear plant amounts to solving a linear program with 100,000 variables (generation and reservoir levels) and about 300,000 constraints. By contrast, a term for any classical thermal plant involves solving a linear program with 50,000 variables and

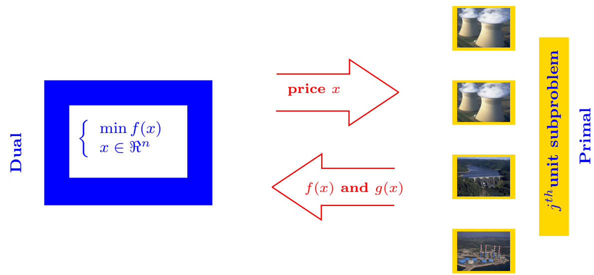


Figure 4: Price decomposition.

constraints. Since, in addition, there is no temporal coupling between nodes, such linear programs are solved in milliseconds. There is a huge time difference between solving all the non-nuclear subproblems and solving only one nuclear subproblem.

For convenience, we regroup terms and write $f(x) = f_{N_1}(x) + f_{N_2}(x) + f_{N_3}(x) + f_{\tilde{N}}(x)$, corresponding, respectively, to nuclear units gathered into three subsets (with 20, 20, and 18 nuclear units each one), and to all the non-nuclear units. The employed cutting-plane model disaggregates the bundle into the same terms defining f, so that each term in the sum has an individual cutting-plane model.

The *incremental* bundle method in [ES10] is an inexact bundle method in which the subproblem solution corresponding to one of the nuclear terms f_{N_j} is skipped, alternating between the three nuclear subsets at different iterations. The missing oracle information is replaced by estimations given by the cutting-plane model for f_{N_j}. For a tolerance corresponding to a deviation in demand satisfaction of 20MW per node of the scenario tree (a negligible amount, if compared to the average power load of about 50,000MW), the incremental bundle method reaches the same precision as the proximal approach [LS97b], but uses 25% less CPU-time.

5.2 Two-Stage Stochastic Programming

For two-stage stochastic programs with recourse, the paper [OSS11] revisits the L-shaped method from a nonsmooth optimization point of view.

For simplicity, consider the particular case of a linear program with random right hand side. Given a convex polyhedron X, the (convex nonsmooth) first-stage problem is

$$\min_{x \in X} c^\top x + \mathcal{Q}(x) \text{ for } \mathcal{Q}(x) := \mathbb{E}[Q(x;\xi)] \,. \tag{3}$$

The expectation in the objective function is taken with respect to the probability distribution of a random variable ξ, over the optimal values of the second-stage problem, given by

$$Q(x;\xi) = \left\{ \begin{array}{ll} \min_{y \geq 0} & q^\top y \\ \text{s.t.} & Tx + Wy = h(\xi) \,. \end{array} \right.$$

In the second-stage constraints, T and W are matrices and h is a function of the random variable ξ.

Suppose there are finitely many realizations ξ_i, each one with probability p_i for $i = 1, \ldots, N$ and let $h_i = h(\xi_i)$. Then the recourse function is separable along scenarios:

$$\mathcal{Q}(x) = \sum_{i=1}^{N} p_i Q_i(x) \text{ with } Q_i(x) := Q(x;\xi_i) \,.$$

In this summation, each term corresponds to solving one linear program, written in a primal or dual form:

$$\begin{aligned} Q_i(x) &= \left\{ \begin{array}{ll} \displaystyle\min_{y \geq 0} & q^\top y \\ \text{s.t.} & Tx + Wy = h_i \end{array} \right. \\ &= \left\{ \begin{array}{ll} \displaystyle\max_{u} & u^\top (h_i - Tx) \\ \text{s.t.} & W^\top u \leq q \,. \end{array} \right. \end{aligned} \tag{4}$$

For any given x^0 and ξ_i, if the primal and dual feasible sets above are nonempty, $Q_i(x^0)$ is finite with subgradients of the form $-T^\top u_i$, where $u_i = u_i(x^0) = \arg\max\{u^\top(h_i - Tx^0) : W^\top u \leq q\}$ is a solution to the dual linear program.

Once again, the objective function in (3) has a sum structure, illustrated by Figure 5.

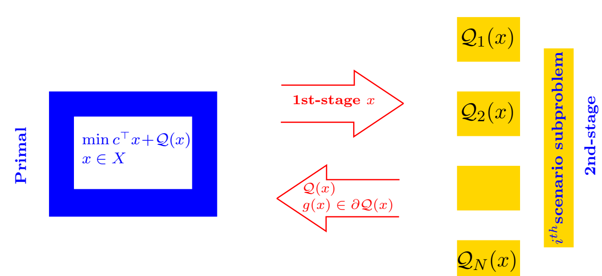


Figure 5: L-shaped decomposition.

When modeling uncertainty, it is desirable to use as many scenarios as possible, to ensure an accurate representation of the underlying stochastic process. But a large number of scenarios means that $\mathcal{Q}(x)$

involves the summation of many terms and, hence, the solution of many linear programs.

In order to save computational time, for some scenarios one can either solve (4) only approximately, or just skip the solution of (4), replacing the missing information by some reasonable value. From the 1st-stage problem perspective, this means having a noisy black box, a situation that is well handled by inexact bundle methods.

In [OSS11] different noisy black boxes are explored. The *collinearity* strategy therein groups scenarios by looking at the cosines

$$cos(h_i - Tx, h_j - Tx) \text{ for all scenarios } 1 \leq i, j \leq N.$$

Indeed, when for a given pair (i, j) such value is small, the optimal dual values for $Q_i(x)$ and $Q_j(x)$ in (4) are similar. It is then possible to solve only one linear program, say the one corresponding to scenario i, and use the dual optimal solution u_i as a proxy for u_j, thus defining a noisy black box for Q_j.

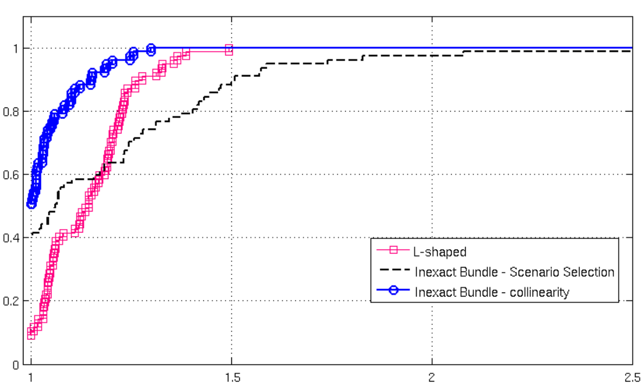


Figure 6: Accuracy and CPU time.

For seventy seven runs of various stochastic linear programs with varying values of N, Figure 6 gives a performance profile comparing the inexact bundle method using a noisy black box given by the collinearity approach (blue circles), an inexact bundle method using a noisy black box based on scenario selection [HR03] (black dashed line), and the L-shaped method (red squares).

The measure in the profile is a combination of accuracy and CPU time (each indicator with weight $\frac{1}{2}$). The figure shows that the use of the inexact approach with the collinearity-based noisy black box gives better results than the L-shaped method. The same can be said for the scenario selection black box, but to a lesser extent, at least for the considered problems, because selecting scenarios makes each iteration more costly and slows down the overall process.

The collinearity strategy is as accurate as the (exact) L-shaped method, but more than 4 times faster. The reason is that while the L-shaped method uses an exact black box (to solve (3) by a cutting-plane method), the inexact bundle method uses the noisy black box, which, in average, solves no more than 17% of the N linear programs per iteration. Table 1 reports the mean and total number of linear programs solved by each method for all the runs in Figure 6.

Method	mean	total
L-shaped	1,082	83,300
Inexact Bundle, scenario selection	158	12,157
Inexact Bundle, collinearity strategy	179	13,812

Table 1: Linear programs solved by each method.

5.3 Composite and $\mathcal{VU}$ optimization

When there is a clear black box, the composite model can be a good option for some classes of functions. Such is the case for the function in (1), as shown by Figure 7, reporting in a semilogarithmic scale the function values of iterates generated by the composite bundle method in [Sag10] (blue circles) and BFGS method, with the special line search in [LO08] (black crosses). The instance corresponds to $n = 100$ and $dim\, \mathcal{R}(A) = 50$. Since BFGS only has a heuristic stopping test, we let both algorithms run until the black box was called 1,000 times.

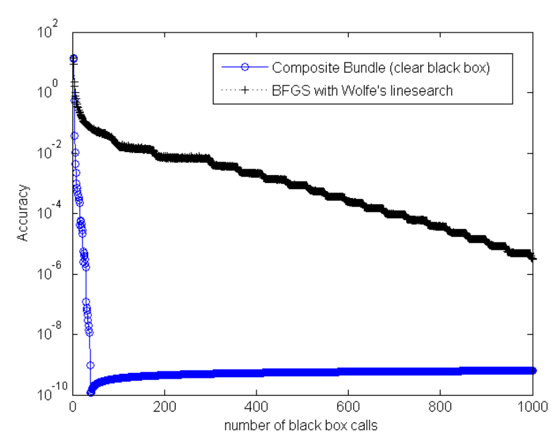


Figure 7: Function value convergence for (1).

For the example, the optimal value is zero, so Figure 7 gives an estimate of the rate of convergence of the methods. For nonsmooth optimization, BFGS cannot have superlinear convergence (BFGS is not even provably globally convergent in such a setting). We observe that, while the linesearch makes the algorithm descend, BFGS's generated points are not converging rapidly. The situation is different with the composite bundle method. In this case, since the clear black box outputs the smooth inner Hessians (all null except for $\nabla^2 c_{n+1}(x) = 2B$), the second-order information is incorporated in the metric defining the quadratic term in the bundle QP. This curvature explains the superlinear-like behavior observed in the figure for the bundle method.

We finish with a nonsmooth test-function, called `maxquad`, [BGLS06, p. 153], given by the piecewise maximum of five convex quadratic functions in $\Re^{10}$. At the unique optimal solution, four of the quadratic functions are active, but the fifth (inactive) piece is stiff, making `maxquad` an interesting academic example for testing different algorithms.

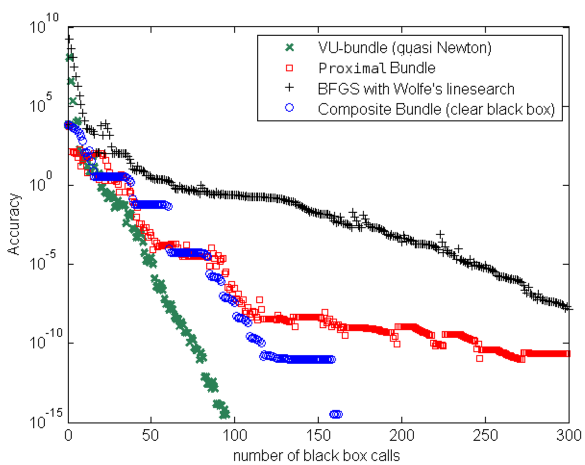


Figure 8: Function value convergence for `maxquad`.

Figure 8 shows the accuracy on function values for $\mathcal{VU}$-bundle iterates, in a quasi-Newton variant (green x), when compared to the composite bundle method (blue circles), the proximal bundle method [LS97a] (red squares), and BFGS (black crosses).

We observe the very rapid rate of convergence of the $\mathcal{VU}$-algorithm. The composite bundle method is fast, but has some plateaux, corresponding to null steps. Both the proximal bundle and BFGS methods tend to stall in speed as they try to get more digits of precision. This is a well-known phenomenon in nonsmooth optimization: usually many methods fail when tightening the stopping tolerance, because the main difficulty near a kink is in catching a solution with high accuracy.

Acknowledgements. Thanks to Bob Mifflin for many suggestions and comments. Research partially supported by AFOSR Grant FA9550-08-1-0370, NSF Grant DMS-0707205, and CNPq & Faperj from Brazil.

REFERENCES

[BF95] J. V. Burke and M. C. Ferris, *A Gauss-Newton method for convex composite optimization*, Mathematical Programming **71** (1995), 179–194.

[BGLS06] J. F. Bonnans, J.-Ch. Gilbert, C. Lemaréchal, and C. Sagastizábal, *Numerical optimization. theoretical and practical aspects*, Universitext, Springer-Verlag, Berlin, 2006, 2nd. editiion, xiv+423 pp.

[BLRS01] L. Bacaud, C. Lemaréchal, A. Renaud, and C. Sagastizábal, *Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners*, Computational Optimization and Applications **20** (2001), no. 3, 227–244.

[ES10] G. Emiel and C. Sagastizábal, *Incremental-like bundle methods with application to energy planning*, Computational Optimization and Applications **46** (2010), 305–332.

[Fle87] R. Fletcher, *Practical methods of optimization* (second edition), John Wiley & Sons, Chichester, 1987.

[HR03] H. Heitsch and W. Römisch, *Scenario reduction algorithms in stochastic programming*, Computational Optimization and Applications **24** (2003), no. 2-3, 187–206.

[HULL93] J.-B. Hiriart-Urruty and C. Lemaréchal, *Convex analysis and minimization algorithms*, Grund. der math. Wiss, no. 305-306, Springer-Verlag, 1993, (two volumes).

[Kiw06] K. C. Kiwiel, *A proximal bundle method with approximate subgradient linearizations*, SIAM J. Optim. **16** (2006), 1007–1023.

[Lem75] C. Lemaréchal, *An extension of Davidon methods to nondifferentiable problems*, Math. Programming Stud. **3** (1975), 95–109.

[Lew02] A. S. Lewis, *Active sets, nonsmoothness and sensitivity*, SIAM J. on Optimization **13** (2002), no. 3, 702–725.

[LO08] A. S. Lewis and M. L. Overton, *Nonsmooth Optimization via BFGS*, `http://www.optimization-online.org/DB_HTML/2008/12/2172.html`, 2008.

[LOP11] C. Lemaréchal, A. Ouorou, and G. Petrou, *A bundle-type algorithm for routing in telecommunication data networks*, Comput. Optim. Appl. **44** (2011), no. 3, 385–409.

[LOS00] C. Lemaréchal, F. Oustry, and C. Sagastizábal, *The $\mathcal{U}$-Lagrangian of a convex function*, Trans. Amer. Math. Soc. **352** (2000), no. 2, 711–729.

[LS97a] C. Lemaréchal and C. Sagastizábal, *Practical aspects of the Moreau-Yosida regularization: theoretical preliminaries*, SIAM J. on Optimization **7** (1997), no. 2, 367–385.

[LS97b] ———, *Variable metric bundle methods: from conceptual to implementable forms*, Math. Program., Ser. A **76** (1997), 393–410.

[LW08] A. S. Lewis and S. J. Wright, *A proximal method for composite minimization*, `http://www.optimization-online.org/DB_HTML/2008/12/2162.html`, 2008.

[MS00] R. Mifflin and C. Sagastizábal, *On $\mathcal{VU}$-theory for functions with primal-dual gradient structure*, SIAM J. on Optimization **11** (2000), no. 2, 547–571.

[MS02] ———, *Proximal points are on the fast track*, Journal of Convex Analysis **9** (2002), no. 2, 563–579.

[MS05] ———, *A $\mathcal{VU}$-algorithm for convex minimization*, Math. Program., Ser. A **104** (2005), no. 2–3, 583–608.

[Nes07] Y. Nesterov, *Gradient methods for minimizing composite objective functions*, `http://www.optimization-online.org/DB_HTML/2007/09/1784.html`.

[OSS11] W. L. Oliveira, C. Sagastizábal, and S. Scheimberg, *Inexact Bundle Methods for Two-Stage Stochastic Programming*, `http://www.optimization-online.org/DB_HTML1G/2010/09/2729.html`. Accepted for publication in SIAM J. on Optimization, 2011.

[Sag10] C. Sagastizábal, *Composite Proximal Bundle Methods*, `http://www.optimization-online.org/DB_HTML/2009/07/2356.html`, 2010.

[Sha03] A. Shapiro, *On a class of nonsmooth composite functions*, Math. Oper. Res. **28** (2003), no. 4, 677–692.

[Wol75] P. Wolfe, *A method of conjugate subgradients for minimizing nondifferentiable functions*, Math. Programming Stud. **3** (1975), 145–173.

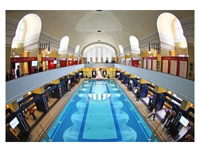
Bulletin

11 Things to do in Darmstadt

Beer-Tasting at Braustübl Darmstadt: Go to Braustübl Darmstadt (close to main train station and Hotel Maritim) and order a "Bier-Probe". Get 5 little beers and find the beer that's best for you. Enjoy the open and friendly atmosphere with the local people of Darmstadt, called "Heiner".

Mathildenhöhe. Have a walk over the Mathildenhöhe. Enjoy the Five-Finger-Tower, climb its top and have a perfect view over Darmstadt. And don't miss visiting the Russian Chapel and the art nouveau mansions. Right now, there is an exhibition on Serious Games: War — Media — Art.

Jugendstilbad (Swimming and Relaxing). Relax in the art nouveau style indoor swimming pool. Take a sauna or relax in a huge spa area in a newly refurbished oasis of well-being. Perfect after a long conference day.



Hundertwasser House. Friedensreich Hundertwasser is one of the most famous european artists and architects with an unmistakable style. The building is also called "Waldspirale" (forest spiral) because of its distinctive style. Can be perfectly combined with Point 5.

Bavarian Beergarden. Have the typical Bavarian beergarden feeling with beer, sausages and pretzels. The typical German way to finish a hard working day and meet some friends.

Castle Frankenstein. A well-preserved castle close to Darmstadt. It is rumored that the novel of monster Frankenstein by Mary Shelley was inspired by an alchemist living in Castle Frankenstein. It's located about 10km to the south of Darmstadt at the beginning of the green valleys of the Odenwald-mountains.

Oberwaldhaus. When a "Heiner" (local Darmstadtian) wants to relax, he travels to the Oberwaldhaus. Here you can make little boating trips on a small lake, play miniature golf or ride a pony.

Rosenhöhe, Oberfeld. Have a walk in the beautiful site of Rosenhöhe with more than 10.000 roses blooming especially in the famous rose-dome.

Market place. Enjoy the atmosphere at the central market place, not far from the conference location. Drink a house-brewed beer at the "Ratskeller", enjoy (probably Darmstadt's best) ice-cream at "Eis Venezia", or have a meal at one of the restaurants located here.

Kletterwald. Climbing high in the trees, getting filled with adrenaline, simply fun. You'll get to know your own limits in height, courage and power. Located at "Hochschulstadion" in the southwest of Darmstadt, close to TU Lichtwiese.

Messel Pit. UNESCO World heritage, famous for many fossils found in the pit. A new tourist centre will welcome you on your trip into the past with many lost creatures.

Kai Habermehl and Stefan Ulbrich, TU Darmstadt

Chairman's Column

I'm writing this as I prepare to come to Darmstadt for what promises to be a most exciting meeting, with a great variety of invited and contributed presentations and minisymposia. Let me just point out a couple of sessions you might be interested in attending, as well as giving an idea of what will be discussed at the SIAM Optimization Activity Group business meeting.

The bulletin above mentions the mini-tutorial on Thursday on convex relaxations and applications. Related to this is the SIAG/Opt Outstanding Paper Prize session on Tuesday from 1:30–2:30pm, where the winners, Christine Bachoc from the University of Bordeaux and Frank Vallentin from CWI, Amsterdam, will present their work on new upper bounds on kissing numbers from semidefinite programming. This shows the power of optimization in the analysis of problems in discrete geometry and coding theory. Interestingly, both authors have a background in number theory.

On Thursday from 11:45–12:45pm, there will be a session on future directions in optimization. The panel members come from a wide range of backgrounds — Andreas Griewank, Humboldt University Berlin, Germany; Juan C. Meza, Lawrence Berkeley National Laboratory, USA; Franz Rendl, Universität Klagenfurt, Austria; Claudia Sagastizábal, CEPEL, Brazil; Philippe L. Toint, University of Namur, Belgium; and Andreas Waechter, IBM T.J. Watson Research Center, USA — and their insights should be fascinating. Following this is a panel of representatives of funding agencies, from Canada, Germany, and the US, which should also be of interest to many. All of these sessions, panels, and the business meeting take place in Spectrum A.

Wednesday's business meeting takes place from 6:45–7:15pm. Among the agenda items are:

- Reports on SIAM publications in optimization;
- The SIAG/Opt outstanding paper prize;
- Upcoming related conferences;
- Comments from sister organizations in optimization;
- Discussion of the next SIAM optimization conference OP14; and
- Summary of the current SIAG/Opt membership.

Optimization is now the second largest of SIAM's activity groups! I hope to see at the meeting, or if not, at the conference.

Gute Reise nach Darmstadt!

Michael J. Todd, SIAG/OPT Chair
School of Operations Research and Information Engineering
Cornell University
229 Rhodes Hall
Ithaca, NY 14853
USA
`mjt7@cornell.edu`
`http://people.orie.cornell.edu/~miketodd/todd.html`