Bundle methods for stochastic programs
Level bundle method

Welington de Oliveira

BAS Lecture 26, June 14, 2016, IMPA

Proximal bundle method: convergence analysis

Level bundle method

General formulation

The problem of interest is

$$\min f(x) \quad \text{s.t.} \quad x \in X\,,$$

with

- $f : \Re^n \to \Re$ a convex but nonsmooth function
- $X \subset \Re^n$ a convex set (e.g. $X = \{x \in \Re^n_+ : Ax = b\}$, $X = \Re^n$)

This formulation covers many practical optimization problems, for instance

- Two-stage stochastic programming problems
- Multistage stochastic programming problems

Algorithm: proximal bundle method

$$f_k^M(x) = \max_{j \in \mathcal{B}_k} \{f(x_j) + g_j^\top (x - x_j)\}, \quad x_{k+1} = \arg\min \left\{ f_k^M(x) + \frac{1}{2t_k} \|x - \hat{x}_k\|^2 : \; x \in X \right\}$$

Step 0. Choose $\kappa \in (0,1)$, $t_1 \geq t_{\min} > 0$, $x_1 \in X$ and tolerance $\text{tol} > 0$. Call the oracle to compute $(f(x_1), g_1)$. Define $\hat{x}_1 \leftarrow x_1$, $k \leftarrow 1$, $\mathcal{B}_1 \leftarrow \{1\}$,

Step 1. Solve the QP to obtain x_{k+1}. Define $\hat{g}_k \leftarrow (\hat{x}_k - x_{k+1})/t_k$, $v_k \leftarrow f(\hat{x}_k) - \check{f}_k(x_{k+1})$, and $\hat{e}_k \leftarrow v_k - t_k \|\hat{g}_k\|^2$

Step 2. If $\hat{e}_k \leq \text{tol}$ and $\|\hat{g}_k\| \leq \text{tol}$, stop: $\hat{x}_k$ is an approximate solution

Step 3. Call the oracle to obtain $(f(x_{k+1}), g_{k+1})$

Serious step. **If** $f(x_{k+1}) \leq f(\hat{x}_k) - \kappa v_k$, **then** $\hat{x}_{k+1} \leftarrow x_{k+1}$ and choose $t_{k+1} \geq t_k$

Null step. **Otherwise**, define $\hat{x}_{k+1} \leftarrow \hat{x}_k$ and choose $t_{k+1} \in [t_{\min}, t_k]$

Step 4. Choose $\mathcal{B}_{k+1} \supset \{k+1, k^a\}$

Set $k \leftarrow k+1$ and go back to Step 1.

Convergence analysis

Suppose that tol $= 0$ and the algorithm does not stop. We consider the two possible cases

- either the algorithm generates infinitely many serious steps
- or only finitely many serious steps are generated

Our aim is to show that

$$\lim_{k \in K} (\hat{e}_k, \hat{g}_k) = (0, 0), \qquad \text{for some index set} \quad K \subset \mathbb{N}$$

Recall the important inequality

$$f(\hat{x}_k) \leq f(x) + \hat{e}_k + \|\hat{g}_k\| \|\hat{x}_k - x\| \quad \text{for all } x \in X \text{ and } k\,.$$

We assume that $\inf_X f(x) > -\infty$ and $t_k \geq t_{\min} > 0$

Infinitely many serious steps

Propositon

Suppose the algorithm generates infinitely many serious steps. Then there exists an index set K such that

$$\lim_{k \in K} (\hat{e}_k, \hat{g}_k) = (0, 0) .$$

Finitely many serious steps

There exists $\bar{k} > 1$ such that $\hat{x}_k = \hat{x}$ for all $k \geq \bar{k}$. In addition, $\{t_k\}_{k \geq \bar{k}}$ is non-increasing.

Lemma

$$\lim_{k \to \infty} [f(x_k) - f^M_{k-1}(x_k)] = 0\,.$$

Proof: *Convex proximal bundle methods in depth: a unified analysis for inexact oracles.*
Mathematical Programming, 2014, Volume 148, Issue 1-2, pp 241-277
W. de Oliveira, C Sagastizábal and C. Lemaréchal.

Propositon

Define $K = \{k : k \geq \bar{k}\}$. *Then,*

$$\lim_{k \in K} (\hat{e}_k, \hat{g}_k) = 0\,.$$

Convergence analysis

Theorem

Consider the proximal bundle algorithm applied to the problem

$$(P) \quad f^* := \min_{x \in X} f(x)\,.$$

Then,

- *there exists* K *such that* $\lim_{k \in K}(\hat{e}_k, \hat{g}_k) = 0$
- $\{f(\hat{x}_k)\} \downarrow f^*$
- *all accumulation points (if any) of* $\{\hat{x}_k\}$ *is a solution to (P)*

One can show that when (P) has a solution, then the whole sequence $\{\hat{x}_k\}$ converges to a solution.

Bundle methods

Main ingredients

(i) a convex model $f_k^M \leq f$ (eg. cutting-plane model)

(ii) a stability center $\hat{x}_k$ (eg.: the best point so far)

(iii) a parameter t_k (or f_k^{lev}) to be updated at every iteration

The next trial point x_{k+1} of a bundle method depends on the above 3 ingredients, whose organization define different methods:

Proximal bundle method $(t_k > 0)$

$$x_{k+1} := \arg\min \left\{ f_k^M(x) + \frac{1}{2t_k}\|x - \hat{x}_k\|^2 : \ x \in X \right\} .$$

Level bundle method $(f_k^{\text{lev}} \in \Re)$

$$x_{k+1} := \arg\min \left\{ \frac{1}{2}\|x - \hat{x}_k\|^2 : \ f_k^M(x) \leq f_k^{\text{lev}}, \ x \in X \right\} .$$

Relation between proximal and level bundle methods

Depending on parameters t_k and f_k^{lev}, the solution of the proximal bundle subproblem

$$\min\left\{ f_k^M(x) + \frac{1}{2t_k}\|x - \hat{x}_k\|^2 :\ x \in X \right\}$$

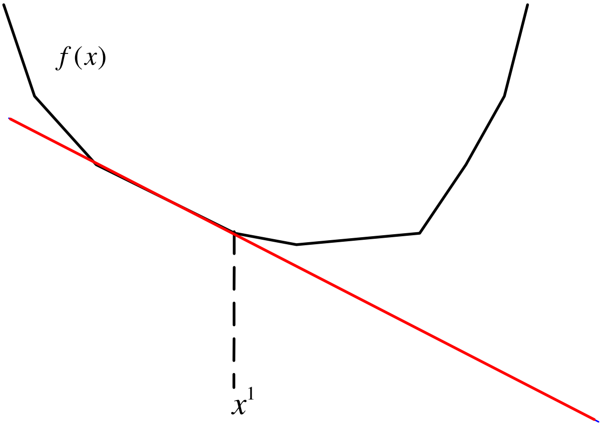
is also a solution of the level bundle subproblem

$$\min\left\{ \frac{1}{2}\|x - \hat{x}_k\|^2 :\ f_k^M(x) \leq f_k^{\text{lev}},\ x \in X \right\}.$$

The prox-parameter $t_k > 0$ can be seen as a Lagrange multiplier associate to the level constraint $f_k^M(x) \leq f_k^{\text{lev}}$

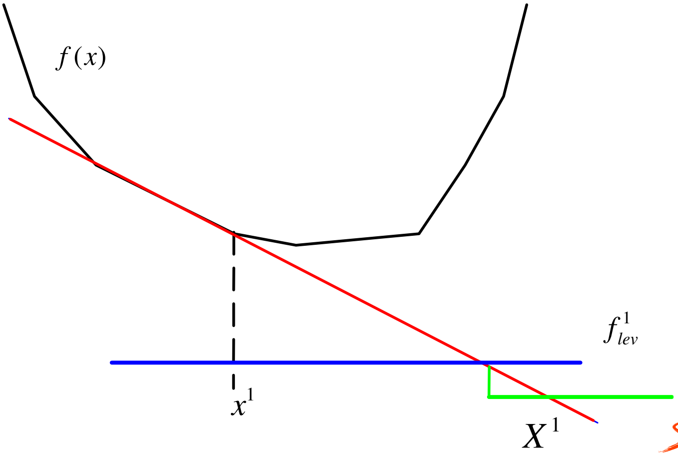
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \leq f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



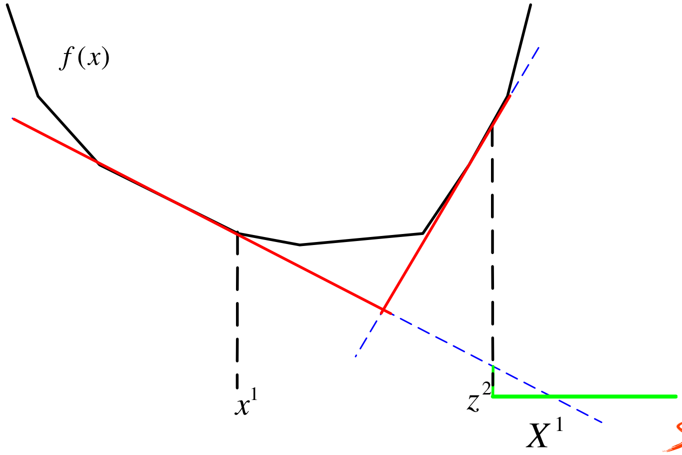
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\rm lev}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



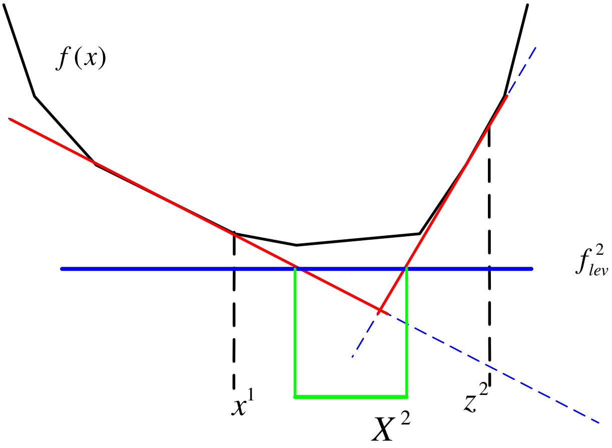
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \leq f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



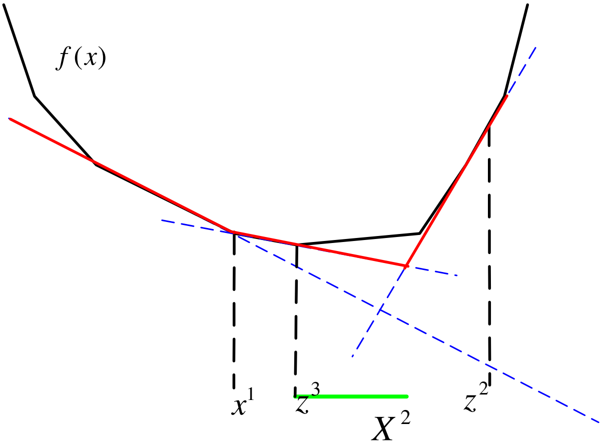
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \leq f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



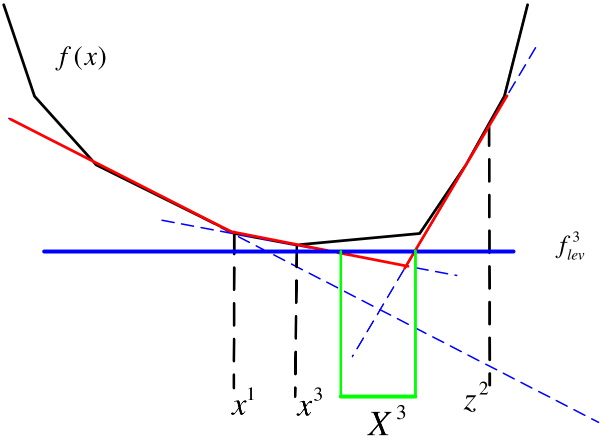
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\rm lev}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



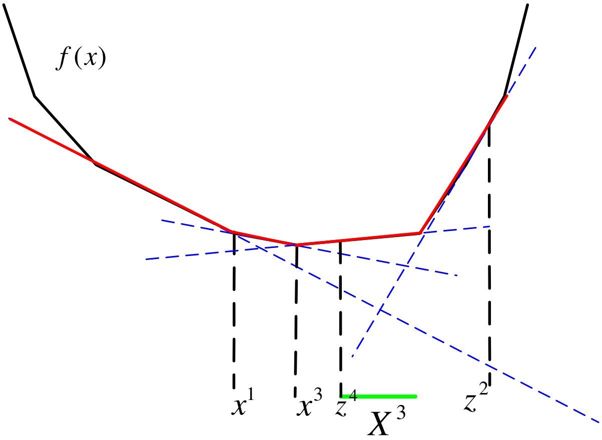
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



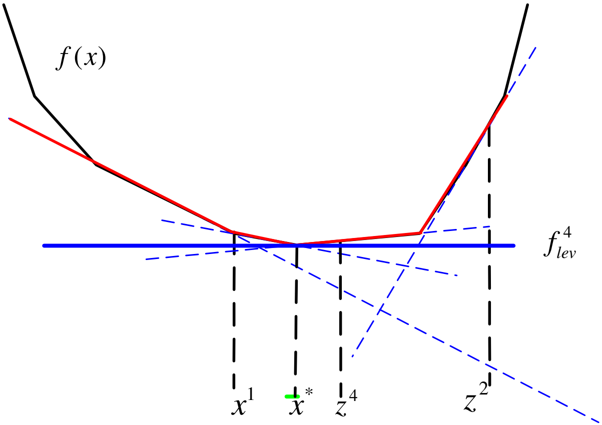
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \leq f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



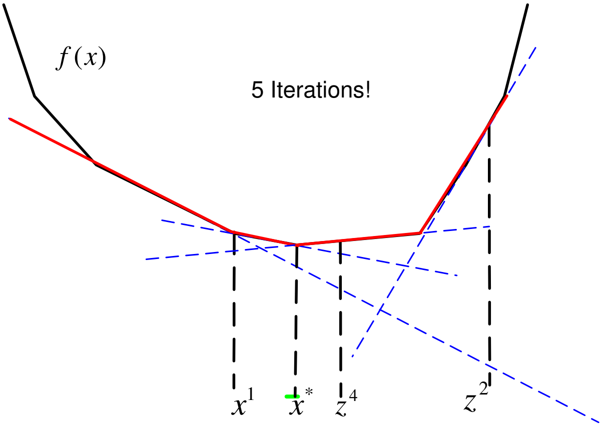
Level bundle method

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



Level bundle method

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\text{lev}}\}, \qquad x_{k+1} := \texttt{Proj}_{X^k}(\hat{x}_k)$$



The first level bundle method was proposed in 1995 by C. Lemaréchal, Nemiroviski and Nesterov. In the same year C. Kiwiel, proposed a limited-memory level bundle method.

There are several level bundle methods with different rules for

- updating the stability center $\hat{x}_k$
  - $\hat{x}_k = x_k$ for all k
  - $\hat{x}_k = x_1$ for all k
  - the serious step rule can also be employed
- updating the level parameter f_k^{lev}
  - for instance, by solving an extra LP

In addition, depending on the feasible set X a level bundle method can be preferable to others

- Level bundle methods for oracles with on-demand accuracy. Optimization Methods & Software, 2014, W. de Oliveira and C. Sagastizábal.
- Level bundle-like algorithms for convex optimization. Journal of Global Optimization, 2014. J. Y. Bello Cruz and W. de Oliveira.
- Regularized optimization methods for convex MINLP problems. TOP (Operations Research & Decision Theory), 2016, W. de Oliveira.

Level bundle method

Let's focus on a simple variant of the method

Assumption

$$(P) \quad \min_{x \in X} f(x),$$

where X is a convex and compact set, having a simple structure (eg. X is a bounded polyhedron).

Remark: there are level bundle methods able to deal with a unbounded feasible sets X

We employ the same kind of model used in the proximal bundle method:

$$f_k^M(x) := \max_{j \in \mathcal{B}_k} \{ f(x_j) + \langle g_j, x - x_j \rangle \}$$

with $\mathcal{B}_k$ a given index subset.

Next iterate

$$X^k := \{x \in X : f_k^M(x) \le f_k^{\text{lev}}\}, \qquad x_{k+1} := \mathtt{Proj}_{X^k}(\hat{x}_k)$$

Propositon

Suppose that X is polyhedral or $\mathtt{ri}(X) \neq \emptyset$. The point x_{k+1} is the projection of $\hat{x}_k \notin X^k$ onto X^k if and only if $x_{k+1} \in X$, $f_k^M(x_{k+1}) \le f_k^{\text{lev}}$ and there exist $p_f^k \in \partial f_k^M(x_{k+1})$, $p_X^k \in \partial i_X(x_{k+1})$ and a stepsize $t_k > 0$, such that

$$x_{k+1} = \hat{x}_k - t_k \hat{g}_k \quad \text{and} \quad t_k(f_k^M(x_{k+1}) - f_k^{\text{lev}}) = 0\,,$$

where $\hat{g}_k = p_X^k / t_k + p_f^k$. In addition, the aggregate linearization

$f_{k^a}^L(\cdot) := f_k^M(x_{k+1}) + \langle \hat{g}_k, \cdot - x_{k+1}\rangle$ satisfies $f_{k^a}^L(x) \le f_k^M(x) \le f(x)$ for all $x \in X$.

Besides, for the aggregate level set $X^{k^a} := \{x \in X : f_{k^a}^L(x) \le f_k^{\text{lev}}\}$, we have that

$$P_{X^{k^a}}(\hat{x}_k) = P_{X^k}(\hat{x}_k)\,.$$

Algorithm

$$f_k^M(x) = \max_{j \in \mathcal{B}_k} \{f(x_j) + \langle g_j, x - x_j \rangle\}, \quad x_{k+1} = \arg\min \left\{ \frac{1}{2}\|x - \hat{x}_k\|^2 : \check{f}_k(x) \le f_k^{\text{lev}}, \ x \in X \right\}$$

Step 0. Choose $\gamma \in (0, 1)$ and tol > 0. Given x_1 and $f_0^{\text{low}} \le f^*$, call the oracle to compute $(f(x_1), g_1)$. Define $f_1^{\text{up}} \leftarrow f(x_1)$, $\hat{x}_1 \leftarrow x_1$, $k \leftarrow 1$, $\mathcal{B}_1 \leftarrow \{1\}$

Step 1. Set $\Delta_k \leftarrow f_k^{\text{up}} - f_k^{\text{low}}$. If $\Delta_k \le$ tol, stop.

Step 2. Choose a stability center $\hat{x}_k$ (e.g., $\hat{x}_k \leftarrow \hat{x}_{k-1}$)

Step 3. Set $f_k^{\text{lev}} = f_k^{\text{low}} + \gamma\Delta_k$ and try to obtain x_{k+1}. If the QP is infeasible, set $f_k^{\text{low}} \leftarrow f_k^{\text{lev}}$ and go back to Step 1.

Step 4. Call the oracle to compute $(f(x_{k+1}), g_{k+1})$. Set $f_k^{\text{up}} \leftarrow \min\{f_k^{\text{up}}, f(x_{k+1})\}$

Step 5. Set $\mathcal{B}_{k+1} \leftarrow \mathcal{B}_k \cup \{k+1\}$, $f_{k+1}^{\text{low}} \leftarrow f_k^{\text{low}}$, $k \leftarrow k+1$ and go back to Step 1.

Some comments

The given algorithm updates the lower bound f_k^{low} only when the QP is infeasible

Lower bound updating

Although more expensive, a more efficient updating rule consists in solving the following LP at Step 5:

$$f_{k+1}^{\text{low}} \leftarrow \min \{ f_{k+1}^M(x) \quad \text{s.t.} \quad f_{k+1}^M(x) \geq f_k^{\text{low}}, \quad x \in X \}$$

This rule is advisable when the oracle is more expensive than solving the above LP (eg., in stochastic programs).

Some comments

Bundle compression

Similarly to the proximal bundle method, Step 5 can reduce the bundle of information:

Step 5. Choose $\mathcal{B}_{k+1} \supset \{k+1, k^a\}$. Set $k \leftarrow k+1$ and go back to Step 1.

Remember: $f^L_{k^a}(x) = f^M_k(x_{k+1}) + \langle \hat{g}_k, x - x_{k+1} \rangle$

Stability center rule

An efficient for Step 2 is the following one: set $k(l) = 0$ and $l = 0$ in Step 0. Then:

Step 2. If $\Delta_k \leq (1-\gamma)\Delta_{k(l)}$, set $k(l) \leftarrow k$, $l \leftarrow l+1$ and set $\hat{x}_k$ as the best candidate x_j. Otherwise set $\hat{x}_k \leftarrow \hat{x}_{k-1}$

Some comments

This rule ensures that

- stability center are fixed along cycles $k(l) \leq j < k(l+1)$, $l = 1, 2, \ldots,$
- $\Delta_{k(l+1)} \leq (1-\gamma)\Delta_{k(l)}$ for all l
- $\Delta_{k(l+1)} \leq (1-\gamma)^l \Delta_{k(1)}$ for all l

Furthermore, $\lim_{l\to\infty} \Delta_{k(l)} = 0$.

If we show that $l \uparrow \infty$, then we have convergence!