Bundle methods for stochastic programs

Proximal bundle method

Welington de Oliveira

BAS Lecture 25, June 9, 2016, IMPA

General formulation

In this part of the course we will focus on efficient optimization methods to solve convex programs of the form

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

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

Two-stage stochastic linear programming

In two-stage stochastic linear programming problems with finitely many scenarios $\xi^i = (q^i, T^i, W^i, h^i)$ we wish to solve the high dimensional LP

$$\left\{ \begin{array}{ll} \min & c^\top x + \sum_{i=1}^N p_i [{q^i}^\top y^i] \\ \text{s.t.} & Ax = b,\ x \geq 0 \\ & T^i x + W^i y^i = h^i,\ y^i \geq 0, \quad i = 1, \ldots, N \end{array} \right.$$

Two-stage stochastic linear programming

In two-stage stochastic linear programming problems with finitely many scenarios $\xi^i = (q^i, T^i, W^i, h^i)$ we wish to solve the high dimensional LP

$$\left\{\begin{array}{ll} \min & c^\top x + \sum_{i=1}^N p_i [{q^i}^\top y^i] \\ \text{s.t.} & Ax = b,\ x \geq 0 \\ & T^i x + W^i y^i = h^i,\ y^i \geq 0, \quad i = 1, \ldots, N \end{array}\right.$$

Two-stage decomposition

$$\min\ f(x) \quad \text{s.t.} \quad x \in X\,, \qquad \text{with} \qquad f(x) := c^\top x + \sum_{i=1}^{N} p_i Q(x, \xi^i)\,,$$

$$Q(x,\xi) = \left\{\begin{array}{ll} \min & q^\top y \\ \text{s.t.} & Wy = h - Tx \\ & y \geq 0\,. \end{array}\right. \qquad \text{and} \qquad X := \{x \in \Re^n_+ : Ax = b\}$$

We know that $g = c - \sum_{i=1}^N p_i {T^i}^\top \pi^i \in \partial f(x)$, where π^i is a dual solution of $Q(x, \xi^i)$

Multistage stochastic linear programs

$$\min_{\substack{A_1 x_1 = b_1 \\ x_1 \geq 0}} c_1^\top x_1 + \mathbb{E}\left[\min_{\substack{B_2 x_1 + A_2 x_2 = b_2 \\ x_2 \geq 0}} c_2^\top x_2 + \mathbb{E}\left[\cdots + \mathbb{E}[\min_{\substack{B_T x_{T-1} + A_T x_T = b_T \\ x_T \geq 0}} c_T^\top x_T]\right]\right]$$

- Some elements of the data $\xi = (c_t, B_t, A_t, b_t)$ depend on uncertainties.

By assuming finitely many scenarios and dualizing the nonantecipativity constraints (that can be written as $Gx = 0$) we get

Multistage stochastic linear programs

(See Lecture 17)

Dual problem

$$\min_u \; f(u)\,, \quad \text{with} \quad f(u) := -\sum_{i=1}^{N} D^i(u)$$

$$D^i(u) := \begin{cases} \min_{x^i} & p_i \sum_{t=1}^{T} (c_t^i)^\top x_t^i + u^\top G^i x^i \\ \text{s.t.} & A_1 x_1 = b_1 \\ & B_t^i x_{t-1}^i + A_t^i x_t^i = b_t^k, \quad t = 2, \ldots, T \\ & x_t^i \geq 0\,. \end{cases}$$

Computing $f(u)$ for each given u amounts to solving N LPs.

We know that $g = -Gx(u) \in \partial f(u)$, where $x(u) = (x^1(u), \ldots, x^N(u))$ and $x^i(u)$ is a solution of $D^i(u)$

Let's stick with the more compact and general formulation

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

with $f : \Re^n \to \Re$ a convex but nonsmooth function and $X \subset \Re^n$ a convex set.

We'll assume the availability of an oracle providing us with first-order information on f:

$$x \longrightarrow \boxed{\textbf{Oracle}} \longrightarrow \left\{ \begin{array}{ll} \text{function value} & f(x) \\ \text{subgradient} & g \in \partial f(x) \end{array} \right.$$

In stochastic programming, the oracle should be smart enough to use parallel computing:

- the oracle consists of solving N optimization subproblems to compute $f(x)$ and a subgradient g
- most of time dedicate to minimize f is spent in the oracle!

Therefore, subgradient and (pure) cutting-plane methods are not very efficient[1]...

[1] These methods require, in general, many oracle calls.

Cutting-plane method

Consider the problem

$$\min_{x \in X} f(x)$$

and suppose that X is a compact set.

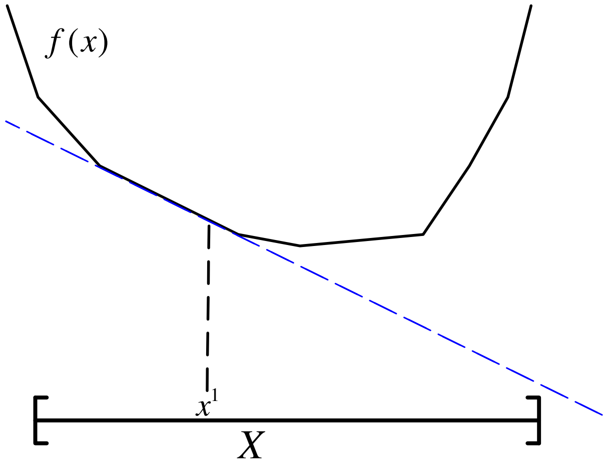
Algorithm

1. Given $x_0 \in X$, call the oracle to compute $f(x_0)$ and $g_0 \in \partial f(x_0)$. Set $f_0^{\mathrm{up}} = f(x_0)$ and $k = 0$
2. (iterate) Find $x_{k+1} = \arg\min_{x \in X} \check{f}_k(x)$. Let $f_k^{\mathrm{low}} = \check{f}_k(x_{k+1})$.
3. (stopping test) If $f_k^{\mathrm{up}} - f^{\mathrm{low}}$ is small enough, stop.
4. (oracle) Compute $f(x_{k+1})$, $g_{k+1} \in \partial f(x_k)$ and set $f_{k+1}^{\mathrm{up}} = \min\{f(x_{k+1}), f_k^{\mathrm{up}}\}$.
5. (loop) Set $k \leftarrow k + 1$ and go back to Step 2.

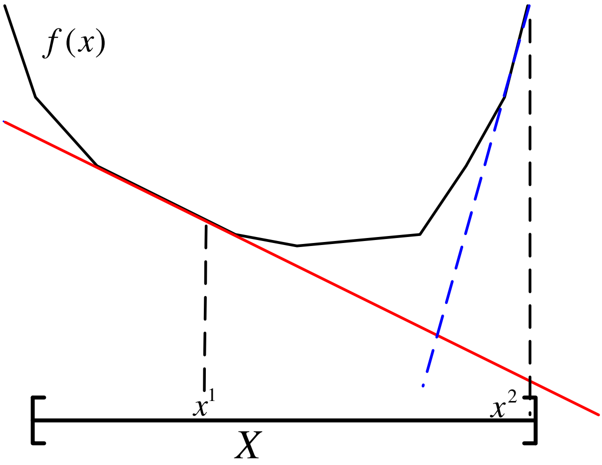
Cutting-plane model

$$\check{f}_k(\cdot) = \max_{j=1,\ldots,k} \{f(x_j) + g_j^{\top}(\cdot - x_j)\}$$

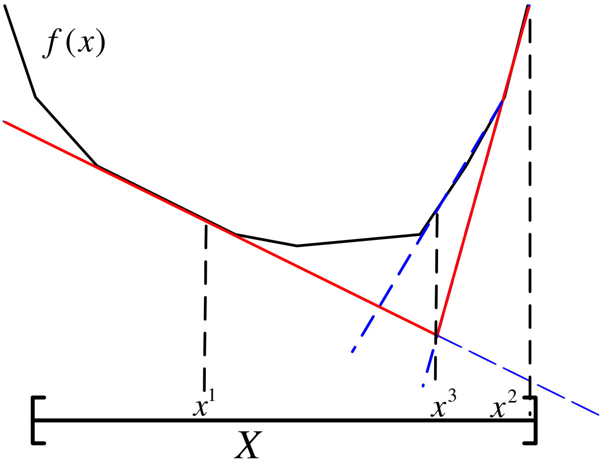
Cutting-plane method



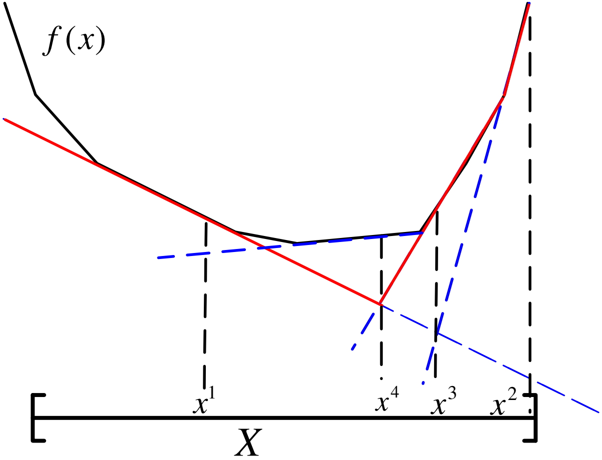
Cutting-plane method



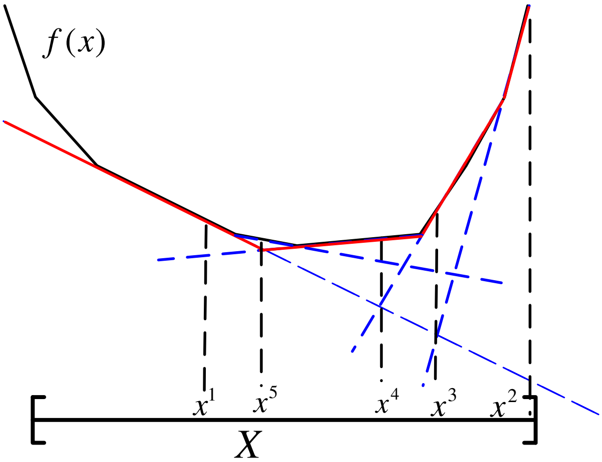
Cutting-plane method



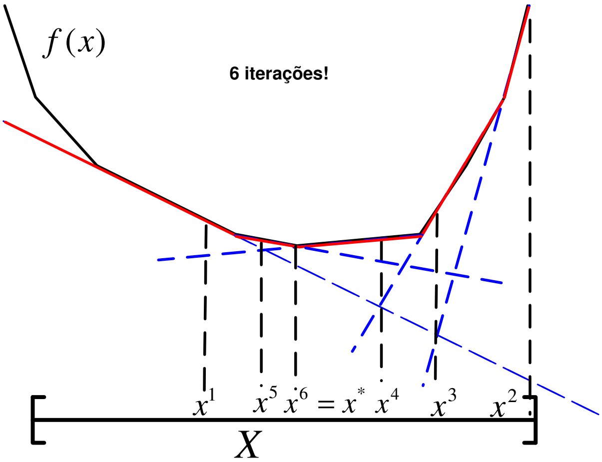
Cutting-plane method



Cutting-plane method



Cutting-plane method



Cutting-plane method

The method requires solving a LP at each iteration

$$x_{k+1} = \arg\min_{x \in X} \check{f}_k(x), \quad \check{f}_k(\cdot) = \max_{j=1,\ldots,k} \{f(x_j) + {g_j}^\top(\cdot - x_j)\}$$

that is equivalent to

$$\begin{cases} \min_{x,r} & r \\ s.t. & f(x_j) + {g_j}^\top(x - x_j) \le r, \quad j = 1, \ldots, k \\ & x \in X, \; r \in \Re. \end{cases}$$

A new constraint is added at each iteration!

Cutting-plane method

Pros × Cons

- only computes a single subgradient per iteration
- easy to code
- easy and reliable stopping test

- $f(x_{k+1}) \not< f(x_k)$ (it is not a descent method)
- instable and has low convergence rate
- requires compactness of the feasible set
- doesn't exploit good starting points
- subproblem becomes heavier and heavier...

The *Regularized Decomposition Method* (1986) for 2-SLP address some of the above drawbacks.
Regularized Decomposition Method is just a particular case of (proximal) Bundle Methods!

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 $(t_k > 0)$

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

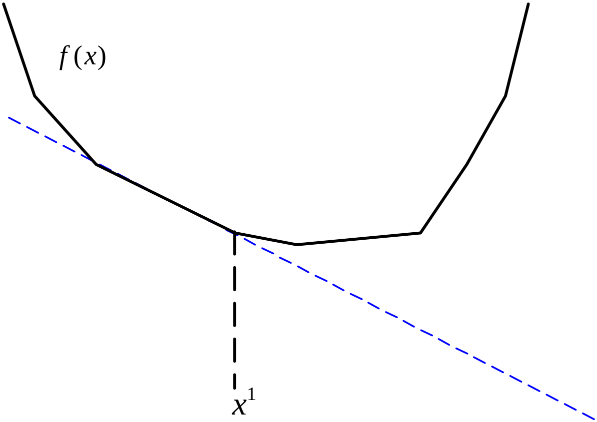
Level bundle method $(f_k^{\text{lev}} \in \Re)$

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

Today we focus on proximal bundle method!

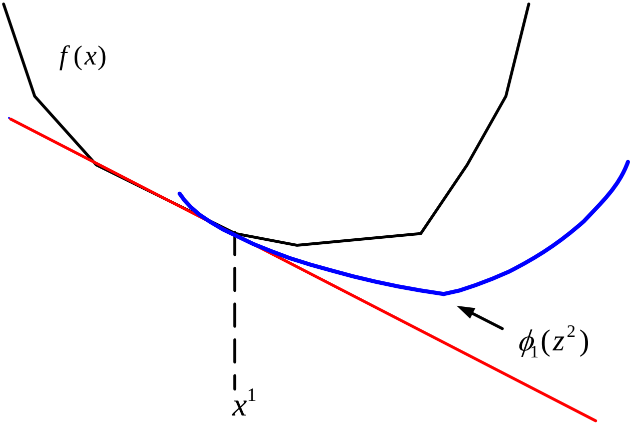
Proximal bundle method

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



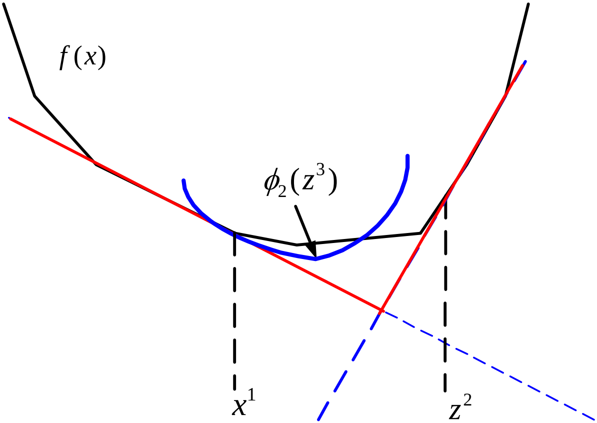
Proximal bundle method

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



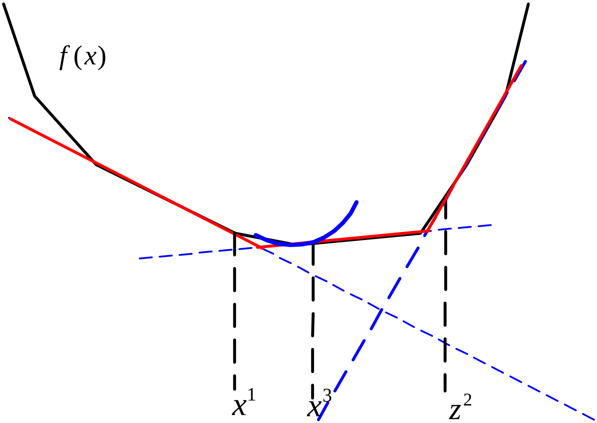
Proximal bundle method

$$f^M \equiv \check{f}, \qquad x_{k+1} := \arg\min\left\{\check{f}_k(x) + \frac{1}{2t_k}\|x - \hat{x}_k\|^2 \, : \; x \in X\right\}$$



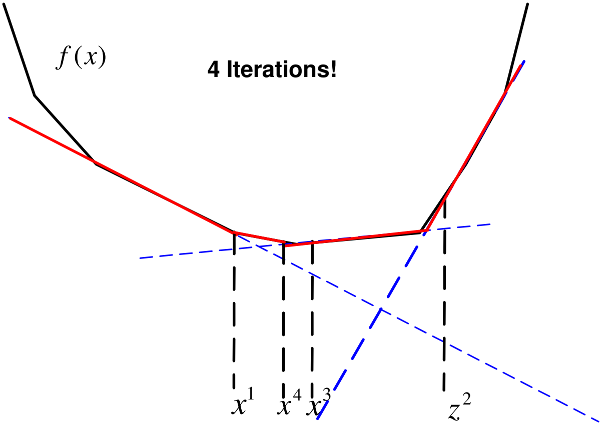
Proximal bundle method

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



Proximal bundle method

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



Proximal bundle method

Pros × Cons

- only computes a single subgradient per iteration
- easy and reliable stopping test
- stable
- does not require X to be compact
- it is a descent method
- exploit good-quality initial points
- subproblem defining x_{k+1} can be kept small

Proximal bundle method

Pros × Cons

- only computes a single subgradient per iteration
- easy and reliable stopping test
- stable
- does not require X to be compact
- it is a descent method
- exploit good-quality initial points
- subproblem defining x_{k+1} can be kept small

- convergence analysis is more involving...

Proximal bundle method

Let's consider a more economical model:

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

- The cutting-plane method takes $\mathcal{B}_k := \{1, 2, \ldots, k\}$. We will consider $\mathcal{B}_k \subset \{1, 2, \ldots, k\}$ (or something a bit different)
- The method generates a sequence of trial points $\{x_k\} \subset X$ by solving a QP:

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

Solving the QP subproblem

The QP

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

can be rewritten as

$$\left\{\begin{array}{ll}\min_{x,r} & r+\frac{1}{2t_k}\|x-\hat{x}_k\|^2\\ \text{s.a} & f(x_j)+g_j^\top(x-x_j)\le r,\quad j\in\mathcal{B}_k\\ & x\in X,\ r\in\Re\,.\end{array}\right.$$

We can apply specialized softwares.

Proximal bundle method

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

A rule decide when to update the stability center $\hat{x}_k$. Such rule depends on the predicted decrease by the model f_k^M

$$v_k = f(\hat{x}_k) - f_k^M(x_{k+1})$$

and a constant $\kappa \in (0, 1)$:

- Serious step: if $f(x_{k+1}) \leq f(\hat{x}_k) - \kappa v_k$, then

$$\hat{x}_{k+1} \leftarrow x_{k+1}$$

- Null step: if $f(x_{k+1}) > f(\hat{x}_k) - \kappa v_k$, then

$$\hat{x}_{k+1} \leftarrow \hat{x}_k$$

The serious-step sequence $\{\hat{x}_k\}$ is a subsequence of $\{x_k\}$

Next iterate

Lemma

Suppose that X is a polyhedron or $\boldsymbol{ri}(X) \neq \emptyset$. Then

$$x_{k+1} = \hat{x}_k - t_k \hat{g}_k \quad com \quad \hat{g}_k = p_f^k + p_X^k \,,$$

where $p_f^k \in \partial f_k^M(x_{k+1})$ and $p_X^k \in \partial i_X(x_{k+1})$.
(i_X is the indicator function of X.)

Furthermore, the affine function

$$f_{k^a}^L(x) := f_k^M(x_{k+1}) + \langle \hat{g}_k, x - x_{k+1} \rangle$$

is a lower approximation for the model f_k^M:

$$f_{k^a}^L(x) \leq f_k^M(x) \quad \forall\, x \in X.$$

Optimality measure

Propositon

Let the predicted decrease and aggregate linearization error defined by

$$v_k := f(\hat{x}_k) - f_k^M(x_{k+1}) \quad \textit{and} \quad \hat{e}_k := f(\hat{x}_k) - f_{k^a}^L(\hat{x}_k)\,.$$

Then,

$$\hat{e}_k \geq 0, \quad \hat{e}_k + t_k \|\hat{g}_k\|^2 = v_k \geq 0 \quad \textit{for all} \quad k.$$

Furthermore

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

$$\boxed{\text{If } (\hat{e}_k, \hat{g}_k) = 0, \text{ then } \hat{x}_k \text{ is solution to the problem}}$$

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 $\mathrm{tol} > 0$. 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 $\hat{e}_k \leq \mathrm{tol}$ and $\|\hat{g}_k\| \leq \mathrm{tol}$, 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.

Some comments

- Only 2 linearizations are required: f_k^L and $f_{k^a}^L$, i.e.,

$$\mathcal{B}_{k+1} = \{k+1, k^a\} \quad \text{suffices!}$$

- the prox-parameter t_k is non-increasing along null steps
- a simple heuristic to update the prox-parameter is the following
  - compute $t_{\mathtt{aux}} := t_k \left(1 + \frac{(g_{k+1}-g_k)^\top (x_{k+1}-x_k)}{\|g_{k+1}-g_k\|^2}\right)$
  - if null step: $t_{k+1} \leftarrow \min\{t_k,\ \max\{t_{\mathtt{aux}}, t_k/2, t_{\min}\}\}$
  - if serious step: $t_{k+1} \leftarrow \max\{t_k,\ \min\{t_{\mathtt{aux}}, 10t_k\}\}$
- it is advisable to consider different tolerances for the measures $\hat{e}_k$ and $\hat{g}_k$
- the sequence $\{f(\hat{x}_k)\}$ is non-increasing
- any accumulation point of $\{\hat{x}_k\}$ is a solution to the problem