Inexact Oracles in NonDifferentiable Optimization: Deflected Conditional Subgradient Methods and Generalized Bundle Methods

Antonio Frangioni

Dipartimento di Informatica, Università di Pisa

48^{th} Workshop on Nonsmooth Analysis Optimization and Applications
Dedicated to V.F. Demyanov
Erice, May 14, 2008

Outline

1. Introduction, Motivation

Outline

Outline

1. Introduction, Motivation
2. Subgradient methods: introduction
3. Polyak-type stepsize: the abstract case

Outline

Outline

1. Introduction, Motivation
2. Subgradient methods: introduction
3. Polyak-type stepsize: the abstract case
4. Polyak-type stepsize: the implementable case
5. Deflection-restricted rules

Outline

1. Introduction, Motivation
2. Subgradient methods: introduction
3. Polyak-type stepsize: the abstract case
4. Polyak-type stepsize: the implementable case
5. Deflection-restricted rules
6. Bundle methods

Outline

Lagrangian Relaxation

- Difficult structured problem

$$z(P) = \sup_u \{ c(u) : h(u) \leq 0 , u \in U \} \quad (1)$$

with complicating constraints $h(u) \leq 0$ over easy set U

[1] Lemaréchal, Renaud "A geometric study of duality gaps, with applications", Math. Prog., 2001

Lagrangian Relaxation

- Difficult structured problem

$$z(P) = \sup_u \{ c(u) : h(u) \leq 0 , u \in U \} \tag{1}$$

with complicating constraints $h(u) \leq 0$ over easy set U

- Assume Lagrangian relaxation of complicating constraints easy

$$f(x) = \sup_u \{ c(u) + xh(u) : u \in U \} \tag{2}$$

[1] Lemaréchal, Renaud "A geometric study of duality gaps, with applications", Math. Prog., 2001

Lagrangian Relaxation

- Difficult structured problem

$$z(P) \;=\; \sup_u \{\; c(u) \;:\; h(u) \leq 0 \;,\; u \in U \;\} \qquad (1)$$

with complicating constraints $h(u) \leq 0$ over easy set U

- Assume Lagrangian relaxation of complicating constraints easy

$$f(x) \;=\; \sup_u \{\; c(u) + xh(u) \;:\; u \in U \;\} \qquad (2)$$

- f convex $\Rightarrow$ corresponding Lagrangian dual easy

$$z(\Pi) \;=\; \inf_x \{\; f(x) \;:\; x \geq 0 \;\}$$

[1] Lemaréchal, Renaud "A geometric study of duality gaps, with applications", Math. Prog., 2001

Lagrangian Relaxation

- Difficult structured problem

$$z(P) \;=\; \sup_u \{\ c(u)\ :\ h(u) \leq 0\ ,\ u \in U\ \} \tag{1}$$

with complicating constraints $h(u) \leq 0$ over easy set U

- Assume Lagrangian relaxation of complicating constraints easy

$$f(x) \;=\; \sup_u \{\ c(u) + xh(u)\ :\ u \in U\ \} \tag{2}$$

- f convex $\Rightarrow$ corresponding Lagrangian dual easy

$$z(\Pi) \;=\; \inf_x \{\ f(x)\ :\ x \geq 0\ \}$$

- Equivalent to primal relaxation

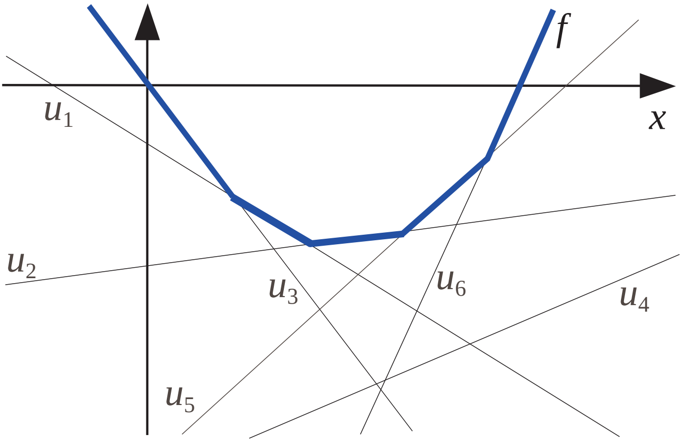
$$\sup \{\ v\ :\ (u, v, 0) \in \mathcal{U}^{**}\ \} \tag{3}$$

where $\mathcal{U} = \{\ (u, v, r)\ :\ u \in U\ ,\ v \leq c(u)\ ,\ r \geq h(u)\ \}$

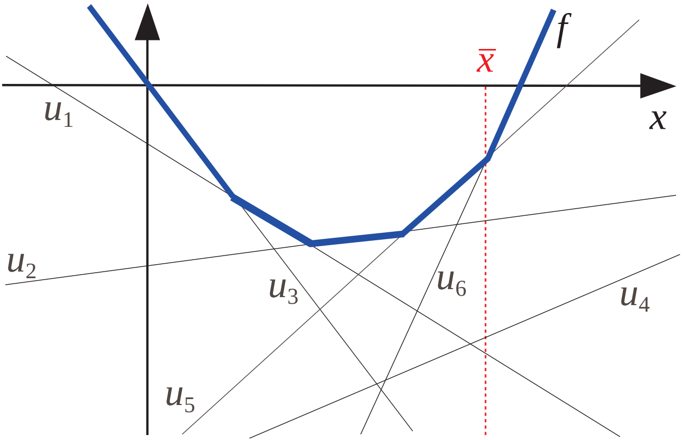
(a more palatable object if problem "affine enough")[1]

[1] Lemaréchal, Renaud "A geometric study of duality gaps, with applications", Math. Prog., 2001

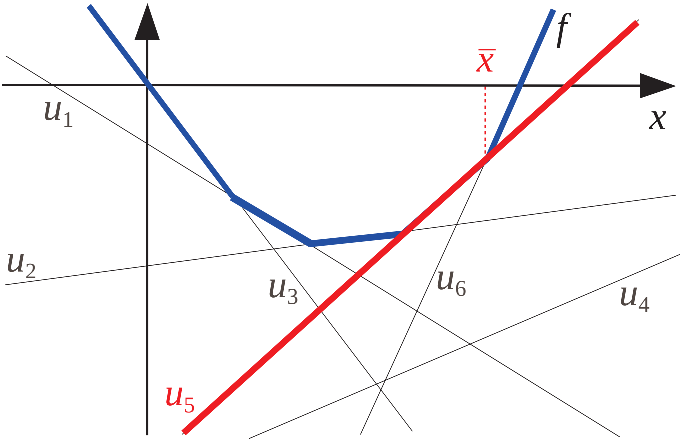
Lagrangian Relaxation (graphically)



Lagrangian Relaxation (graphically)

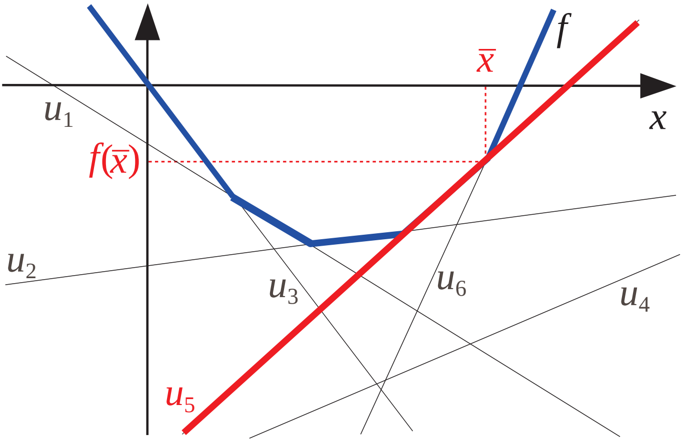


Lagrangian Relaxation (graphically)



- Oracle to (efficiently) perform the maximization (structure inside)

Lagrangian Relaxation (graphically)



- Oracle to (efficiently) perform the maximization (structure inside)
- Solving exactly (2) provides both function value and subgradient

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]

2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy"
   $\Rightarrow$ safe (and effective) stopping criterion

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]
2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy"
   $\Rightarrow$ safe (and effective) stopping criterion

- Trade off: "difficult" (2) $\Rightarrow$ "good bound"[3]

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]
2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy" $\Rightarrow$ safe (and effective) stopping criterion

- Trade off: "difficult" (2) $\Rightarrow$ "good bound"[3]
- Enumerative approaches: do this at each of very many nodes

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]
2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy"
   $\Rightarrow$ safe (and effective) stopping criterion

- Trade off: "difficult" (2) $\Rightarrow$ "good bound"[3]
- Enumerative approaches: do this at each of very many nodes
- (Π) has to be (approximately) solved very efficiently $=$
  fast convergence $+$ low iteration cost

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

1. Primal "continuous" solutions useful to drive heuristics for (1)[2]

2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy"
   $\Rightarrow$ safe (and effective) stopping criterion

- Trade off: "difficult" (2) $\Rightarrow$ "good bound"[3]
- Enumerative approaches: do this at each of very many nodes
- (Π) has to be (approximately) solved very efficiently $=$
  fast convergence $+$ low iteration cost
- It thus makes sense to solve (2) approximately

[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

Lagrangian Relaxation: What For?

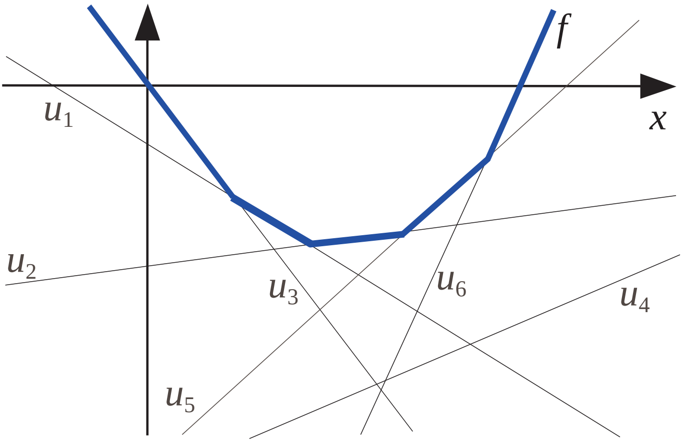
1. Primal "continuous" solutions useful to drive heuristics for (1)[2]
2. Mainly upper bounding: $z(\Pi) \geq z(P)$, "near" if (2) "not too easy"
   $\Rightarrow$ safe (and effective) stopping criterion

- Trade off: "difficult" (2) $\Rightarrow$ "good bound"[3]
- Enumerative approaches: do this at each of very many nodes
- (Π) has to be (approximately) solved very efficiently =
  fast convergence + low iteration cost
- It thus makes sense to solve (2) approximately
- Which may mean different things

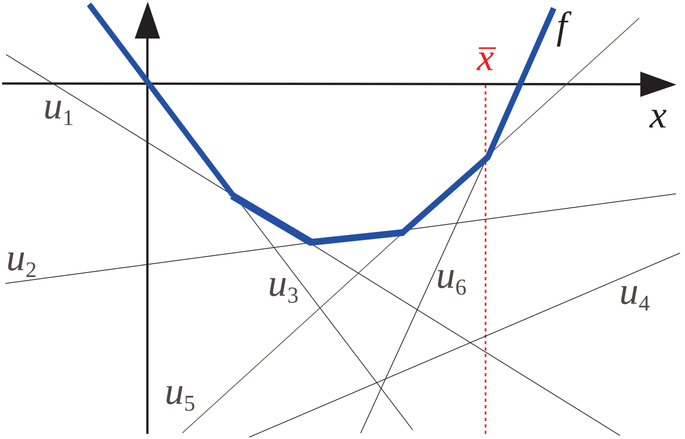
[2] F., Gentile, Lacalandra "Solving Unit Commitment Problems with General Ramp Contraints", IJEPES, 2008

[3] F. "About Lagrangian Methods in Integer Optimization", Ann. O.R., 2005

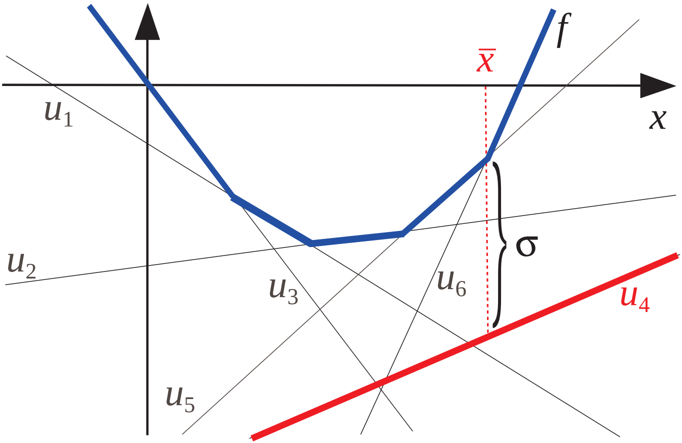
Approximate Lagrangian Relaxation I (graphically)



Approximate Lagrangian Relaxation I (graphically)

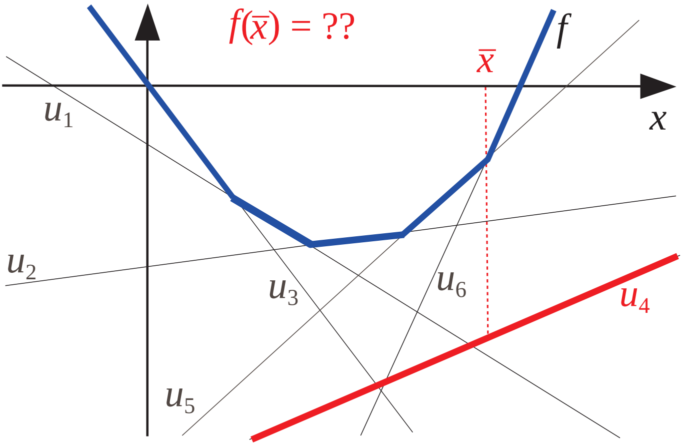


Approximate Lagrangian Relaxation I (graphically)



- Approximate solution ⇒ σ-subgradient, $\sigma \geq 0$

Approximate Lagrangian Relaxation I (graphically)



- Approximate solution ⇒ σ-subgradient, $\sigma \geq 0$
- Heuristics ⇒ no measure of σ ⇒ useless for bounding purposes

Approximate Lagrangian Relaxation II

- Heuristics have no (or too weak in practice) performance guarantee
- Different approach: an exact algorithm for solving (2)

[4] Beltran, Tadonki, Vial "Solving the p-Median Problem with a Semi-Lagrangian Relaxation", COAP, 2006

Approximate Lagrangian Relaxation II

- Heuristics have no (or too weak in practice) performance guarantee
- Different approach: an exact algorithm for solving (2)
- Three main components:
  - a heuristic producing $\bar{u} \in U \Rightarrow c(\bar{u}) + xh(\bar{u}) \leq f(x)$
  - an upper bound $\bar{f}(x) \geq f(x)$ (further relaxation)
  - enumeration to squeeze the two together (branching)
- Iterative process where $c(\bar{u}) + xh(\bar{u}) \rightarrow f(x) \leftarrow \bar{f}(x)$

[4] Beltran, Tadonki, Vial "Solving the p-Median Problem with a Semi-Lagrangian Relaxation", COAP, 2006

Approximate Lagrangian Relaxation II

- Heuristics have no (or too weak in practice) performance guarantee
- Different approach: an exact algorithm for solving (2)
- Three main components:
  - a heuristic producing $\bar{u} \in U \Rightarrow c(\bar{u}) + xh(\bar{u}) \leq f(x)$
  - an upper bound $\bar{f}(x) \geq f(x)$ (further relaxation)
  - enumeration to squeeze the two together (branching)
- Iterative process where $c(\bar{u}) + xh(\bar{u}) \rightarrow f(x) \leftarrow \bar{f}(x)$
- (2) "as difficult" as (1) in theory (but largely less so in practice[4])
- The gap $\sigma = \bar{f}(x) - c(\bar{u}) - xh(\bar{u}) \geq 0$ may decrease rather slowly

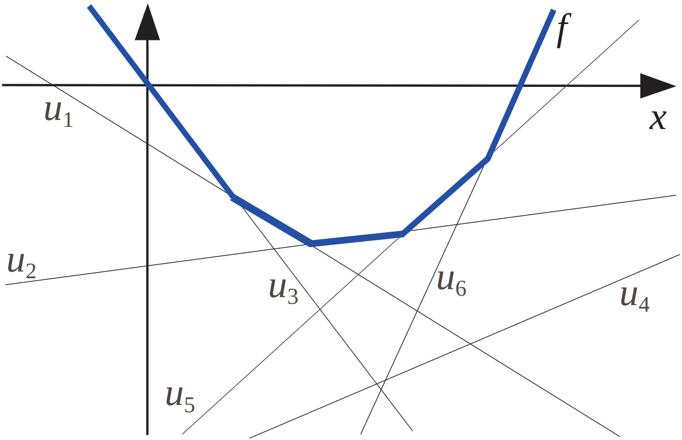
[4] Beltran, Tadonki, Vial "Solving the p-Median Problem with a Semi-Lagrangian Relaxation", COAP, 2006

Approximate Lagrangian Relaxation II

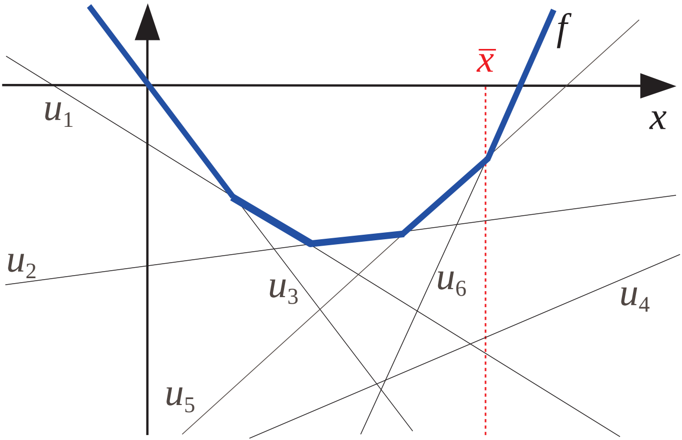
- Heuristics have no (or too weak in practice) performance guarantee
- Different approach: an exact algorithm for solving (2)
- Three main components:
  - a heuristic producing $\bar{u} \in U \Rightarrow c(\bar{u}) + xh(\bar{u}) \leq f(x)$
  - an upper bound $\bar{f}(x) \geq f(x)$ (further relaxation)
  - enumeration to squeeze the two together (branching)
- Iterative process where $c(\bar{u}) + xh(\bar{u}) \rightarrow f(x) \leftarrow \bar{f}(x)$
- (2) "as difficult" as (1) in theory (but largely less so in practice[4])
- The gap $\sigma = \bar{f}(x) - c(\bar{u}) - xh(\bar{u}) \geq 0$ may decrease rather slowly
- For bounding purposes, $\bar{f}(x)$ "is" $f(x)$

[4] Beltran, Tadonki, Vial "Solving the p-Median Problem with a Semi-Lagrangian Relaxation", COAP, 2006

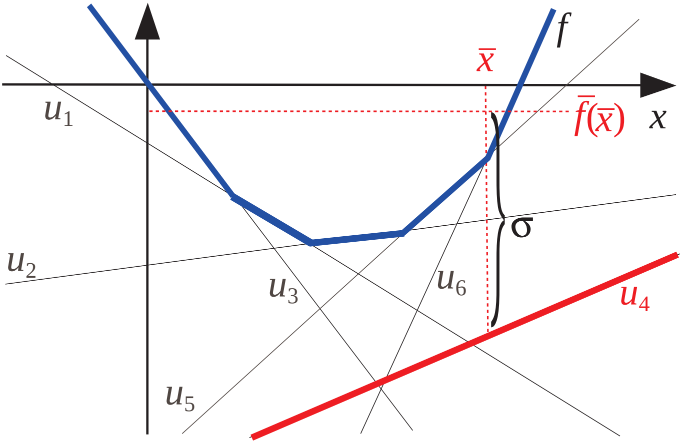
Approximate Lagrangian Relaxation II (graphically)



Approximate Lagrangian Relaxation II (graphically)

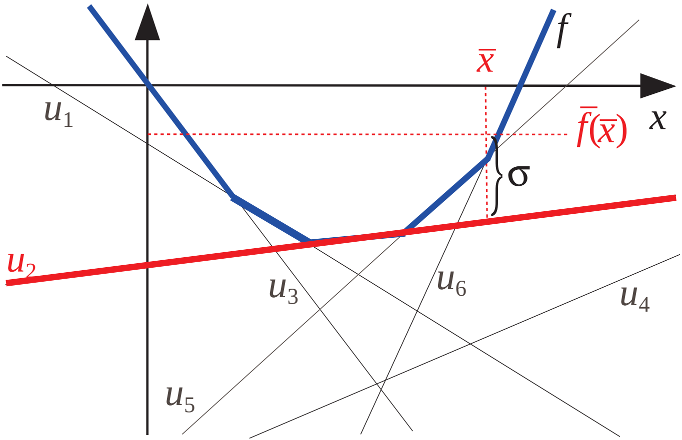


Approximate Lagrangian Relaxation II (graphically)



- The upper bound $\bar{f}(x)$ "is" the function value

Approximate Lagrangian Relaxation II (graphically)



- The upper bound $\bar{f}(x)$ "is" the function value
- σ decreases if either $\bar{f}(x)$ decreases or $c(\bar{u}) + xh(\bar{u})$ increases

A Somewhat Different (but related) Case

- The decomposable case:

$$u = (u^1, \ldots, u^k) \in U^1 \times \ldots \times U^k$$
$$c(u) = c^1(u^1) + \ldots + c^k(u^k)$$
$$h(u) = h^1(u^1) + \ldots + h^k(u^k)$$

- Computing $f(x)$ decomposes into k independent subproblems

[5] Nedíc, Bertsekas "Incremental subgradient methods for nondifferentiable optimization", SIOPT, 2001

A Somewhat Different (but related) Case

- The decomposable case:

$$u = (u^1, \dots, u^k) \in U^1 \times \dots \times U^k$$
$$c(u) = c^1(u^1) + \dots + c^k(u^k)$$
$$h(u) = h^1(u^1) + \dots + h^k(u^k)$$

- Computing $f(x)$ decomposes into k independent subproblems
- In some cases, the problems are "easy" but they are "many"
- Avoid computing them all for each x, at least at some iterations [5]

[5] Nedíc, Bertsekas "Incremental subgradient methods for nondifferentiable optimization", SIOPT, 2001

A Somewhat Different (but related) Case

- The decomposable case:

$$
\begin{aligned}
u &= (u^1, \ldots, u^k) \in U^1 \times \ldots \times U^k \\
c(u) &= c^1(u^1) + \ldots + c^k(u^k) \\
h(u) &= h^1(u^1) + \ldots + h^k(u^k)
\end{aligned}
$$

- Computing $f(x)$ decomposes into k independent subproblems
- In some cases, the problems are "easy" but they are "many"
- Avoid computing them all for each x, at least at some iterations [5]
- Something like: lower bound always available, upper bound only available if all k problems are solved

[5] Nedíc, Bertsekas "Incremental subgradient methods for nondifferentiable optimization", SIOPT, 2001

A Somewhat Different (but related) Case

- The decomposable case:

$$u = (u^1, \dots, u^k) \in U^1 \times \dots \times U^k$$
$$c(u) = c^1(u^1) + \dots + c^k(u^k)$$
$$h(u) = h^1(u^1) + \dots + h^k(u^k)$$

- Computing $f(x)$ decomposes into k independent subproblems
- In some cases, the problems are “easy” but they are “many”
- Avoid computing them all for each x, at least at some iterations [5]
- Something like: lower bound always available, upper bound only available if all k problems are solved
- Alternatively: $\bar{f}(x)$ is either $+\infty$ or $f(x)$

[5] Nedíc, Bertsekas “Incremental subgradient methods for nondifferentiable optimization”, SIOPT, 2001

A Somewhat Different (but related) Case

- The decomposable case:

$$u = (u^1, \ldots, u^k) \in U^1 \times \ldots \times U^k$$
$$c(u) = c^1(u^1) + \ldots + c^k(u^k)$$
$$h(u) = h^1(u^1) + \ldots + h^k(u^k)$$

- Computing $f(x)$ decomposes into k independent subproblems
- In some cases, the problems are "easy" but they are "many"
- Avoid computing them all for each x, at least at some iterations [5]
- Something like: lower bound always available, upper bound only available if all k problems are solved
- Alternatively: $\bar{f}(x)$ is either $+\infty$ or $f(x)$
- Then, of course, each subproblem can be solved approximately

[5] Nedíc, Bertsekas "Incremental subgradient methods for nondifferentiable optimization", SIOPT, 2001

The Issue

- Minimizing f using a approximated subgradient (= oracle) possible [6]

[6] Correa, Lemaréchal "Convergence of Some Algorithms for Convex Minimization" Math. Prog., 1993

[7] Kiwiel "Convergence of approximate and incremental subgradient methods for convex minimization", SIOPT, 2004

[8] Kiwiel "A proximal bundle method with approximate subgradient linearizations", SIOPT, 2006

[9] Kiwiel, Lemaréchal "An inexact bundle variant suited to column generation", Math. Prog., 2007

The Issue

- Minimizing f using a approximated subgradient (= oracle) possible [6]
- Lately, the standard has been "nothing is known about σ" [7] [8] [9]

[6] Correa, Lemaréchal "Convergence of Some Algorithms for Convex Minimization" Math. Prog., 1993
[7] Kiwiel "Convergence of approximate and incremental subgradient methods for convex minimization", SIOPT, 2004
[8] Kiwiel "A proximal bundle method with approximate subgradient linearizations", SIOPT, 2006
[9] Kiwiel, Lemaréchal "An inexact bundle variant suited to column generation", Math. Prog., 2007

The Issue

- Minimizing f using a approximated subgradient (= oracle) possible [6]
- Lately, the standard has been "nothing is known about σ" [7] [8] [9]
- But in practice, σ is known (if we accept that $\bar{f}(x)$ "is" $f(x)$)

[6] Correa, Lemaréchal "Convergence of Some Algorithms for Convex Minimization" Math. Prog., 1993
[7] Kiwiel "Convergence of approximate and incremental subgradient methods for convex minimization", SIOPT, 2004
[8] Kiwiel "A proximal bundle method with approximate subgradient linearizations", SIOPT, 2006
[9] Kiwiel, Lemaréchal "An inexact bundle variant suited to column generation", Math. Prog., 2007

The Issue

- Minimizing f using a approximated subgradient (= oracle) possible [6]
- Lately, the standard has been "nothing is known about σ" [7] [8] [9]
- But in practice, σ is known (if we accept that $\bar{f}(x)$ "is" $f(x)$)
- The issue:

Does knowing σ help in (approximately) minimizing f?

[6] Correa, Lemaréchal "Convergence of Some Algorithms for Convex Minimization" Math. Prog., 1993

[7] Kiwiel "Convergence of approximate and incremental subgradient methods for convex minimization", SIOPT, 2004

[8] Kiwiel "A proximal bundle method with approximate subgradient linearizations", SIOPT, 2006

[9] Kiwiel, Lemaréchal "An inexact bundle variant suited to column generation", Math. Prog., 2007

The Issue

- Minimizing f using a approximated subgradient (= oracle) possible [6]
- Lately, the standard has been "nothing is known about σ" [7] [8] [9]
- But in practice, σ is known (if we accept that $\bar{f}(x)$ "is" $f(x)$)
- The issue:

  Does knowing σ help in (approximately) minimizing f?

- Of course, it depends on what approach is used

---

[6] Correa, Lemaréchal "Convergence of Some Algorithms for Convex Minimization" Math. Prog., 1993

[7] Kiwiel "Convergence of approximate and incremental subgradient methods for convex minimization", SIOPT, 2004

[8] Kiwiel "A proximal bundle method with approximate subgradient linearizations", SIOPT, 2006

[9] Kiwiel, Lemaréchal "An inexact bundle variant suited to column generation", Math. Prog., 2007

Subgradient Methods

(with Giacomo d'Antonio)

(approximate) Subgradient Methods

- General problem:

$$\inf_x \{ f(x) \; : \; x \in X \}$$

$f : \mathbb{R}^n \to \mathbb{R}$ convex = approximated oracle, $X \subseteq \mathbb{R}^n$ closed convex

(approximate) Subgradient Methods

- General problem:

$$\inf_x \{ f(x) \ : \ x \in X \}$$

$f : \mathbb{R}^n \to \mathbb{R}$ convex = approximated oracle, $X \subseteq \mathbb{R}^n$ closed convex

- Basic approximate subgradient method:

$$g_k \in \partial_{\sigma_k} f(x_k) \quad , \quad \widehat{x}_{k+1} = x_k - \nu_k g_k \quad , \quad x_{k+1} = P_X(\widehat{x}_{k+1})$$

P_X = orthogonal projection on X (assumed "cheap"), ν_k stepsize

(approximate) Subgradient Methods

- General problem:

$$\inf_x \{ \, f(x) \; : \; x \in X \, \}$$

$f : \mathbb{R}^n \to \mathbb{R}$ convex = approximated oracle, $X \subseteq \mathbb{R}^n$ closed convex

- Basic approximate subgradient method:

$$g_k \in \partial_{\sigma_k} f(x_k) \quad , \quad \widehat{x}_{k+1} = x_k - \nu_k g_k \quad , \quad x_{k+1} = P_X(\widehat{x}_{k+1})$$

P_X = orthogonal projection on X (assumed "cheap"), ν_k stepsize

- Very simple, almost no overhead w.r.t. $f(x)$ computation
- Many variants (dilation methods, Bregman projections, ...)

(approximate) Subgradient Methods

- General problem:

$$\inf_x \{ f(x) \ : \ x \in X \}$$

$f : \mathbb{R}^n \to \mathbb{R}$ convex = approximated oracle, $X \subseteq \mathbb{R}^n$ closed convex

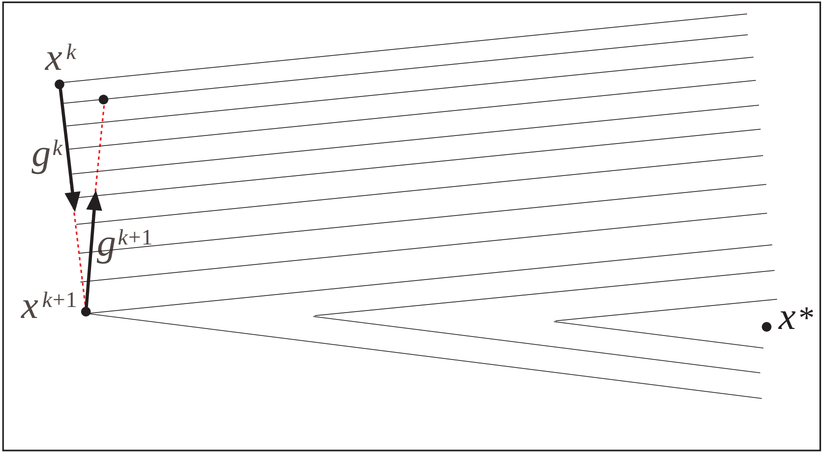
- Basic approximate subgradient method:

$$g_k \in \partial_{\sigma_k} f(x_k) \quad , \quad \widehat{x}_{k+1} = x_k - \nu_k g_k \quad , \quad x_{k+1} = P_X(\widehat{x}_{k+1})$$

P_X = orthogonal projection on X (assumed "cheap"), ν_k stepsize

- Very simple, almost no overhead w.r.t. $f(x)$ computation
- Many variants (dilation methods, Bregman projections, ...)
- Typically rather slow, because:
  - a $(1 - \varepsilon)$th-order method, cannot be fast
  - zig-zagging I: in "deep and narrow valleys", successive subgradients almost orthogonal to each other
  - zig-zagging II: at ∂X, subgradients almost orthogonal to ∂X

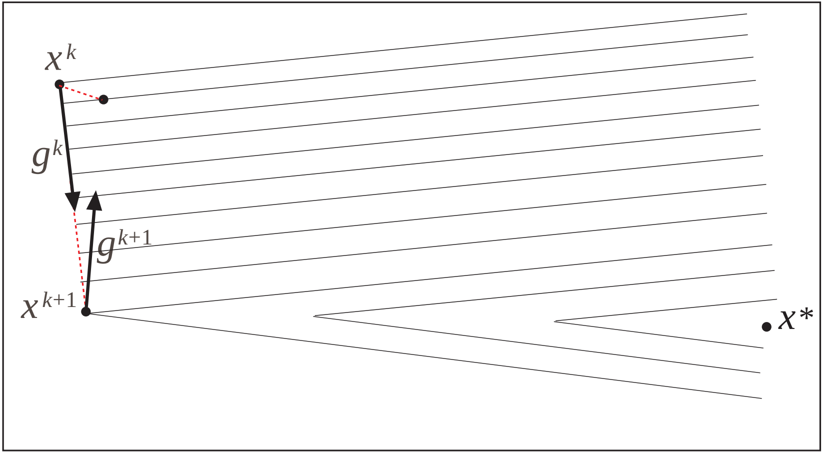
Zig-Zagging I



- Two long steps . . .

[10] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques", Math. Prog., 1975

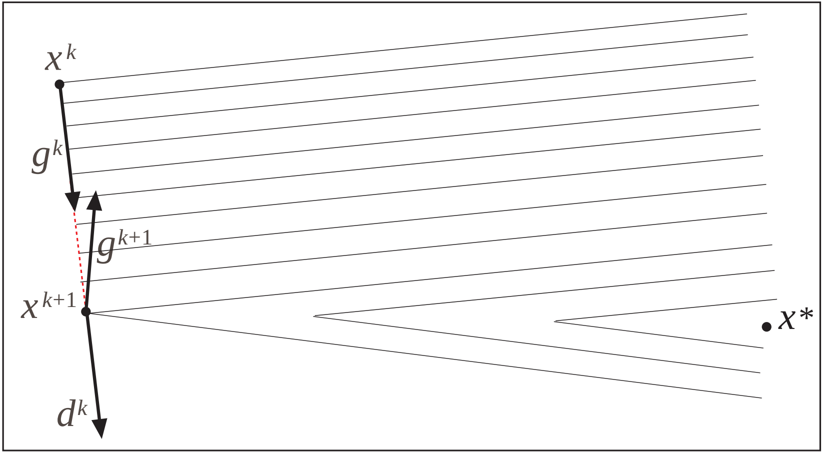
Zig-Zagging I



- Two long steps . . . are one short step

[10] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques", Math. Prog., 1975

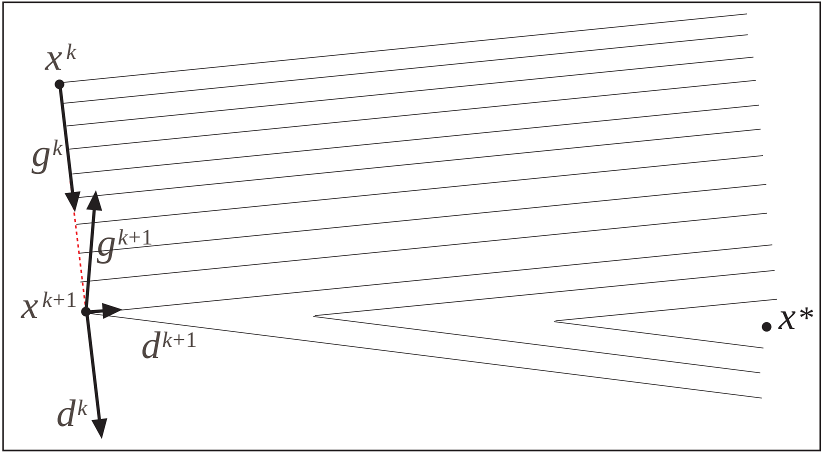
Zig-Zagging I



- Two long steps . . . are one short step
- Solution: use previous direction

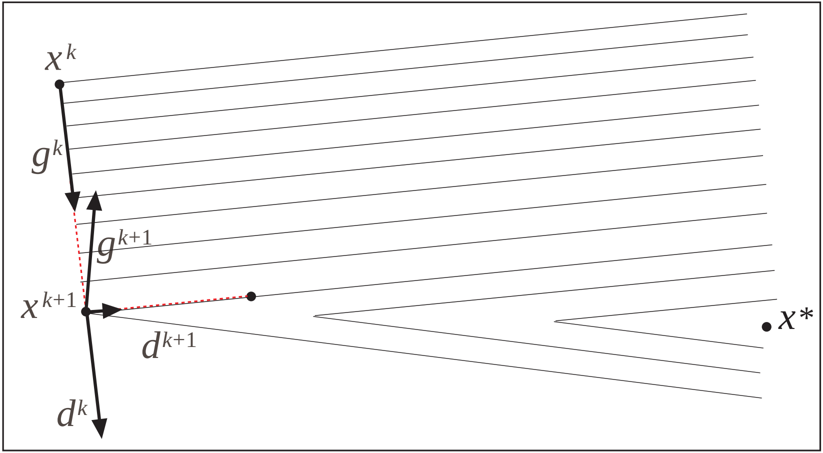
[10] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques", Math. Prog., 1975

Zig-Zagging I



- Two long steps . . . are one short step
- Solution: use previous direction to deflect g_k (e.g. $\rightarrow d_k d_{k-1} \geq 0$)[10]

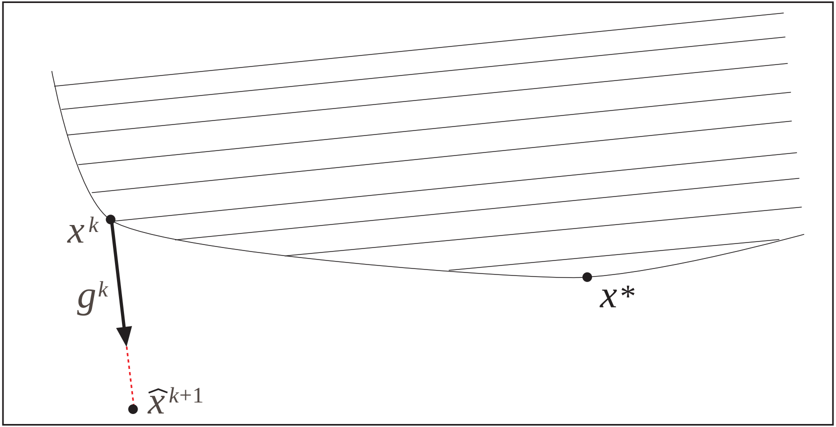
[10] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques", Math. Prog., 1975



- Two long steps . . . are one short step
- Solution: use previous direction to deflect g_k (e.g. $\rightarrow d_k d_{k-1} \geq 0$)[10]

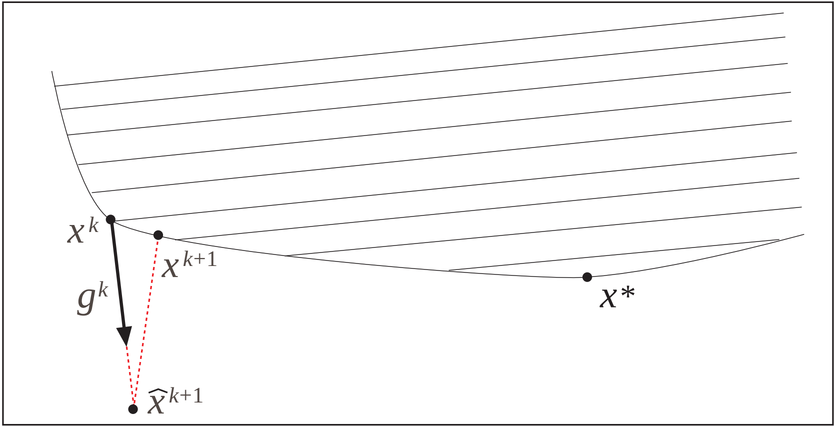
[10] Camerini, Fratta, Maffioli "On Improving Relaxation Methods by Modified Gradient Techniques", Math. Prog., 1975

Zig-Zagging II



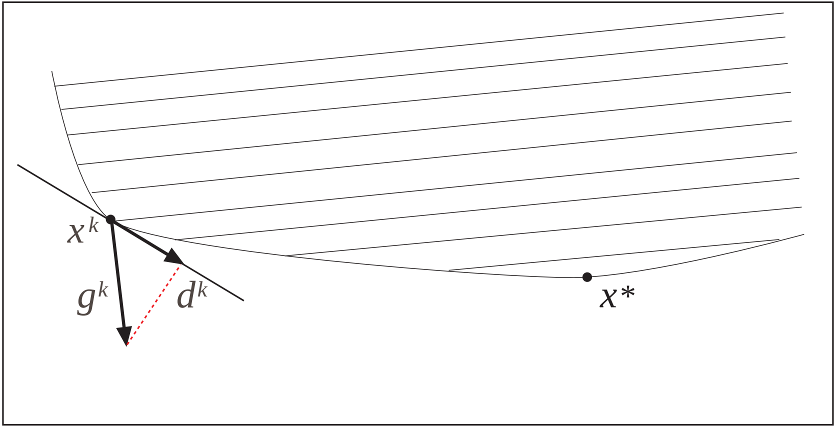
- Projecting a long step . . .

Zig-Zagging II



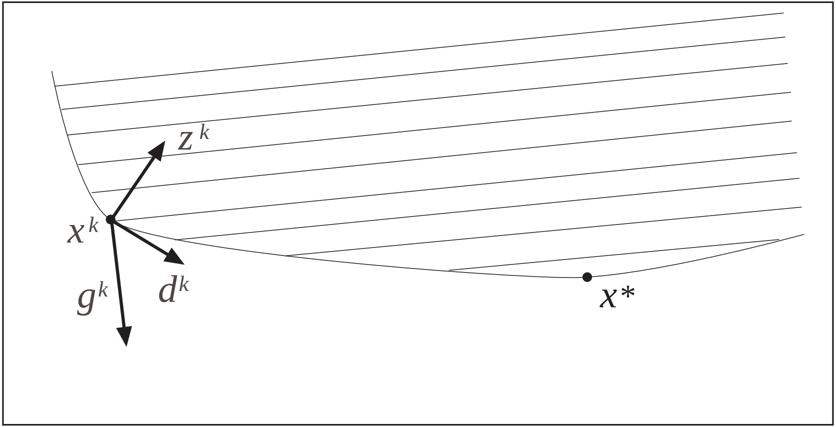
- Projecting a long step . . . may result in a short step

Zig-Zagging II



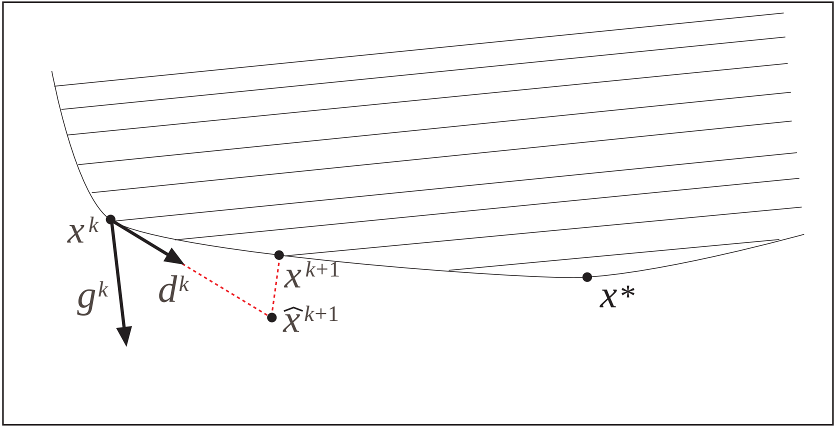
- Projecting a long step . . . may result in a short step
- Solution: project g^k onto the tangent cone at x^k

Zig-Zagging II



- Projecting a long step . . . may result in a short step
- Solution: project g^k onto the tangent cone at x^k . . . or, equivalently, deflect using $-z^k \in \partial I_X(x^k) \rightarrow d_k \in \partial f_X(x^k)$ $(f_X = f + I_X)$

Zig-Zagging II



- Projecting a long step . . . may result in a short step
- Solution: project g^k onto the tangent cone at x^k . . . or, equivalently, deflect using $-z^k \in \partial I_X(x^k) \rightarrow d_k \in \partial f_X(x^k)$ $(f_X = f + I_X)$

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)^{11} \in \partial f_X(x^k)$

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods . . . with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity . . . Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)^{11} \in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)$[11] $\in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$ ... better, w.l.o.g.

$$d_k = \alpha_k g_k + (1 - \alpha_k) d_{k-1} \quad , \quad \alpha_k \in [0, 1]$$

(the missing scaling factor can always be attached to ν_k) [12]

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)^{11} \in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$ ... better, w.l.o.g.

$$d_k = \alpha_k g_k + (1 - \alpha_k) d_{k-1} \quad , \quad \alpha_k \in [0, 1]$$

(the missing scaling factor can always be attached to ν_k) [12]

- Funnily enough, (almost) no conditional deflected subgradient [13]

---

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)^{11} \in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$ ... better, w.l.o.g.

$$d_k = \alpha_k g_k + (1 - \alpha_k) d_{k-1} \quad , \quad \alpha_k \in [0, 1]$$

(the missing scaling factor can always be attached to ν_k) [12]

- Funnily enough, (almost) no conditional deflected subgradient [13]
- Besides: conditional approximate subgradient, yes[7]

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)^{11} \in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$ ... better, w.l.o.g.

$$d_k = \alpha_k g_k + (1 - \alpha_k) d_{k-1} \quad , \quad \alpha_k \in [0, 1]$$

(the missing scaling factor can always be attached to ν_k) [12]

- Funnily enough, (almost) no conditional deflected subgradient [13]
- Besides: conditional approximate subgradient, yes[7]
  ... but deflected approximate subgradient, no.

[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Two Classes of Subgradient Methods

- Conditional subgradient: $d_k = -P_{T_X(x_k)}(-g_k)$[11] $\in \partial f_X(x^k)$
- Deflected subgradient: $d_k = g_k + \eta_k d_{k-1}$ ... better, w.l.o.g.

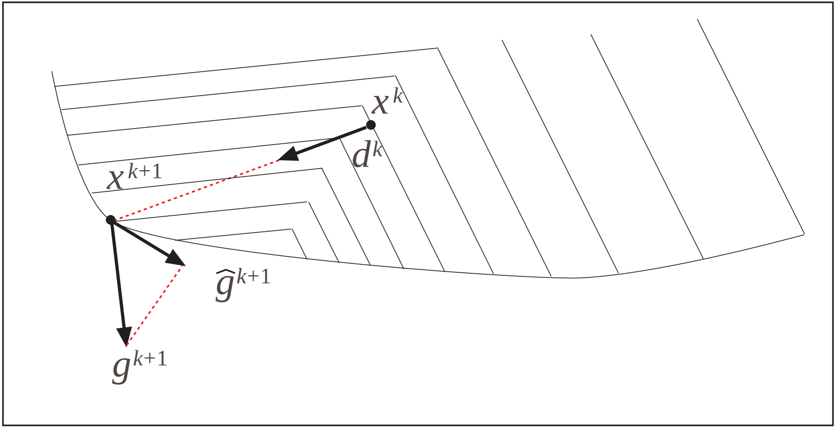
$$d_k = \alpha_k g_k + (1 - \alpha_k) d_{k-1} \quad , \quad \alpha_k \in [0, 1]$$

(the missing scaling factor can always be attached to ν_k) [12]

- Funnily enough, (almost) no conditional deflected subgradient [13]
- Besides: conditional approximate subgradient, yes[7]
  ... but deflected approximate subgradient, no.
- Still there is need for good subgradient methods [14] [15]

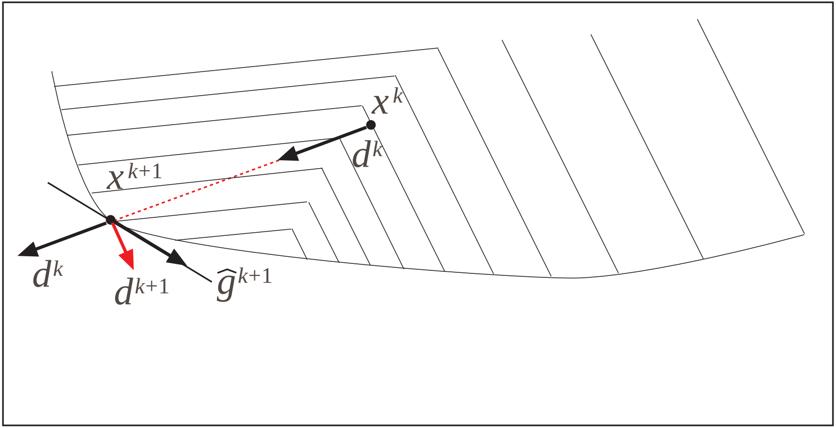
[11] Larsson, Patriksson, Strömberg "Conditional Subgradient Optimization - Theory and Applications", EJOR, 1996
[12] Sherali, Lim "On Embedding the Volume Algorithm in a Variable Target Value Method", ORL, 2004
[13] Guta "Subgradient Optimization Methods ... with an Application to a Radiation Therapy Problem", Ph.D., 2003
[14] Crainic, F., Gendron "Bundle-based Relaxation Methods for Multicommodity ... Network Design", DAM, 2001
[15] F., Lodi, Rinaldi "New Approaches for Optimizing over the Semimetric Polytope", Math. Prog., 2005

Why Conditional + Deflected is Not (entirely) Obvious



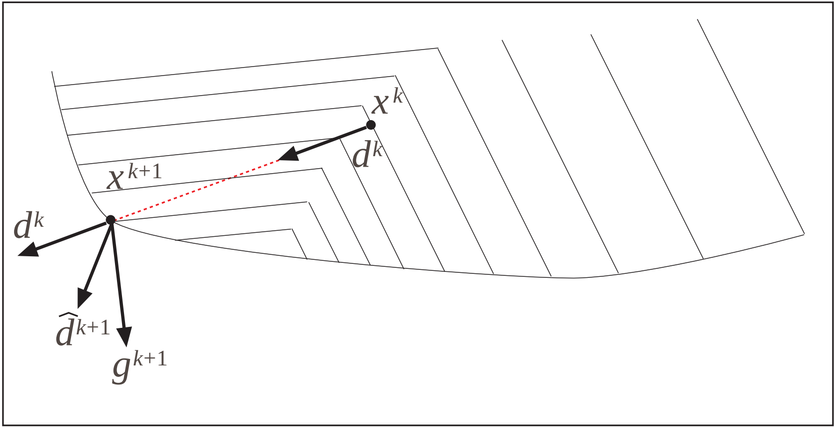
- Projecting ...

Why Conditional + Deflected is Not (entirely) Obvious



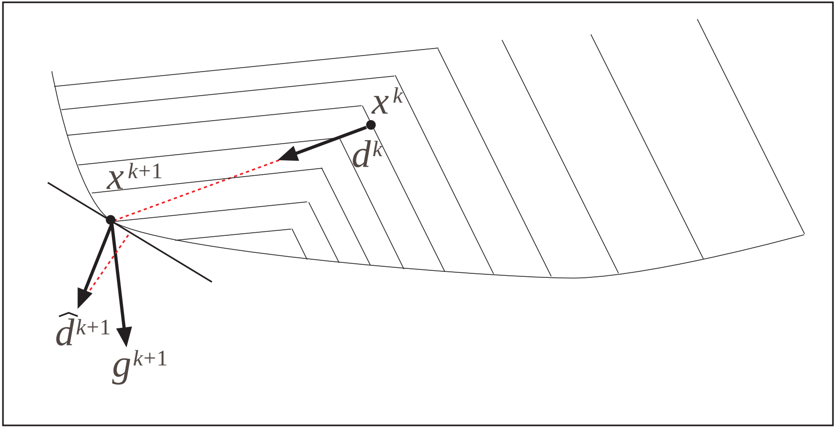
- Projecting . . . and then deflecting gives $d_{k+1} \notin T_X(x_k)$

Why Conditional + Deflected is Not (entirely) Obvious



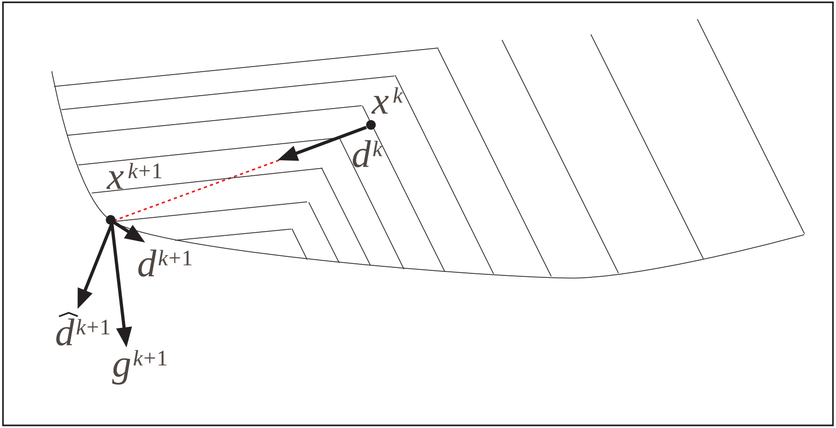
- Projecting . . . and then deflecting gives $d_{k+1} \notin T_X(x_k)$
- Solution: first deflect,

Why Conditional + Deflected is Not (entirely) Obvious



- Projecting . . . and then deflecting gives $d_{k+1} \notin T_X(x_k)$
- Solution: first deflect, then project; now $d_{k+1} \in T_X(x_k)$

Why Conditional + Deflected is Not (entirely) Obvious



- Projecting . . . and then deflecting gives $d_{k+1} \notin T_X(x_k)$
- Solution: first deflect, then project; now $d_{k+1} \in T_X(x_k)$

Conditional Deflected (Approximate) Subgradient

$$\widehat{d}_k = \alpha_k \bar{g}_k + (1 - \alpha_k)\bar{d}_{k-1} \quad d_k = -P_{T_X(x_k)}(-\widehat{d}_k)$$

$$\bar{g}_k = \text{either } g_k \text{ or } \widehat{g}_k \ , \quad \bar{d}_k = \text{either } d_k \text{ or } \widehat{d}_k$$

- Four different schemes, but unified treatment ($\leq$ two projections)

Conditional Deflected (Approximate) Subgradient

$$\widehat{d}_k = \alpha_k \bar{g}_k + (1 - \alpha_k)\bar{d}_{k-1} \quad d_k = -P_{T_X(x_k)}(-\widehat{d}_k)$$

$$\bar{g}_k = \text{either } g_k \text{ or } \widehat{g}_k \ , \quad \bar{d}_k = \text{either } d_k \text{ or } \widehat{d}_k$$

- Four different schemes, but unified treatment ($\leq$ two projections)
- Whatever the choice, $\bar{g}_k \in \partial_{\sigma_k} f_X(x_k)$
- Allows to unify some technical results, like

$$\bar{d}_k(x - x_k) \leq \widehat{d}_k(x - x_k)$$

(trivial if $\bar{d}_k = \widehat{d}_k$, but not otherwise), and

$$\bar{d}_k(x_k - x_{k+1}) \leq \nu_k \|d_k\|^2$$

Conditional Deflected (Approximate) Subgradient

$$\widehat{d}_k = \alpha_k \bar{g}_k + (1 - \alpha_k)\bar{d}_{k-1} \quad d_k = -P_{T_X(x_k)}(-\widehat{d}_k)$$

$$\bar{g}_k = \text{either } g_k \text{ or } \widehat{g}_k \ , \quad \bar{d}_k = \text{either } d_k \text{ or } \widehat{d}_k$$

- Four different schemes, but unified treatment ($\leq$ two projections)
- Whatever the choice, $\bar{g}_k \in \partial_{\sigma_k} f_X(x_k)$
- Allows to unify some technical results, like

$$\bar{d}_k(x - x_k) \leq \widehat{d}_k(x - x_k)$$

(trivial if $\bar{d}_k = \widehat{d}_k$, but not otherwise), and

$$\bar{d}_k(x_k - x_{k+1}) \leq \nu_k \|d_k\|^2$$

- Crucial result (relying on $\alpha_k \in [0, 1]$): $\bar{d}_k \in \partial_{\varepsilon_k} f_X(x_k)$ with

$$\varepsilon_k = (1 - \alpha_k)\big(\ f_k - f_{k-1} - \bar{d}_{k-1}(x_k - x_{k-1}) + \varepsilon_{k-1}\ \big) + \alpha_k \sigma_k \quad (4)$$

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \le \beta_k \le 2$$

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq 2$$

- Abstract rule, as f^* unknown in general

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \le \beta_k \le 2$$

- Abstract rule, as f^* unknown in general
- Technical (but somewhat conceptual) issue: d_k can be 0

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq 2$$

- Abstract rule, as f^* unknown in general
- Technical (but somewhat conceptual) issue: d_k can be 0
- Not an issue if σ_k constant (e.g. $\sigma_k \equiv 0$) and no deflection

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq 2$$

- Abstract rule, as f^* unknown in general
- Technical (but somewhat conceptual) issue: d_k can be 0
- Not an issue if σ_k constant (e.g. $\sigma_k \equiv 0$) and no deflection
- "Technical" solution $\nu_k \|d_k\|^2 \leq \beta_k \lambda_k$ ($\lambda_k = f_k - f^*$), not enough

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \le \beta_k \le 2$$

- Abstract rule, as f^* unknown in general
- Technical (but somewhat conceptual) issue: d_k can be 0
- Not an issue if σ_k constant (e.g. $\sigma_k \equiv 0$) and no deflection
- "Technical" solution $\nu_k \|d_k\|^2 \le \beta_k \lambda_k$ ($\lambda_k = f_k - f^*$), not enough

Observation

$\sigma^* = \limsup_{k \to \infty} \sigma_k < +\infty$ *(asymptotic maximum error of the oracle); no subgradient method can attain error* $< \sigma^*$ *(if* $f^* > -\infty$*)*

(standard) Polyak Stepsize

- Standard Polyak stepsize (assuming $f^* = \inf_x f_X(x) > -\infty$)

$$\nu_k = \beta_k \frac{f_k - f^*}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq 2$$

- Abstract rule, as f^* unknown in general
- Technical (but somewhat conceptual) issue: d_k can be 0
- Not an issue if σ_k constant (e.g. $\sigma_k \equiv 0$) and no deflection
- "Technical" solution $\nu_k \|d_k\|^2 \leq \beta_k \lambda_k$ ($\lambda_k = f_k - f^*$), not enough

Observation

$\sigma^* = \limsup_{k \to \infty} \sigma_k < +\infty$ (asymptotic maximum error of the oracle); no subgradient method can attain error $< \sigma^*$ (if $f^* > -\infty$)

Proof.

$\sigma_k \geq \sigma^*$ and $f(x_0) = f^* + \sigma^* \Rightarrow g_k$ can be 0 $\Rightarrow d_k = 0$: never moves! □

Polyak Stepsize (cont.d)

- Further requirement: $\beta_k \leq \alpha_k \; (\leq 1)$

Polyak Stepsize (cont.d)

- Further requirement: $\beta_k \leq \alpha_k \; (\leq 1)$
- Main technical result (using (4)):

$$\varepsilon_k \leq (1 - \alpha_k)(f_k - f^*) + \bar{\sigma}_k \qquad \text{where} \tag{5}$$

$$\bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k \tag{6}$$

($\alpha_1 = 1$ for "unreliability of past information")

Polyak Stepsize (cont.d)

- Further requirement: $\beta_k \leq \alpha_k \; (\leq 1)$
- Main technical result (using (4)):

$$\varepsilon_k \leq (1 - \alpha_k)(f_k - f^*) + \bar{\sigma}_k \qquad \text{where} \tag{5}$$

$$\bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k \tag{6}$$

($\alpha_1 = 1$ for "unreliability of past information")

- Technical corollary: for each $\bar{x} \in X$

$$d_k(\bar{x} - x_k) \leq \alpha_k(f^* - f_k) + \left[f(\bar{x}) - f^* + \bar{\sigma}_k \right] \tag{7}$$

Polyak Stepsize (cont.d)

- Further requirement: $\beta_k \leq \alpha_k \ (\leq 1)$

- Main technical result (using (4)):

$$\varepsilon_k \leq (1 - \alpha_k)(f_k - f^*) + \bar{\sigma}_k \qquad \text{where} \tag{5}$$

$$\bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k \tag{6}$$

($\alpha_1 = 1$ for "unreliability of past information")

- Technical corollary: for each $\bar{x} \in X$

$$d_k(\bar{x} - x_k) \leq \alpha_k(f^* - f_k) + \left[f(\bar{x}) - f^* + \bar{\sigma}_k \right] \tag{7}$$

- "Exact" convergence result at hand[7]: $\sigma_k \equiv 0 \Rightarrow$

$$\exists \xi \in [0, 1) \qquad \varepsilon_k \leq \xi(2 - \beta_k)(f_k - f^*)/2$$

$\Rightarrow \liminf_{k \to \infty} f_k = f^\infty \leq f^*$

Polyak Stepsize: the Approximate Case

- What about the approximate case?

Polyak Stepsize: the Approximate Case

- What about the approximate case?
- "Asymptotic error": $\limsup_{k \to \infty} \bar{\sigma}_k = \bar{\sigma}^* \leq \sigma^*$

Polyak Stepsize: the Approximate Case

- What about the approximate case?
- "Asymptotic error": $\limsup_{k\to\infty} \bar{\sigma}_k = \bar{\sigma}^* \leq \sigma^*$
- For "Asymptotically non-deflected" method ($\lim_{k\to\infty} \alpha_k = 1$)[7]

$$f^\infty \leq f^* + 2\sigma^* / (2 - \sup_k \beta_k)$$

- Error twice as large than "optimal", basically no deflection

Polyak Stepsize: the Approximate Case

- What about the approximate case?
- "Asymptotic error": $\limsup_{k \to \infty} \bar{\sigma}_k = \bar{\sigma}^* \leq \sigma^*$
- For "Asymptotically non-deflected" method ($\lim_{k \to \infty} \alpha_k = 1$)[7]

$$f^\infty \leq f^* + 2\sigma^* / (2 - \sup_k \beta_k)$$

- Error twice as large than "optimal", basically no deflection

Theorem

Without any assumption on deflection

$$f^\infty \leq f^* + 2\sigma^* / \Gamma \quad \textit{where} \quad \Gamma = \inf_k \; 2\alpha_k - \beta_k \geq \beta^*$$

- Deflecting is possible, but does not look a good idea

Polyak Stepsize: the Approximate Case

- What about the approximate case?
- "Asymptotic error": $\limsup_{k\to\infty} \; \bar{\sigma}_k = \bar{\sigma}^* \leq \sigma^*$
- For "Asymptotically non-deflected" method ($\lim_{k\to\infty} \alpha_k = 1$)[7]

$$f^\infty \leq f^* + 2\sigma^* / (2 - \sup_k \; \beta_k)$$

- Error twice as large than "optimal", basically no deflection

Theorem

Without any assumption on deflection

$$f^\infty \leq f^* + 2\sigma^* / \Gamma \quad \textit{where} \quad \Gamma = \inf_k \; 2\alpha_k - \beta_k \geq \beta^*$$

- Deflecting is possible, but does not look a good idea
- However, knowing σ_k we can do better than that

Corrected Polyak Stepsize

- Corrected Polyak stepsize: $\lambda_k = f_k - f^* - \sigma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{8}$$

Corrected Polyak Stepsize

- Corrected Polyak stepsize: $\lambda_k = f_k - f^* - \sigma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{8}$$

- Issue: $\sigma_k > f_k - f^* \Rightarrow \lambda_k < 0$. Solution:

$$0 \leq \nu_k \|d_k\|^2 \leq \beta_k \lambda_k \quad , \quad 0 \leq \beta_k \leq \alpha_k \leq 1$$

which implies $\lambda_k < 0 \ \Rightarrow \beta_k = 0 \Rightarrow \nu_k = 0$ (loops!)

Corrected Polyak Stepsize

- Corrected Polyak stepsize: $\lambda_k = f_k - f^* - \sigma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{8}$$

- Issue: $\sigma_k > f_k - f^* \Rightarrow \lambda_k < 0$. Solution:

$$0 \leq \nu_k \|d_k\|^2 \leq \beta_k \lambda_k \quad , \quad 0 \leq \beta_k \leq \alpha_k \leq 1$$

  which implies $\lambda_k < 0 \;\Rightarrow \beta_k = 0 \Rightarrow \nu_k = 0$ (loops!)

- In plain words: if the error is too large, stop until it decreases enough

Corrected Polyak Stepsize

- Corrected Polyak stepsize: $\lambda_k = f_k - f^* - \sigma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{8}$$

- Issue: $\sigma_k > f_k - f^* \Rightarrow \lambda_k < 0$. Solution:

$$0 \leq \nu_k \|d_k\|^2 \leq \beta_k \lambda_k \quad , \quad 0 \leq \beta_k \leq \alpha_k \leq 1$$

which implies $\lambda_k < 0 \ \Rightarrow \beta_k = 0 \Rightarrow \nu_k = 0$ (loops!)

- In plain words: if the error is too large, stop until it decreases enough

- Actually, a slightly stronger form is required:

$$\begin{array}{l} \lambda_k \geq 0 \ \Rightarrow \ (\alpha_k \ \geq) \ \beta_k \ \geq \ \beta^* > 0 \ , \\ \lambda_k < 0 \ \Rightarrow \ \alpha_k = 0 \ (\Rightarrow \beta_k = 0) \end{array}$$

Corrected Polyak Stepsize

- Corrected Polyak stepsize: $\lambda_k = f_k - f^* - \sigma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{8}$$

- Issue: $\sigma_k > f_k - f^* \Rightarrow \lambda_k < 0$. Solution:

$$0 \leq \nu_k \|d_k\|^2 \leq \beta_k \lambda_k \quad , \quad 0 \leq \beta_k \leq \alpha_k \leq 1$$

which implies $\lambda_k < 0 \Rightarrow \beta_k = 0 \Rightarrow \nu_k = 0$ (loops!)

- In plain words: if the error is too large, stop until it decreases enough

- Actually, a slightly stronger form is required:

$$\begin{aligned} \lambda_k \geq 0 \;&\Rightarrow\; (\alpha_k \;\geq)\; \beta_k \;\geq\; \beta^* > 0 \;, \\ \lambda_k < 0 \;&\Rightarrow\; \alpha_k = 0 \; (\Rightarrow \beta_k = 0) \end{aligned}$$

- (8) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = \alpha_k \sigma_k$;
good deflecting "shaves away" a part of the error

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$
- Better result than the available ones[7]:
  - Optimal error attained even in inexact case
  - Convergence of the iterates (in the exact case)
  - Deflection does not worsen results

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$
- Better result than the available ones[7]:
  - Optimal error attained even in inexact case
  - Convergence of the iterates (in the exact case)
  - Deflection does not worsen results
- Interesting detail of the proof:
  some things only hold if $\lambda_k \geq 0$ for *infinitely many* k,

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$
- Better result than the available ones[7]:
  - Optimal error attained even in inexact case
  - Convergence of the iterates (in the exact case)
  - Deflection does not worsen results
- Interesting detail of the proof:
  some things only hold if $\lambda_k \geq 0$ for *infinitely many* k,
  which does not necessarily happen

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$
- Better result than the available ones[7]:
  - Optimal error attained even in inexact case
  - Convergence of the iterates (in the exact case)
  - Deflection does not worsen results
- Interesting detail of the proof:
  some things only hold if $\lambda_k \geq 0$ for *infinitely many* k,
  which does not necessarily happen
  but if not, a σ^*-optimal solution is finitely attained

Corrected Polyak Stepsize

- Without any assumption on deflection: (8) $\Rightarrow$
  - $f^\infty \leq f^* + \sigma^*$
  - $X^* \neq \emptyset \Rightarrow \exists$ subsequence $\{x_{k_i}\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
  - $X^* \neq \emptyset$ & $\sigma^* = 0 \Rightarrow$ the whole $\{x_k\} \to x^* \in X^*$
- Better result than the available ones[7]:
  - Optimal error attained even in inexact case
  - Convergence of the iterates (in the exact case)
  - Deflection does not worsen results
- Interesting detail of the proof:
  some things only hold if $\lambda_k \geq 0$ for *infinitely many* k,
  which does not necessarily happen
  but if not, a σ^*-optimal solution is finitely attained
- Potential issue: exact knowledge of σ_k required

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{9}$$

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{9}$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \qquad (9)$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$
- General convergence:

$$f^\infty \leq f^* + 2\Delta/\Gamma$$

$$\Delta = \sigma^* + \bar{\gamma}(\ (1 - \beta^*)/\beta^* + \sup_k\ \alpha_k/2\ )$$
$$\bar{\gamma}\ =\ -\ \min \{\ \gamma^* = \liminf_{k \to \infty}\ \gamma_k\ ,\ 0\ \}$$

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{9}$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$
- General convergence:

$$f^\infty \leq f^* + 2\Delta/\Gamma$$

$$\Delta = \sigma^* + \bar{\gamma}(\ (1 - \beta^*)/\beta^* + \sup_k\ \alpha_k/2\ )$$
$$\bar{\gamma}\ =\ -\ \min\{\ \gamma^* = \liminf_{k\to\infty}\ \gamma_k\ ,\ 0\ \}$$

- "aiming higher than f^*" ($\gamma_k > 0$) good,
  "aiming lower than f^*" ($\gamma_k < 0$) bad

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{9}$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$

- General convergence:

$$f^\infty \leq f^* + 2\Delta/\Gamma$$

$$\Delta = \sigma^* + \bar{\gamma}(\ (1 - \beta^*)/\beta^* + \sup_k \ \alpha_k/2\ )$$
$$\bar{\gamma} \ = \ - \ \min \{\ \gamma^* = \liminf_{k \to \infty} \ \gamma_k \ , \ 0 \ \}$$

- "aiming higher than f^*" ($\gamma_k > 0$) good,
  "aiming lower than f^*" ($\gamma_k < 0$) bad

- On the other hand: aiming too high $\Rightarrow \lambda_k < 0 \Rightarrow$ loop

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \tag{9}$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$
- General convergence:

$$f^\infty \leq f^* + 2\Delta / \Gamma$$

$$\Delta = \sigma^* + \bar{\gamma}(\ (1 - \beta^*)/\beta^* + \sup_k \ \alpha_k/2\ )$$
$$\bar{\gamma} \ = \ - \ \min\{\ \gamma^* = \liminf_{k \to \infty} \ \gamma_k\ ,\ 0\ \}$$

- "aiming higher than f^*" ($\gamma_k > 0$) good,
  "aiming lower than f^*" ($\gamma_k < 0$) bad
- On the other hand: aiming too high $\Rightarrow \lambda_k < 0 \Rightarrow$ loop
- The highest safe value: σ_k (surprised?)

Generalized Corrected Polyak Stepsize

- The general form: $\lambda_k = f_k - f^* - \gamma_k$

$$0 \leq \nu_k = \beta_k \frac{\lambda_k}{\|d_k\|^2} \quad , \qquad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1 \qquad (9)$$

- (9) $\Rightarrow$ (5) + (7) with $\bar{\sigma}_k = (1 - \alpha_k)(\bar{\sigma}_{k-1} - \alpha_{k-1}\gamma_{k-1}) + \alpha_k \sigma_k$

- General convergence:

$$f^\infty \leq f^* + 2\Delta / \Gamma$$

$$\Delta = \sigma^* + \bar{\gamma}(\ (1 - \beta^*)/\beta^* + \sup_k\ \alpha_k/2\ )$$
$$\bar{\gamma}\ =\ -\ \min\left\{\ \gamma^* = \liminf_{k\to\infty}\ \gamma_k\ ,\ 0\ \right\}$$

- "aiming higher than f^*" ($\gamma_k > 0$) good,
  "aiming lower than f^*" ($\gamma_k < 0$) bad

- On the other hand: aiming too high $\Rightarrow \lambda_k < 0 \Rightarrow$ loop

- The highest safe value: σ_k (surprised?)

- What if I do not know σ_k exactly?

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \geq \xi \sigma^* \qquad \xi \in [0, 1] \tag{10}$$

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = \ 0 \ \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \ \geq \ \xi \sigma^* \qquad\qquad \xi \in [0, 1] \tag{10}$$

- (10) $\Rightarrow \bar{\sigma}_k \approx \sigma_k(\ 1 - (1 - \alpha_k)\xi \ )$ (technical form really ugly)

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \geq \xi \sigma^* \qquad \xi \in [0, 1] \tag{10}$$

- (10) $\Rightarrow \bar{\sigma}_k \approx \sigma_k(\, 1 - (1 - \alpha_k)\xi \,)$ (technical form really ugly)
- Convergence: (10) $\Rightarrow f^\infty \leq f^* + \sigma^*(\, \xi + 2(1 - \xi)/\Gamma)$

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \geq \xi \sigma^* \qquad \xi \in [0, 1] \tag{10}$$

- (10) $\Rightarrow \bar{\sigma}_k \approx \sigma_k (1 - (1 - \alpha_k)\xi)$ (technical form really ugly)
- Convergence: (10) $\Rightarrow f^\infty \leq f^* + \sigma^* (\xi + 2(1 - \xi)/\Gamma)$
- $\xi = 1 \Rightarrow$ "optimal" error

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \geq \xi \sigma^* \qquad \xi \in [0, 1] \tag{10}$$

- (10) $\Rightarrow \bar{\sigma}_k \approx \sigma_k(1 - (1 - \alpha_k)\xi)$ (technical form really ugly)
- Convergence: (10) $\Rightarrow f^\infty \leq f^* + \sigma^*(\xi + 2(1 - \xi)/\Gamma)$
- $\xi = 1 \Rightarrow$ "optimal" error
- Again, asymptotic results require $\lambda_k \geq 0$ for infinitely many k, if not a solution with prescribed accuracy finitely attained

Generalized (approximately) Corrected Polyak Stepsize

- Reminder: $\gamma_k = 0 \Rightarrow \bar{\sigma}_k = (1 - \alpha_k)\bar{\sigma}_{k-1} + \alpha_k \sigma_k$
  $\gamma_k = \sigma_k \Rightarrow \bar{\sigma}_k = \alpha_k \sigma_k$
- What if $\gamma_k > 0$ and "not too far" from σ_k?
- Abstract condition ($\Rightarrow \bar{\gamma} = 0$):

$$\liminf_{k \to \infty} \gamma_k = \gamma^* \geq \xi \sigma^* \qquad \xi \in [0, 1] \tag{10}$$

- (10) $\Rightarrow \bar{\sigma}_k \approx \sigma_k(1 - (1 - \alpha_k)\xi)$ (technical form really ugly)
- Convergence: (10) $\Rightarrow f^\infty \leq f^* + \sigma^*(\xi + 2(1 - \xi)/\Gamma)$
- $\xi = 1 \Rightarrow$ "optimal" error
- Again, asymptotic results require $\lambda_k \geq 0$ for infinitely many k, if not a solution with prescribed accuracy finitely attained
- Is (10) reasonable?

Target-level Approaches

- In general, f^* unknown (and it may be $-\infty$)

Target-level Approaches

- In general, f^* unknown (and it may be $-\infty$)
- Solution: replace it with a target f_{lev}^k, revise it appropriately

$$0 \leq \nu_k = \beta_k \frac{f_k - f_{lev}^k}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1$$

Target-level Approaches

- In general, f^* unknown (and it may be $-\infty$)
- Solution: replace it with a target f_{lev}^k, revise it appropriately

$$0 \leq \nu_k = \beta_k \frac{f_k - f_{lev}^k}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1$$

- Usually, $f_{lev}^k = f_{ref}^k$ (reference) $-\delta_k$ (threshold)
- Typical choice: $f_{ref}^k = f_{rec}^k \; = \; \min_{h \leq k} \; f(x_h)$ (record value)

Target-level Approaches

- In general, f^* unknown (and it may be $-\infty$)
- Solution: replace it with a target f_{lev}^k, revise it appropriately

$$0 \leq \nu_k = \beta_k \frac{f_k - f_{lev}^k}{\|d_k\|^2} \quad , \quad 0 < \beta^* \leq \beta_k \leq \alpha_k \leq 1$$

- Usually, $f_{lev}^k = f_{ref}^k$ (reference) $-\delta_k$ (threshold)
- Typical choice: $f_{ref}^k = f_{rec}^k \ = \ \min_{h \leq k} \ f(x_h)$ (record value)
- Looks uncorrected but it is not necessarily so:

$$\lambda_k = f_k - f_{lev}^k = f_k - f^* - (f_{ref}^k - f^* - \delta_k)$$

$\gamma_k = f_{ref}^k - f^* - \delta_k$ unknown

Target-level Approaches

- In general, f^* unknown (and it may be $-\infty$)
- Solution: replace it with a target f_{lev}^k, revise it appropriately

$$0 \le \nu_k = \beta_k \frac{f_k - f_{lev}^k}{\|d_k\|^2} \quad , \quad 0 < \beta^* \le \beta_k \le \alpha_k \le 1$$

- Usually, $f_{lev}^k = f_{ref}^k$ (reference) $-\delta_k$ (threshold)
- Typical choice: $f_{ref}^k = f_{rec}^k \; = \; \min_{h \le k} \; f(x_h)$ (record value)
- Looks uncorrected but it is not necessarily so:

$$\lambda_k = f_k - f_{lev}^k = f_k - f^* - (f_{ref}^k - f^* - \delta_k)$$

  $\gamma_k = f_{ref}^k - f^* - \delta_k$ unknown

- Small technical hurdle: all previous proofs require $f^* > -\infty$
- Solution: $f_{rec}^\infty = -\infty \Rightarrow f^* = -\infty$, otherwise feasible target $\bar{f} > -\infty$, $\bar{f} \ge f^*$, $\bar{f} \le f_{rec}^\infty$ ($\Rightarrow f_k - \bar{f} \ge 0$)

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = -\infty \ , \qquad \text{or} \qquad \liminf_{k \to \infty} \delta_k = \delta^* > 0$$

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = -\infty \;\;, \qquad \text{or} \qquad \liminf_{k\to\infty} \delta_k = \delta^* > 0$$

- Implementation: $\mu \in [0, 1)$

$$\delta_{k+1} \in \begin{cases} [\ \delta^*\ ,\ \infty\ ) & \text{if } f_{k+1} \leq f_{lev}^k \\ [\ \delta^*\ ,\ \max\{\ \delta^*\ ,\ \mu\delta_k\ \}\ ] & \text{if } f_{k+1} > f_{lev}^k \end{cases}$$

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f^{\infty}_{ref} = -\infty \;, \qquad \text{or} \qquad \liminf_{k\to\infty} \delta_k = \delta^* > 0$$

- Implementation: $\mu \in [0, 1)$

$$\delta_{k+1} \in \begin{cases} [\ \delta^*\ ,\ \infty\ ) & \text{if } f_{k+1} \leq f^k_{lev} \\ [\ \delta^*\ ,\ \max\{\ \delta^*\ ,\ \mu\delta_k\ \}\ ] & \text{if } f_{k+1} > f^k_{lev} \end{cases}$$

- Convergence: either $f^{\infty}_{ref} = -\infty = f^*$, or $f^{\infty}_{ref} \leq f^* + \xi\sigma^* + \delta^*$ where $0 \leq \xi = \max\{\ 1 - \delta^*\Gamma/2\sigma^*\ ,\ 0\ \} < 1$

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = -\infty \ , \qquad \text{or} \qquad \liminf_{k \to \infty} \delta_k = \delta^* > 0$$

- Implementation: $\mu \in [0, 1)$

$$\delta_{k+1} \in \begin{cases} [\ \delta^*\ ,\ \infty\ ) & \text{if } f_{k+1} \leq f_{lev}^k \\ [\ \delta^*\ ,\ \max\{\ \delta^*\ ,\ \mu\delta_k\ \}\ ] & \text{if } f_{k+1} > f_{lev}^k \end{cases}$$

- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \xi\sigma^* + \delta^*$ where $0 \leq \xi = \max\{\ 1 - \delta^*\Gamma/2\sigma^*\ ,\ 0\ \} < 1$

- Proof: (almost) straightforward, $\gamma^* \ \geq \ \xi\sigma^*$

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = -\infty \ , \qquad \text{or} \qquad \liminf_{k \to \infty} \delta_k = \delta^* > 0$$

- Implementation: $\mu \in [0, 1)$

$$\delta_{k+1} \in \begin{cases} [\ \delta^*\ ,\ \infty\ ) & \text{if } f_{k+1} \leq f_{lev}^k \\ [\ \delta^*\ ,\ \max\{\ \delta^*\ ,\ \mu\delta_k\ \}\ ] & \text{if } f_{k+1} > f_{lev}^k \end{cases}$$

- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \xi\sigma^* + \delta^*$ where $0 \leq \xi = \max\{\ 1 - \delta^*\Gamma/2\sigma^*\ ,\ 0\ \} < 1$

- Proof: (almost) straightforward, $\gamma^* \ \geq \ \xi\sigma^*$

- Compares favorably with $f_{ref}^{\infty} \leq f^* + \sigma^* + \delta^*$ (without deflection)[7]

Non-vanishing Threshold

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = -\infty \ , \qquad \text{or} \qquad \liminf_{k \to \infty} \delta_k = \delta^* > 0$$

- Implementation: $\mu \in [0, 1)$

$$\delta_{k+1} \in \begin{cases} [\ \delta^*\ ,\ \infty\ ) & \text{if } f_{k+1} \leq f_{lev}^k \\ [\ \delta^*\ ,\ \max\{\ \delta^*\ ,\ \mu\delta_k\ \}\ ] & \text{if } f_{k+1} > f_{lev}^k \end{cases}$$

- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \xi\sigma^* + \delta^*$ where $0 \leq \xi = \max\{\ 1 - \delta^*\Gamma/2\sigma^*\ ,\ 0\ \} < 1$

- Proof: (almost) straightforward, $\gamma^* \ \geq \ \xi\sigma^*$

- Compares favorably with $f_{ref}^{\infty} \leq f^* + \sigma^* + \delta^*$ (without deflection)[7]

- Note: it may seem that "small ξ is good", but $\xi\sigma^* + \delta^* \geq \sigma^*$

Vanishing Threshold

- Abstract property:

$$\text{either} \quad f_{ref}^{\infty} = f^* = -\infty \quad , \qquad \text{or} \quad \liminf_{k \to \infty} \delta_k = 0 \;\; \text{and} \;\; \sum_{k=1}^{\infty} \lambda_k / \|d_k\|^2 = \infty$$

[16] Lim, Sherali "Convergence . . . for Some Variable Target Value and Subgradient Deflection Methods", COAP, 2006

Vanishing Threshold

- Abstract property:

$$\text{either} \quad f_{ref}^{\infty} = f^* = -\infty \quad , \qquad \text{or} \quad \liminf_{k\to\infty} \delta_k = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} \lambda_k / \|d_k\|^2 = \infty$$

- Implementation: $R > 0$ and $\mu \in [0, 1)$
  - $f_{ref}^1 = f(x_1)$, $\delta_1 \in (0, \infty)$, $r_1 = 0$;
  - if $f_k \leq f_{ref}^k - \delta_k / 2$ (*sufficient descent condition*) then $f_{ref}^k = f_{rec}^k$, $r_k = 0$;
  - else, if $r_k > R$ (*target infeasibility condition*) then $\delta_k = \mu \delta_{k-1}$, $r_k = 0$;
  - otherwise, $f_{ref}^k = f_{ref}^{k-1}$, $\delta_k = \delta_{k-1}$, $r_k = r_{k-1} + \|\widehat{x}_{k+1} - x_k\|$

[16] Lim, Sherali “Convergence . . . for Some Variable Target Value and Subgradient Deflection Methods”, COAP, 2006

Vanishing Threshold

- Abstract property:

$$\text{either} \quad f_{ref}^{\infty} = f^* = -\infty \quad , \qquad \text{or} \quad \liminf_{k \to \infty} \delta_k = 0 \;\text{ and }\; \sum_{k=1}^{\infty} \lambda_k / \|d_k\|^2 = \infty$$

- Implementation: $R > 0$ and $\mu \in [0, 1)$
  - $f_{ref}^1 = f(x_1)$, $\delta_1 \in (0, \infty)$, $r_1 = 0$;
  - if $f_k \leq f_{ref}^k - \delta_k / 2$ (*sufficient descent condition*) then $f_{ref}^k = f_{rec}^k$, $r_k = 0$;
  - else, if $r_k > R$ (*target infeasibility condition*) then $\delta_k = \mu \delta_{k-1}$, $r_k = 0$;
  - otherwise, $f_{ref}^k = f_{ref}^{k-1}$, $\delta_k = \delta_{k-1}$, $r_k = r_{k-1} + \|\widehat{x}_{k+1} - x_k\|$
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Proof: again (almost) straightforward, $\gamma^* \geq \sigma^*$ ($\xi = 1$), minor quirks

[16] Lim, Sherali "Convergence . . . for Some Variable Target Value and Subgradient Deflection Methods", COAP, 2006

Vanishing Threshold

- Abstract property:

$$\text{either } \; f_{ref}^{\infty} = f^* = -\infty \;\; , \qquad \text{or } \; \liminf_{k\to\infty} \delta_k = 0 \; \text{ and } \; \sum_{k=1}^{\infty} \lambda_k / \|d_k\|^2 = \infty$$

- Implementation: $R > 0$ and $\mu \in [0, 1)$
  - $f_{ref}^1 = f(x_1)$, $\delta_1 \in (0, \infty)$, $r_1 = 0$;
  - if $f_k \leq f_{ref}^k - \delta_k/2$ (*sufficient descent condition*) then $f_{ref}^k = f_{rec}^k$, $r_k = 0$;
  - else, if $r_k > R$ (*target infeasibility condition*) then $\delta_k = \mu\delta_{k-1}$, $r_k = 0$;
  - otherwise, $f_{ref}^k = f_{ref}^{k-1}$, $\delta_k = \delta_{k-1}$, $r_k = r_{k-1} + \|\widehat{x}_{k+1} - x_k\|$
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Proof: again (almost) straightforward, $\gamma^* \geq \sigma^*$ ($\xi = 1$), minor quirks
- Optimal error, extends known results[16] to projection and errors

[16] Lim, Sherali "Convergence . . . for Some Variable Target Value and Subgradient Deflection Methods", COAP, 2006

Vanishing Threshold

- Abstract property:

$$\text{either} \quad f_{ref}^{\infty} = f^* = -\infty \quad , \qquad \text{or} \quad \liminf_{k \to \infty} \delta_k = 0 \quad \text{and} \quad \sum_{k=1}^{\infty} \lambda_k / \|d_k\|^2 = \infty$$

- Implementation: $R > 0$ and $\mu \in [0, 1)$
  - $f_{ref}^1 = f(x_1)$, $\delta_1 \in (0, \infty)$, $r_1 = 0$;
  - if $f_k \leq f_{ref}^k - \delta_k / 2$ (*sufficient descent condition*) then $f_{ref}^k = f_{rec}^k$, $r_k = 0$;
  - else, if $r_k > R$ (*target infeasibility condition*) then $\delta_k = \mu \delta_{k-1}$, $r_k = 0$;
  - otherwise, $f_{ref}^k = f_{ref}^{k-1}$, $\delta_k = \delta_{k-1}$, $r_k = r_{k-1} + \|\widehat{x}_{k+1} - x_k\|$
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Proof: again (almost) straightforward, $\gamma^* \geq \sigma^*$ ($\xi = 1$), minor quirks
- Optimal error, extends known results[16] to projection and errors
- Weaker results than (8) ($f^{\infty} \to f_{ref}^{\infty}$, no convergence of $\{x_k\}$)

[16] Lim, Sherali "Convergence ... for Some Variable Target Value and Subgradient Deflection Methods", COAP, 2006

Diminishing/Square Summable Stepsize

- Other main class of stepsize rules: diminishing/square summable

$$\sum_{k=1}^{\infty} \nu_k = \infty \quad , \quad \sum_{k=1}^{\infty} \nu_k^2 < \infty \tag{11}$$

Diminishing/Square Summable Stepsize

- Other main class of stepsize rules: diminishing/square summable

$$\sum_{k=1}^{\infty} \nu_k = \infty \quad , \quad \sum_{k=1}^{\infty} \nu_k^2 < \infty \tag{11}$$

  - Pros: do not need f^*, not even any estimate

Diminishing/Square Summable Stepsize

- Other main class of stepsize rules: diminishing/square summable

$$\sum_{k=1}^{\infty} \nu_k = \infty \quad , \quad \sum_{k=1}^{\infty} \nu_k^2 < \infty \tag{11}$$

  - Pros: do not need f^*, not even any estimate
  - Cons: no control over ε_k (cf. (5), (6))

- All our results hinge over these estimates

Diminishing/Square Summable Stepsize

- Other main class of stepsize rules: diminishing/square summable

$$\sum_{k=1}^{\infty} \nu_k = \infty \quad , \quad \sum_{k=1}^{\infty} \nu_k^2 < \infty \tag{11}$$

  - Pros: do not need f^*, not even any estimate
  - Cons: no control over ε_k (cf. (5), (6))

- All our results hinge over these estimates

- Solution: restrict the deflection instead of the stepsize

$$0 \leq \zeta_k = \frac{\nu_{k-1} \| d_{k-1} \|^2}{(\; f_k - f^* \;) + \nu_{k-1} \| d_{k-1} \|^2} \leq \alpha_k \leq 1$$

Diminishing/Square Summable Stepsize

- Other main class of stepsize rules: diminishing/square summable

$$\sum_{k=1}^{\infty} \nu_k = \infty \quad , \quad \sum_{k=1}^{\infty} \nu_k^2 < \infty \tag{11}$$

  - Pros: do not need f^*, not even any estimate
  - Cons: no control over ε_k (cf. (5), (6))

- All our results hinge over these estimates

- Solution: restrict the deflection instead of the stepsize

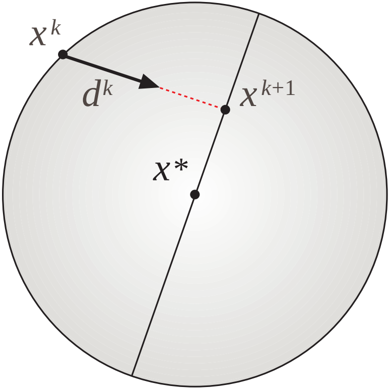
$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f^*\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

- Gives analogous to (5), (6)

$$\varepsilon_k \leq f_k - f^* + \bar{\sigma}_k \tag{12}$$

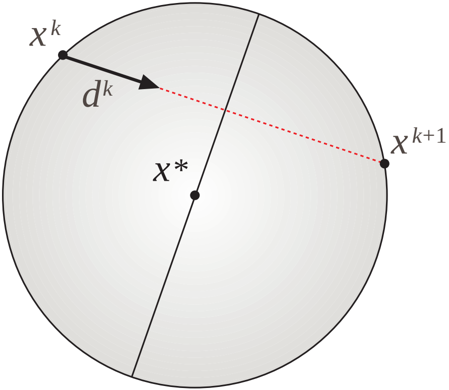
where $\bar{\sigma}_k = \alpha_k \sigma_k + (1 - \alpha_k)\bar{\sigma}_{k-1}$

Deflection Rule (geometrically)



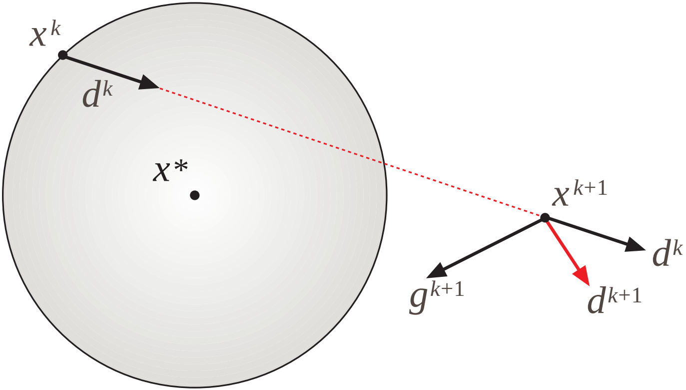
- Moving "towards x^*" is a short enough step

Deflection Rule (geometrically)



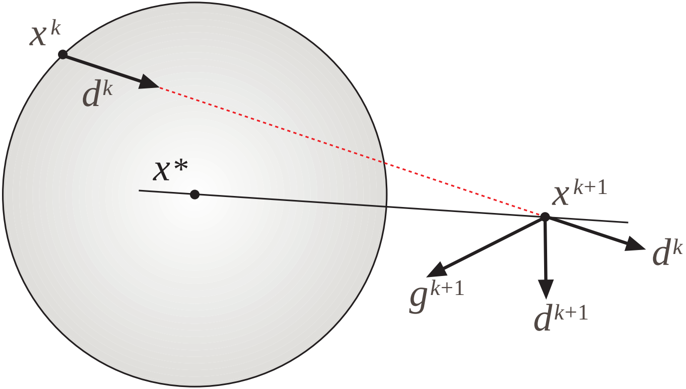
- Moving "towards x^*" is a short enough step and then any deflection

Deflection Rule (geometrically)



- Moving "towards x^*" is a short enough step and then any deflection
- . . . or any step

Deflection Rule (geometrically)



- Moving "towards x^*" is a short enough step and then any deflection
- . . . or any step and a proper deflection

Corrected Deflection Rule

- We learnt our lesson: corrected deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f^* - \gamma_k\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

Corrected Deflection Rule

- We learnt our lesson: corrected deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f^* - \gamma_k\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

- Avoid ζ_k is undefined ($\lambda_k = f_k - f^* - \gamma_k$):

$$\nu_{k-1}\|d_{k-1}\|^2 \leq \alpha_k(\lambda_k + \nu_{k-1}\|d_{k-1}\|^2) \tag{13}$$

- Avoid negative λ_k: makes (13) impossible

$$\begin{array}{l} \lambda_k \geq 0 \;\Rightarrow\; \alpha_k \geq \alpha^* > 0 \\ \lambda_k < 0 \;\Rightarrow\; \alpha_k = 0\ (\Rightarrow \nu_k = 0) \end{array} \tag{14}$$

Corrected Deflection Rule

- We learnt our lesson: corrected deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f^* - \gamma_k\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

- Avoid ζ_k is undefined ($\lambda_k = f_k - f^* - \gamma_k$):

$$\nu_{k-1}\|d_{k-1}\|^2 \leq \alpha_k(\lambda_k + \nu_{k-1}\|d_{k-1}\|^2) \qquad (13)$$

- Avoid negative λ_k: makes (13) impossible

$$\begin{array}{l} \lambda_k \geq 0 \ \Rightarrow \ \alpha_k \geq \alpha^* > 0 \\ \lambda_k < 0 \ \Rightarrow \ \alpha_k = 0 \ (\Rightarrow \nu_k = 0) \end{array} \qquad (14)$$

- Now ε_k is controlled: (12) holds with

$$\bar{\sigma}_k = \alpha_k(\sigma_k - \gamma_k) + (1 - \alpha_k)\bar{\sigma}_{k-1}$$

- Yields the crucial technical relationship, similar to (7)

$$\bar{d}_k(\bar{x} - x_k) \leq f(\bar{x}) - f^* + \bar{\sigma}_k$$

Convergence Results

- Relationships between σ^* and $\bar{\sigma}^*$:
  - in general, $\bar{\sigma}^* \leq \sigma^* + \bar{\gamma}$
  - $\gamma_k \geq \xi \sigma_k$ $\forall k$ large enough $\Rightarrow \bar{\sigma}^* \leq (1 - \xi)\sigma^*$

Convergence Results

- Relationships between σ^* and $\bar{\sigma}^*$:
  - in general, $\bar{\sigma}^* \leq \sigma^* + \bar{\gamma}$
  - $\gamma_k \geq \xi \sigma_k \; \forall k$ large enough $\Rightarrow \bar{\sigma}^* \leq (1 - \xi)\sigma^*$
- Convergence: under $\sup_k \; \|d_k\| < \infty$
  i) in general, $f^\infty \leq f^* + \gamma^{\sup} + (\sigma^* + \bar{\gamma})/\alpha^*$
  ii) $\gamma_k \geq \xi \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*(\, 1 + (1 - \xi)(1 - \alpha^*)/\alpha^* \,)$
  iii) $\gamma_k = \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*$
  furthermore, $X^* \neq \emptyset \Rightarrow \{x_k\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$

Convergence Results

- Relationships between σ^* and $\bar{\sigma}^*$:
  - in general, $\bar{\sigma}^* \leq \sigma^* + \bar{\gamma}$
  - $\gamma_k \geq \xi \sigma_k \ \forall k$ large enough $\Rightarrow \bar{\sigma}^* \leq (1 - \xi)\sigma^*$
- Convergence: under $\sup_k \ \|d_k\| < \infty$
  i) in general, $f^\infty \leq f^* + \gamma^{\text{sup}} + (\sigma^* + \bar{\gamma})/\alpha^*$
  ii) $\gamma_k \geq \xi \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*(\ 1 + (1 - \xi)(1 - \alpha^*)/\alpha^*\ )$
  iii) $\gamma_k = \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*$
  furthermore, $X^* \neq \emptyset \Rightarrow \{x_k\} \rightarrow x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
- Analogous to previous results, optimal error
- Boundedness assumption easily attained (bounding strategies[7])

Convergence Results

- Relationships between σ^* and $\bar{\sigma}^*$:
  - in general, $\bar{\sigma}^* \leq \sigma^* + \bar{\gamma}$
  - $\gamma_k \geq \xi\sigma_k \ \forall k$ large enough $\Rightarrow \bar{\sigma}^* \leq (1-\xi)\sigma^*$
- Convergence: under $\sup_k \|d_k\| < \infty$
  i) in general, $f^\infty \leq f^* + \gamma^{\sup} + (\sigma^* + \bar{\gamma})/\alpha^*$
  ii) $\gamma_k \geq \xi\sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*(\ 1 + (1-\xi)(1-\alpha^*)/\alpha^*\ )$
  iii) $\gamma_k = \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*$
  furthermore, $X^* \neq \emptyset \Rightarrow \{x_k\} \to x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
- Analogous to previous results, optimal error
- Boundedness assumption easily attained (bounding strategies[7])
- Technical notes: $\nu_k = 0$ from (14) at odds with the very (11) $\Rightarrow$ finite case to be considered carefully

Convergence Results

- Relationships between σ^* and $\bar{\sigma}^*$:
  - in general, $\bar{\sigma}^* \leq \sigma^* + \bar{\gamma}$
  - $\gamma_k \geq \xi \sigma_k \ \forall k$ large enough $\Rightarrow \bar{\sigma}^* \leq (1 - \xi)\sigma^*$
- Convergence: under $\sup_k \ \|d_k\| < \infty$

  i) in general, $f^\infty \leq f^* + \gamma^{\sup} + (\sigma^* + \bar{\gamma})/\alpha^*$

  ii) $\gamma_k \geq \xi \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*(\ 1 + (1 - \xi)(1 - \alpha^*)/\alpha^* \ )$

  iii) $\gamma_k = \sigma_k \Rightarrow f^\infty \leq f^* + \sigma^*$
  furthermore, $X^* \neq \emptyset \Rightarrow \{x_k\} \rightarrow x^\infty \in X$ s.t. $f(x^\infty) = f^\infty$
- Analogous to previous results, optimal error
- Boundedness assumption easily attained (bounding strategies[7])
- Technical notes: $\nu_k = 0$ from (14) at odds with the very (11) $\Rightarrow$ finite case to be considered carefully
- As usual, f^* not available (and may be $-\infty$) $\Rightarrow$ same trick

Target Value Deflection

- Target value deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1} \| d_{k-1} \|^2}{(f_k - f_{lev}^k) + \nu_{k-1} \| d_{k-1} \|^2} \leq \alpha_k \leq 1$$

(as before, looks uncorrected but it is not: γ_k unknown)

Target Value Deflection

- Target value deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f_{lev}^k\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

(as before, looks uncorrected but it is not: γ_k unknown)

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = f^* = -\infty \ , \qquad \text{or} \qquad \liminf_{k\to\infty} \delta_k = 0 \ .$$

Target Value Deflection

- Target value deflection rule

$$0 \leq \zeta_k = \frac{\nu_{k-1}\|d_{k-1}\|^2}{(\ f_k - f_{lev}^k\ ) + \nu_{k-1}\|d_{k-1}\|^2} \leq \alpha_k \leq 1$$

(as before, looks uncorrected but it is not: γ_k unknown)

- Abstract property:

$$\text{either} \qquad f_{ref}^{\infty} = f^* = -\infty \ , \qquad \text{or} \qquad \liminf_{k \to \infty} \delta_k = 0 \ .$$

- Implementation:

$$\delta_{k+1} \in \begin{cases} [\ \Delta_{r(k)+1}\ ,\ \infty\ ) & \text{if } f(x_{k+1}) \leq f_{lev}^k \\ \{\Delta_{k+1}\} & \text{if } f(x_{k+1}) > f_{lev}^k \end{cases}$$

where $r(k) = \#h \leq k$ s.t. $f_{h+1} \leq f_{lev}^h$ and

$$\Delta_k > 0 \qquad , \qquad \liminf_{k \to \infty} \Delta_k = 0 \qquad , \qquad \sum_{k=1}^{\infty} \Delta_k = \infty$$

Target Value Deflection (cont.d)

- Similar technical hurdles (reference value, ...)
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Easy proof (all the dirty work done already)

Target Value Deflection (cont.d)

- Similar technical hurdles (reference value, . . .)
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Easy proof (all the dirty work done already)
- Same as stepsize-restricted (but it was not obvious beforehand)

Target Value Deflection (cont.d)

- Similar technical hurdles (reference value, . . .)
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Easy proof (all the dirty work done already)
- Same as stepsize-restricted (but it was not obvious beforehand)

Conclusions (for now)

Target Value Deflection (cont.d)

- Similar technical hurdles (reference value, ...)
- Convergence: either $f^{\infty}_{ref} = -\infty = f^*$, or $f^{\infty}_{ref} \leq f^* + \sigma^*$
- Easy proof (all the dirty work done already)
- Same as stepsize-restricted (but it was not obvious beforehand)

Conclusions (for now)

1. If σ^* is your error, then $f^* + \sigma^*$ is your target

Target Value Deflection (cont.d)

- Similar technical hurdles (reference value, ...)
- Convergence: either $f_{ref}^{\infty} = -\infty = f^*$, or $f_{ref}^{\infty} \leq f^* + \sigma^*$
- Easy proof (all the dirty work done already)
- Same as stepsize-restricted (but it was not obvious beforehand)

Conclusions (for now)

1. If σ^* is your error, then $f^* + \sigma^*$ is your target
2. Knowing σ_k, even approximately, is useful

Bundle Methods

(with Giovanni Giallombardo)

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$

[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$
- Idea: use $\mathcal{B}$ to construct a model $f^k_{\mathcal{B}}$ of f, e.g.

$$\hat{f}^k_{\mathcal{B}}(x) = \sup_{\bar{z}} \{ \bar{z}x - f^*(\bar{z}) \ : \ \bar{z} \in \mathcal{B} \}$$

(cutting plane model)

[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$

- Idea: use $\mathcal{B}$ to construct a model $f^k_{\mathcal{B}}$ of f, e.g.

$$\hat{f}^k_{\mathcal{B}}(x) = \sup_{\bar{z}} \{\ \bar{z}x - f^*(\bar{z})\ :\ \bar{z} \in \mathcal{B}\ \}$$

(cutting plane model)

- Immediate consequence: cutting plane algorithm

$$x_{k+1} = \text{argmin} \{\ \hat{f}^k_{\mathcal{B}}(x)\ :\ x \in X\ \}$$

[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$
- Idea: use $\mathcal{B}$ to construct a model $f^k_{\mathcal{B}}$ of f, e.g.

$$\hat{f}^k_{\mathcal{B}}(x) = \sup_{\bar{z}} \{ \; \bar{z}x - f^*(\bar{z}) \; : \; \bar{z} \in \mathcal{B} \; \}$$

(cutting plane model)

- Immediate consequence: cutting plane algorithm

$$x_{k+1} = \operatorname{argmin} \{ \; \hat{f}^k_{\mathcal{B}}(x) \; : \; x \in X \; \}$$

- Simple to implement, one linear program at each iteration

[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$
- Idea: use $\mathcal{B}$ to construct a model $f^k_{\mathcal{B}}$ of f, e.g.

$$\hat{f}^k_{\mathcal{B}}(x) = \sup_{\bar{z}} \{ \, \bar{z}x - f^*(\bar{z}) \; : \; \bar{z} \in \mathcal{B} \, \}$$

  (cutting plane model)

- Immediate consequence: cutting plane algorithm

$$x_{k+1} = \mathrm{argmin} \{ \, \hat{f}^k_{\mathcal{B}}(x) \; : \; x \in X \, \}$$

- Simple to implement, one linear program at each iteration
- Unfortunately, often rather slow in practice (with exceptions)[17]

[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

(exact) Bundle Methods: the Basic Ideas

- Any iterative algorithm produces a sequence $\{x_k\}$ of tentative points $\Rightarrow$ the f-values sequence $\{f_k\}$ and the bundle $\mathcal{B} = \{z_k \in \partial f(x_k)\}$
- Idea: use $\mathcal{B}$ to construct a model $f^k_{\mathcal{B}}$ of f, e.g.

$$\hat{f}^k_{\mathcal{B}}(x) = \sup_{\bar{z}} \{ \; \bar{z}x - f^*(\bar{z}) \; : \; \bar{z} \in \mathcal{B} \; \}$$

(cutting plane model)

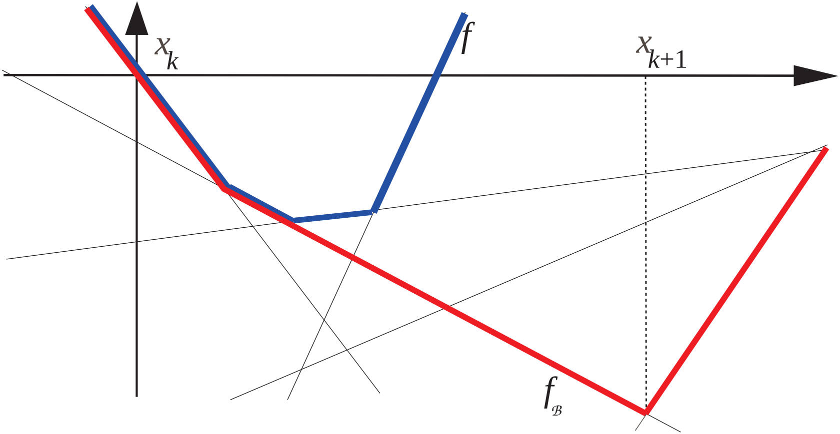
- Immediate consequence: cutting plane algorithm

$$x_{k+1} = \text{argmin} \{ \; \hat{f}^k_{\mathcal{B}}(x) \; : \; x \in X \; \}$$

- Simple to implement, one linear program at each iteration
- Unfortunately, often rather slow in practice (with exceptions)[17]
- Problem: instability

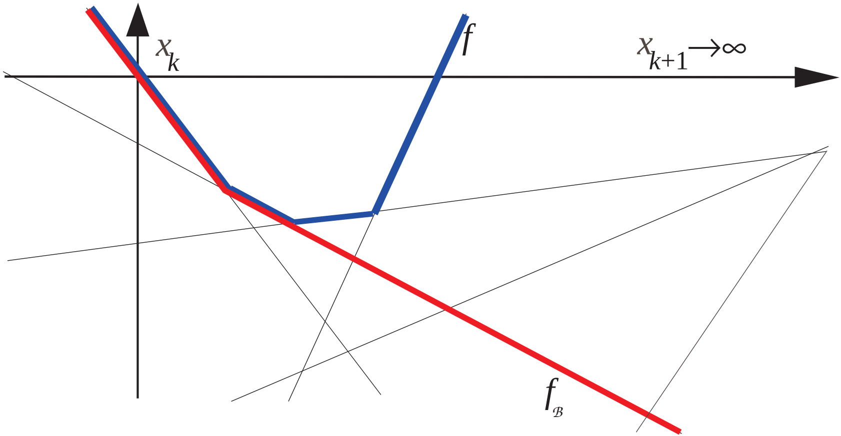
[17] Jones, Lustig, Farwolden, Powell "Multicommodity Network Flows: The Impact of Formulation on Decomposition" Math. Prog., 1993

Instability and Stabilization



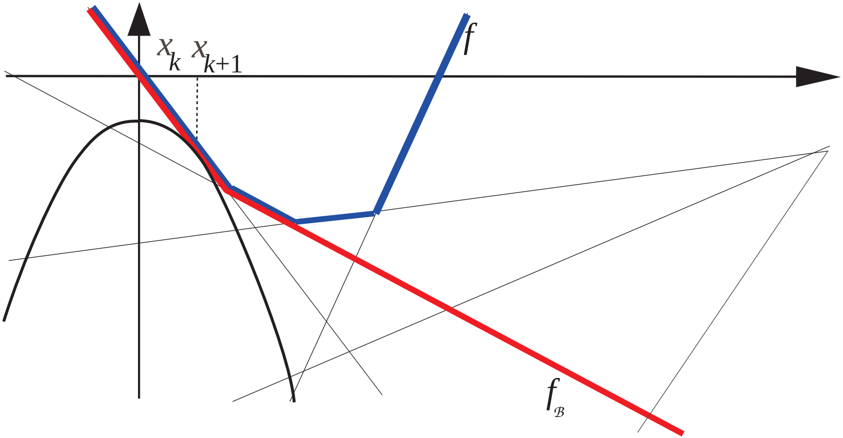
- Issue: x_{k+1} can be far from x_k

Instability and Stabilization



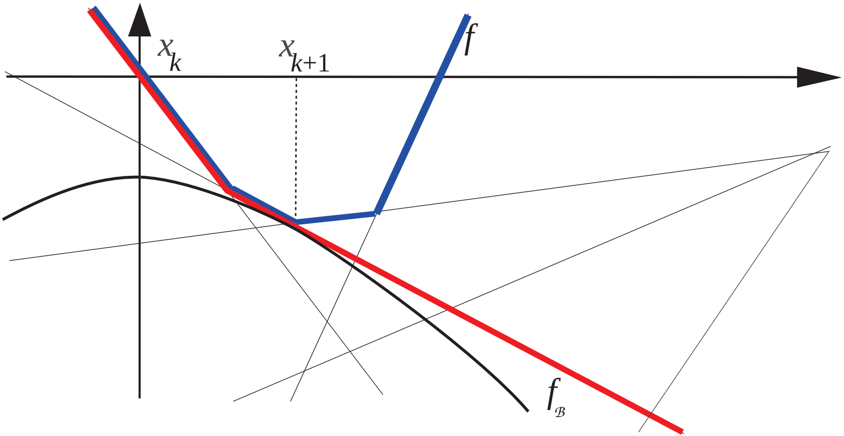
- Issue: x_{k+1} can be far from x_k ... even infinitely far

Instability and Stabilization



- Issue: x_{k+1} can be far from x_k . . . even infinitely far
- Solution: stabilize the model

Instability and Stabilization



- Issue: x_{k+1} can be far from x_k ...even infinitely far
- Solution: stabilize the model ...with the right weight

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \left\{ \, f(\bar{x} + d) + D_t(d) \, \right\} \tag{15}$$

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \left\{ f(\bar{x} + d) + D_t(d) \right\} \tag{15}$$

  - current point $\bar{x}$

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \{ \ f(\bar{x} + d) + D_t(d) \ \} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \{ \ f(\bar{x} + d) + D_t(d) \ \} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \left\{ \, f(\bar{x} + d) + D_t(d) \, \right\} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

- With proper D_t, good properties (e.g. smooth)

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \left\{ f(\bar{x} + d) + D_t(d) \right\} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

- With proper D_t, good properties (e.g. smooth)

- But computing ϕ_t with an oracle for f is difficult $\Rightarrow$ approximation

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \{ \; f(\bar{x} + d) + D_t(d) \; \} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

- With proper D_t, good properties (e.g. smooth)

- But computing ϕ_t with an oracle for f is difficult $\Rightarrow$ approximation

- Stabilized primal master problem

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \phi_{\mathcal{B},t}(\bar{x}) = \inf_d \{ \; f_{\mathcal{B}}(\bar{x} + d) + D_t(d) \; \} \tag{16}$$

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \{ \; f(\bar{x} + d) + D_t(d) \; \} \tag{15}$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

- With proper D_t, good properties (e.g. smooth)

- But computing ϕ_t with an oracle for f is difficult $\Rightarrow$ approximation

- Stabilized primal master problem

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \phi_{\mathcal{B},t}(\bar{x}) = \inf_d \{ \; f_{\mathcal{B}}(\bar{x} + d) + D_t(d) \; \} \tag{16}$$

  - $x_{k+1} = \bar{x} + d^*$, compute f_{k+1}, $\mathcal{B} = \mathcal{B} \cup \{z_{k+1}\}$

Primal View of (Generalized) Bundle Methods

- Stabilization: stabilized primal problem ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\bar{x},t}) \qquad \phi_t(\bar{x}) = \inf_d \{ \ f(\bar{x} + d) + D_t(d) \ \} \qquad (15)$$

  - current point $\bar{x}$
  - ϕ_t = (generalized) Moreau–Yosida regularization of f
  - D_t = stabilizing term ($\approx$ norm), t = proximity weight

- With proper D_t, good properties (e.g. smooth)

- But computing ϕ_t with an oracle for f is difficult $\Rightarrow$ approximation

- Stabilized primal master problem

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \phi_{\mathcal{B},t}(\bar{x}) = \inf_d \{ \ f_{\mathcal{B}}(\bar{x} + d) + D_t(d) \ \} \qquad (16)$$

  - $x_{k+1} = \bar{x} + d^*$, compute f_{k+1}, $\mathcal{B} = \mathcal{B} \cup \{z_{k+1}\}$
  - if $f_{k+1} \ll f(\bar{x})$, then $\bar{x} = x_{k+1}$

Dual View of (Generalized) Bundle Methods

- Dual of (Π)[18]: $\quad (\Delta) \qquad f^*(0) \;=\; \inf_z \left\{ \, f^*(z) \;:\; z = 0 \, \right\}$

[18] F. "Generalized Bundle Methods", SIOPT, 2002

Dual View of (Generalized) Bundle Methods

- Dual of (Π)[18]: $\quad (\Delta) \qquad f^*(0) \;=\; \inf_z \{\, f^*(z) \;:\; z = 0 \,\}$
- May look funny, but then every f is a Lagrangian function:

$$(\Delta_{\bar{x}}) \qquad f(\bar{x}) = -\inf_z \{\, f^*(z) - z\bar{x} \,\}$$

[18] F. "Generalized Bundle Methods", SIOPT, 2002

Dual View of (Generalized) Bundle Methods

- Dual of (Π)[18]: $\quad (\Delta) \qquad f^*(0) \;=\; \inf_z \{\; f^*(z) \;:\; z = 0 \;\}$
- May look funny, but then every f is a Lagrangian function:

$$(\Delta_{\bar{x}}) \qquad f(\bar{x}) = -\inf_z \{\; f^*(z) - z\bar{x} \;\}$$

- Further, (15) has a non-weird (Fenchel's) dual

$$(\Delta_{\bar{x},t}) \qquad \inf_z \{\; f^*(z) - z\bar{x} + D^*{}_t(-z) \;\}$$

$=$ (generalized) Augmented Lagrangian of (Δ) $\Rightarrow$ so has (16)

$$(\Delta_{\mathcal{B},\bar{x},t}) \qquad \inf_z \{\; f^*_{\mathcal{B}}(z) - z\bar{x} + D^*_t(-z) \;\}$$

[18] F. "Generalized Bundle Methods", SIOPT, 2002

Dual View of (Generalized) Bundle Methods

- Dual of (Π)[18]: $\quad (\Delta) \qquad f^*(0) \; = \; \inf_z \{ \; f^*(z) \; : \; z = 0 \; \}$
- May look funny, but then every f is a Lagrangian function:

$$(\Delta_{\bar{x}}) \qquad f(\bar{x}) = -\inf_z \{ \; f^*(z) - z\bar{x} \; \}$$

- Further, (15) has a non-weird (Fenchel's) dual

$$(\Delta_{\bar{x},t}) \qquad \inf_z \{ \; f^*(z) - z\bar{x} + D^*{}_t(-z) \; \}$$

$=$ (generalized) Augmented Lagrangian of $(\Delta) \Rightarrow$ so has (16)

$$(\Delta_{\mathcal{B},\bar{x},t}) \qquad \inf_z \{ \; f^*_{\mathcal{B}}(z) - z\bar{x} + D^*_t(-z) \; \}$$

- Illustration: $f_{\mathcal{B}} = \hat{f}_{\mathcal{B}}$, $g(u) = Au - b$, $x \geq 0$

$$(\Delta_{\mathcal{B},\bar{x},t}) \equiv \sup_u \left\{ \begin{array}{l} c(u) + \bar{x}z - D^*_t(-z) \\ z = b + \omega - Au \; , \; \omega \geq 0 \; , \; u \in co\,\mathcal{B} \subseteq U \end{array} \right.$$

$\Rightarrow$ actually solving the weird convexification (3)

[18] F. "Generalized Bundle Methods", SIOPT, 2002

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$
- Better yet: use separate models $f^h_{\mathcal{B}}$ for each component

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$
- Better yet: use separate models $f^h_{\mathcal{B}}$ for each component
- Disaggregated master problems ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \inf_d \left\{ \sum_{h \in \mathcal{K}} f^h_{\mathcal{B}}(\bar{x} + d) + D_t(d) \right\}$$

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \inf_z \left\{ \sum_{h \in \mathcal{K}} (f^h_{\mathcal{B}})^*(z^h) - \left(\sum_{h \in \mathcal{K}} z^h \right) \bar{x} + D^*_t \left(- \sum_{h \in \mathcal{K}} z^h \right) \right\}$$

---

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$
- Better yet: use separate models $f^h_{\mathcal{B}}$ for each component
- Disaggregated master problems ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \inf_d \left\{ \sum_{h \in \mathcal{K}} f^h_{\mathcal{B}}(\bar{x} + d) + D_t(d) \right\}$$

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \inf_z \left\{ \sum_{h \in \mathcal{K}} (f^h_{\mathcal{B}})^*(z^h) - \left(\sum_{h \in \mathcal{K}} z^h \right) \bar{x} + D^*_t \left(- \sum_{h \in \mathcal{K}} z^h \right) \right\}$$

- Often more efficient in practice[17] [19], for good reasons

---

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$
- Better yet: use separate models $f^h_{\mathcal{B}}$ for each component
- Disaggregated master problems ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \inf_d \left\{ \sum_{h \in \mathcal{K}} f^h_{\mathcal{B}}(\bar{x} + d) + D_t(d) \right\}$$

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \inf_z \left\{ \sum_{h \in \mathcal{K}} (f^h_{\mathcal{B}})^*(z^h) - \left(\sum_{h \in \mathcal{K}} z^h \right) \bar{x} + D^*_t\left(- \sum_{h \in \mathcal{K}} z^h \right) \right\}$$

- Often more efficient in practice[17] [19], for good reasons
- Master problem more costly to solve, but faster convergence

---

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

The Decomposable Case

- $f(x) = \sum_{h \in \mathcal{K}} f^h(x)$, computing each f^h produces $z^h \in \partial f^h(x)$
- Can aggregate: $\sum_{h \in \mathcal{K}} z^h = z \in \partial f(x)$
- Better yet: use separate models $f^h_{\mathcal{B}}$ for each component
- Disaggregated master problems ($X = \mathbb{R}^n$ for simplicity)

$$(\Pi_{\mathcal{B},\bar{x},t}) \qquad \inf_d \left\{ \sum_{h \in \mathcal{K}} f^h_{\mathcal{B}}(\bar{x} + d) + D_t(d) \right\}$$

$$(\Delta_{\mathcal{B},\bar{x},t}) \quad \inf_z \left\{ \sum_{h \in \mathcal{K}} (f^h_{\mathcal{B}})^*(z^h) - \left(\sum_{h \in \mathcal{K}} z^h \right) \bar{x} + D^*_t\left(- \sum_{h \in \mathcal{K}} z^h \right) \right\}$$

- Often more efficient in practice[17] [19], for good reasons
- Master problem more costly to solve, but faster convergence
- No incremental version as yet

[19] Bacaud, Lemaréchal, Renaud, Sagastizábal "Bundle methods in stochastic optimal power management: a disaggregated approach using preconditioners" COAP, 2001

Approximate Bundle Methods

- Proposal exist only using lower bound [8] [9] or for finite min-max[20]
- Unify and extend these.

[20] Gaudioso, Giallombardo, Miglionico "An Incremental Method for Solving Convex Finite Minmax Problems" Math. of O.R., 2006

Approximate Bundle Methods

- Proposal exist only using lower bound [8] [9] or for finite min-max[20]
- Unify and extend these.

Definition

Incremental inexact oracle for f: inputs $\bar{x} \in \Re^n$, outputs:

- $\underline{f} \leq f(\bar{x})$, $z \in \Re^n$ s.t. $\underline{f} + z(x - \bar{x}) \leq f(x)$ $\forall x$ (lower linearization)
- $\bar{f} \geq f(\bar{x})$ (upper bound, may be $+\infty$)

Can be called repeatedly on the same $\bar{x}$.

- Different rules governing the produced sequences $\{\underline{f}_j\}$, $\{\bar{f}_j\}$

[20] Gaudioso, Giallombardo, Miglionico "An Incremental Method for Solving Convex Finite Minmax Problems" Math. of O.R., 2006

Approximate Bundle Methods

- Proposal exist only using lower bound [8] [9] or for finite min-max[20]
- Unify and extend these.

Definition

Incremental inexact oracle for f: inputs $\bar{x} \in \Re^n$, outputs:

- $\underline{f} \leq f(\bar{x})$, $z \in \Re^n$ s.t. $\underline{f} + z(x - \bar{x}) \leq f(x)$ $\forall x$ (lower linearization)
- $\bar{f} \geq f(\bar{x})$ (upper bound, may be $+\infty$)

Can be called repeatedly on the same $\bar{x}$.

- Different rules governing the produced sequences $\{\underline{f}_j\}$, $\{\bar{f}_j\}$
- Bundle algorithm works in different "modes" (LB/UB following)

[20] Gaudioso, Giallombardo, Miglionico "An Incremental Method for Solving Convex Finite Minmax Problems" Math. of O.R., 2006

Approximate Bundle Methods

- Proposal exist only using lower bound [8] [9] or for finite min-max[20]
- Unify and extend these.

Definition

Incremental inexact oracle for f: inputs $\bar{x} \in \Re^n$, outputs:

- $\underline{f} \leq f(\bar{x})$, $z \in \Re^n$ s.t. $\underline{f} + z(x - \bar{x}) \leq f(x)$ $\forall x$ (lower linearization)
- $\bar{f} \geq f(\bar{x})$ (upper bound, may be $+\infty$)

Can be called repeatedly on the same $\bar{x}$.

- Different rules governing the produced sequences $\{\underline{f}_j\}$, $\{\bar{f}_j\}$
- Bundle algorithm works in different "modes" (LB/UB following)
- Results still preliminary, but knowing the gap helps

[20] Gaudioso, Giallombardo, Miglionico "An Incremental Method for Solving Convex Finite Minmax Problems" Math. of O.R., 2006

Conclusions

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008

Conclusions

- Errors are a fact of life

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008

Conclusions

- Errors are a fact of life
- You can pretend they don't exist, but you're better off not to

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008

Conclusions

- Errors are a fact of life
- You can pretend they don't exist, but you're better off not to
- Knowing something about them helps

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008

Conclusions

- Errors are a fact of life
- You can pretend they don't exist, but you're better off not to
- Knowing something about them helps
- Errors may even be a good thing

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008

Conclusions

- Errors are a fact of life
- You can pretend they don't exist, but you're better off not to
- Knowing something about them helps
- Errors may even be a good thing
- Lots of work still to be done
  - incremental subgradient
  - "dual" subgradient convergence[21]
  - incremental bundle
  - software development/refinement, numerical testing

---

[21] Nesterov "Primal-dual subgradient methods for convex problems" Math. Prog., 2008