Nonsmooth analysis for stochastic optimization : theory, algorithms and applications in energy

Jérôme Malick, CNRS, LJK

based on joint work with Wim van Ackooij (EDF), Nicolas Lebbe (PhD), Welington de Oliveria (Rio, Brasil), and Sofia Zaourar (Xerox)

Journées SMAI-MAS (Modélisation Aléatoire et Statistique)

Grenoble – August 2016

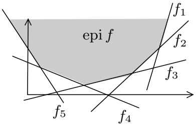
About the key-words of the title

"Nonsmooth analysis for stochastic optimization: theory, algorithms and applications in energy"

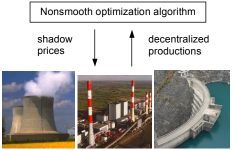
1. Stochastic optimization : optimization problems with random variables... in action !
   (vs randomized optimization algorithms and convergence analysis)

2. Nonsmooth : non-differentiable (e.g. max-fct)

   $\rightarrow$ difficult nonsmooth functions call for cutting-planes algorithms
   (vs proximal or conditional gradient algos)



3. Energy : electricity generation (EDF) with randomness due to renewable sources
   (vs distribution, prediction, savings,...)



About this talk

- This talk can be seen as a (light, incomplete, biased) introduction to stochastic optimization, from a nonsmooth analysis perspective
- Bird-eye view
  - presentation of advanced algorithms on pictures !
  - presentation of real-life industrial problems in words/photos !
- Many technical details hidden (modeling issues, assumptions, mathematical details, convergence analysis,...)
- Emphasis on ideas and applications – with a goal :

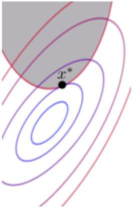
  advertize nonsmooth optimization for real-life energy problems

- Let's pick random topics in this talk: probability functions, eventual convexity, robust optimization, large-scale heterogeneous problems, decomposition algorithm, quadratic stabilization,...

Outline

Outline

A domain of applied maths : Mathematical Optimization



- Mathematical optimization
  ($\simeq$ the maths of "doing-better" or of the decision)

$$\left\{\begin{array}{ll} \min \quad f(x) & \text{(objective-function)} \\ \quad g(x) \leqslant 0 & \text{(constraints)} \\ \quad x \in X \subset \mathbb{R}^n & \end{array}\right.$$

- Intractable in general... but "easy" for linear optimization (f and g affine) or more generally for convex optimization (f and g convex) due to nice geometrical properties (globality, duality, guarantees...)

- Applications : recent explosion in data science
  on top of traditional applications in industry, services,... and energy !

- Basic example in machine learning or robust regression : LASSO
  (f data-fidelity term, g regularizer)

$$\left\{\begin{array}{ll} \min & \sum_{i=1}^n (a_i{}^\top x - b_i)^2 \\ & \|x\|_{\ell_1} \leqslant \mu \end{array}\right. \quad \text{or} \quad \left\{\begin{array}{ll} \min & \sum_{i=1}^n (a_i{}^\top x - b_i)^2 + \lambda \|x\|_{\ell_1} \\ & x \in \mathbb{R}^n \end{array}\right.$$

Optimization face to uncertainty

Adding unknown uncertainty ξ

$$\begin{cases} \min_{(x,\xi)} \quad f(x,\xi) \\ \quad g(x,\xi) \leqslant 0 \end{cases} \qquad \text{(not well-posed)}$$

Two main ways to model uncertainty

1. Robust optimization: $\xi \in \Xi$ with a known uncertainty set

$$\begin{cases} \min_x \quad \max_{\xi \in \Xi} \quad f(x,\xi) \\ \quad g(x,\xi) \leqslant 0 \quad \text{for all } \xi \in \Xi \end{cases}$$

2. Stochastic optimization: $\xi \sim \mathbb{P}$ with a (known) probability law

$$\begin{cases} \min_x \quad \mathbb{E}[f(x,\xi)] \\ \quad \mathbb{P}[g(x,\xi) \leqslant 0] \geqslant p \end{cases}$$

for an uncertainty level p ($p = 1$ is almost-sure)

In practice: modeling is as important as solving the resulting optimization problem (and both interact ! see forthcoming examples)

Multi-stage stochastic optimization

- In some problems, we can take a second decision (= correction) once the uncertainty is known

$$x \rightsquigarrow \xi \rightsquigarrow y$$

The "recourse" variable y is a random variable depending on ξ

- This yields to "two-stage" stochastic optimization problems

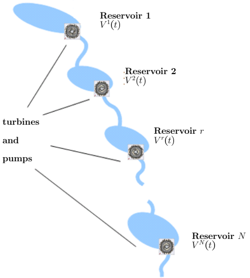
$$\begin{cases} \min_x \quad \mathbb{E}[f(x,\xi,y)] \\ \quad \mathbb{P}[g(x,\xi,y) \leqslant 0] \geqslant p \end{cases}$$

- Examples from this morning : 2-stage linear
- Generalizes to multi-stage

$$x^0 \rightsquigarrow \xi^1 \rightsquigarrow x^1 \rightsquigarrow \xi^2 \rightsquigarrow \cdots \rightsquigarrow x^T$$

- Stochastic optimization problems are more complex, but very structured $\rightarrow$ to be exploited in solution algorithms !

Example 1 : hydro-reservoir managment



Hydrological valley with N reservoirs

ξ random inflows

x decision variables (turbining, pumping)

$V^i(t) = V_0^i + A^i x(t) + \xi(t)$ volume of reservoir i

Simplified model, see more in vanAckooij Henrion Moller Zorgati '14

- variable : $x \in X$ to be decided upon before observing ξ
- objective : maximizing remuneration for a price signal π
- bounds constraints : $\forall i, t,\ V^i_{\min} \leqslant V^i(t) \leqslant V^i_{\max}$ (irrigation, navigability,...)

$$\boxed{\left\{\begin{array}{l} \max \quad \pi^\top x \\ \quad \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}] \geqslant p \\ \quad x \in X \end{array}\right.}$$

probability-constrained optimization problem

Probability constraints

- In the example : if we assume ξ gaussian, we have good properties :
  - $\varphi(x) = \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}]$ is log-concave Prekopa '95
  - φ is differentiable (when cov. matrix is def. pos.) vanAckooij Henrion '10
  - Efficient numerical integration scheme Deak '00
    to compute the value $\varphi(x)$ together with the gradient $\nabla\varphi(x)$

Probability constraints

- In the example : if we assume ξ gaussian, we have good properties :
  - $\varphi(x) = \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}]$ is log-concave Prekopa '95
  - φ is differentiable (when cov. matrix is def. pos.) vanAckooij Henrion '10
  - Efficient numerical integration scheme Deak '00
    to compute the value $\varphi(x)$ together with the gradient $\nabla\varphi(x)$

- In general : many theoretical questions on $\varphi(x) = \mathbb{P}[g(x, \xi) \leqslant 0]$
  - simple counter-examples show that
    continuity, convexity, differentiability of g do not transfert trivially to φ
  - important question in view of optimization :
    convexity of the constraint $\varphi(x) \geqslant p$?

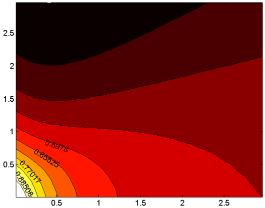
Probability constraints

- In the example : if we assume ξ gaussian, we have good properties :
  - $\varphi(x) = \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}]$ is log-concave Prekopa '95
  - φ is differentiable (when cov. matrix is def. pos.) vanAckooij Henrion '10
  - Efficient numerical integration scheme Deak '00
    to compute the value $\varphi(x)$ together with the gradient $\nabla\varphi(x)$

- In general : many theoretical questions on $\varphi(x) = \mathbb{P}[g(x,\xi) \leqslant 0]$
  - simple counter-examples show that
    continuity, convexity, differentiability of g do not transfert trivially to φ
  - important question in view of optimization :
    convexity of the constraint $\varphi(x) \geqslant p$?

- In particular : recent result vanAckooij Malick '16
  Under mild assumption on g, eventual convexity holds for a large family of laws (normal, log-normal, student,...) :
  there exists $\bar{p} < 1$ such that

$$\bar{p} \leqslant p \leqslant 1 \quad \{x : \mathbb{P}[g(x,\xi) \leqslant 0] \geqslant p\} \text{ convex}$$



Example 2 : optimization of electricity generation

- In France : electricity produced by $N \simeq 200$ production units

nuclear 80%

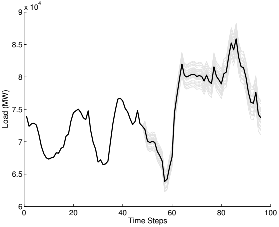
oil + coal 3%

water 17%

- Day-to-day optimization of production : finding a minimal cost production schedule for the next day that satisfies the operational constraints and that meets customer demand

- Classical problem : "unit-commitment"

- Uncertainty on weather conditions
  $\rightarrow$ consumption
  $\rightarrow$ renewable sources (wind, solar, water)
  $\rightarrow$ demand is uncertain



Deterministic unit-commitment

Simplified model : for $N \simeq 200$ units, on $T = 96 = 2 * 48$ periods of time

- variable : schedule $x = (x_1, \dots, x_N) \in X = X_1 \times \cdots \times X_N$
- technical constraints : $x_i \in X_i \quad i = 1, \dots, N$
- demand constraints : $m^t \leqslant \sum_i x_i^t - d^t \leqslant M^t \quad t = 1, \dots, T$
- objective : linear costs $c^\top x = \sum_i {c_i}^\top x_i$

Hard optimization problem: large-scale, heterogeneous, complex

$$\begin{cases} \min & c^\top x \\ & x \in X, \quad m \leqslant \sum_i x_i - d \leqslant M \end{cases}$$

Approach by duality : decomposition of computation over each unit

$$\begin{cases} \min & {\pi_i}^\top x_i \\ & x_i \in X_i \end{cases}$$

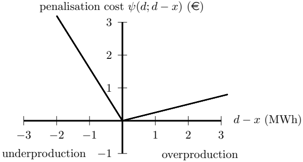
(with $\pi_i = c_i + A^\top \lambda$ where λ dual variable or price)

$$\boxed{\text{how to handle random } d = \xi \text{ in this situation ?}}$$

Model 1 : Robust unit-commitment

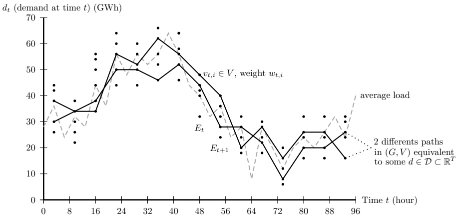
A simple robust approach
(VanAckooij Lebbe Malick '16)

- get rid of bound constraint
- penalize instead the worst gap



$$\begin{cases} \min \quad c^\top x + \max\limits_{\xi \in \Xi} \sum_{t=1}^{T} \psi\big(\sum_i x_i^t - \xi^t\big) \\ \quad x \in X \end{cases}$$

Complex model of uncertainty set Ξ (vs Ξ finite or $\Xi = [d_{\min}, d_{\max}]^T$)



The model of Minoux 2012

- is finite but of high cardinality
- expresses temporal dependencies
- preserves a fast computability

Model 2 : Two-stage stochastic unit-commitment 1/2

- The schedule x is sent to the grid-operator (RTE) before being activated and before observing uncertainty
- But in real time, a new production schedule can be sent to the grid-operator at specific moments in time
- After τ periods :
  - $\xi_1, ..., \xi_\tau$: the observed net customer load of the previous time
  - $\xi_{\tau+1}, ..., \xi_T$: the current best forecast of net customer load after τ

Model 2 : Two-stage stochastic unit-commitment 1/2

- The schedule x is sent to the grid-operator (RTE) before being activated and before observing uncertainty
- But in real time, a new production schedule can be sent to the grid-operator at specific moments in time
- After τ periods :
  - $\xi_1, ..., \xi_\tau$: the observed net customer load of the previous time
  - $\xi_{\tau+1}, ..., \xi_T$: the current best forecast of net customer load after τ
- vanAckooij Malick '15 proposes two-stage model

$$c(x,\xi) = \begin{cases} \min \quad c^\top y \\ \quad y \in X, \quad m \leqslant \sum_i y_i - \xi \leqslant M \\ \quad y \text{ coincides with } x \text{ on the } \tau \text{ first steps} \end{cases}$$

- The second-stage problem has the same form as the initial problem with a smaller horizon $T - \tau$

fine operational modeling vs difficult to compute

Model 2 : Two-stage stochastic unit-commitment 2/2

- Stochastic optimization problem

$$\boxed{\left\{\begin{array}{l} \min \quad c^\top x + \mathbb{E}[c(x,\xi)] \\ \quad x \in X, \quad m \leqslant \sum_i x_i - d \leqslant M \end{array}\right.}$$

- Complexity of $c(x,\xi)$ only allows for simple modeling of randomness
- VanAckooij Malick '15 uses finite distribution

$$\mathbb{P}(\xi = \xi_s) = p_s \quad (s = 1, \dots, S)$$

- (Convex) implicit second-stage function

$$v(x) = \mathbb{E}[c(x,\xi)] = \sum_{s=1}^{S} p_s c(x,\xi_s)$$

- For our numerical experiments (see later) $S = 50, 150, 250$

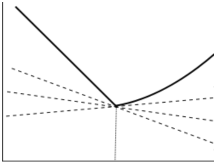
  problems with more than 1 million of variables and constraints...

Outline

Cutting plane models and algorithms

Cutting-plane models of implicit function

- The convex function v only known partially through an oracle : given an entry x, it returns $v(x)$ and $g \in \partial v(x)$



- From points $x_1, \ldots, x_k$, we can build the cutting plane model for v :

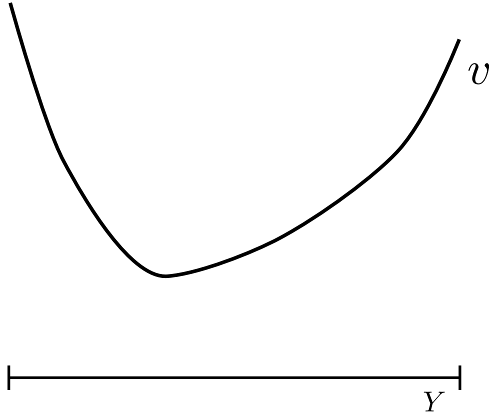
$$\check{v}_k(x) := \max_{j=1,\ldots,k} \{v(x_j) + {g_j}^\top (x - x_j)\}$$

- Convexity yields : $\check{v}_k(x) \leqslant v(x)$ for all x

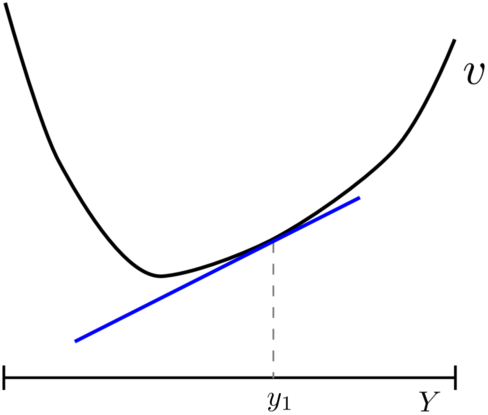
Cutting-plane algorithm (Kelley '60)

- Instead of $\min_{x \in X} v(x)$ we solve $\min_{x \in X} \check{v}_k(x)$ to get x_{k+1}
- When X is polyhedral this is a mere linear optimization problem
- Cultural note : cutting is a foundational technique of Operations Research in general

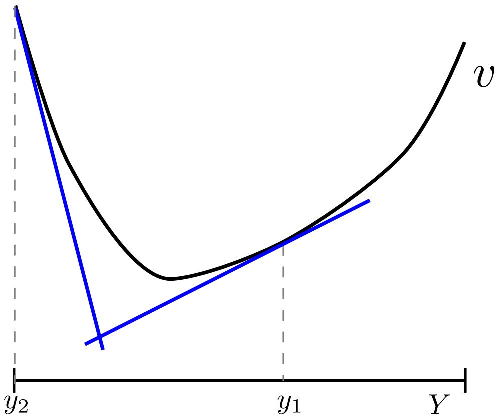
Cutting-plane algorithm on a picture



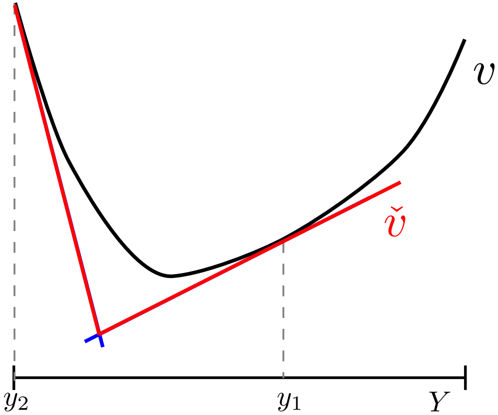
Cutting-plane algorithm on a picture



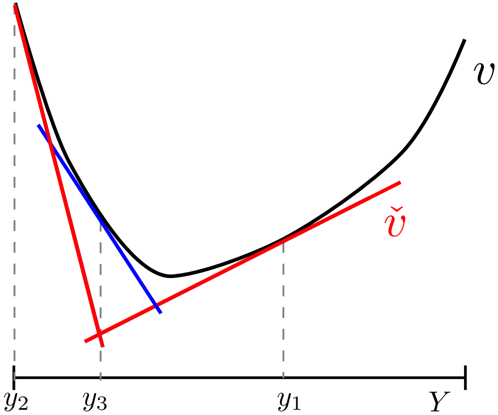
Cutting-plane algorithm on a picture



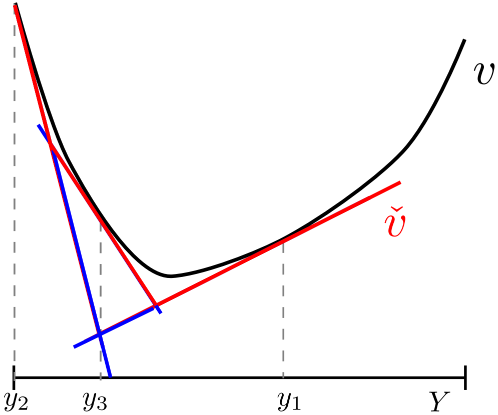
Cutting-plane algorithm on a picture



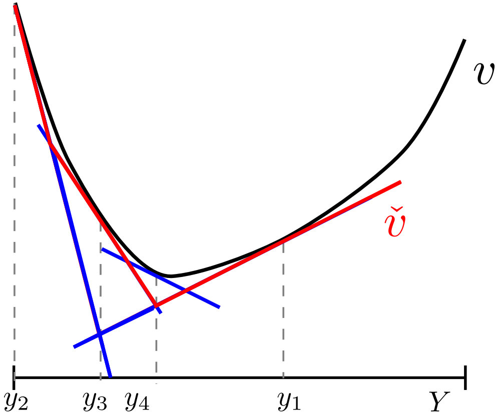
Cutting-plane algorithm on a picture



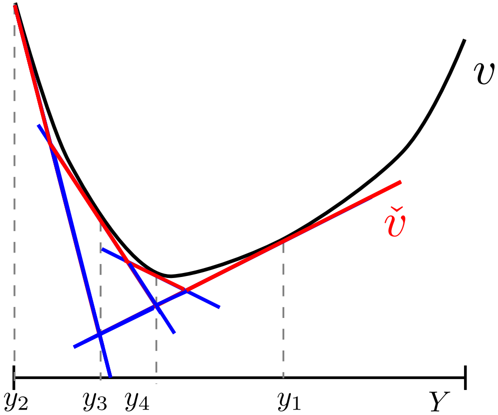
Cutting-plane algorithm on a picture



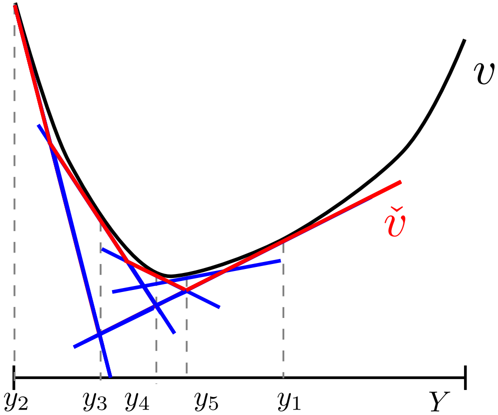
Cutting-plane algorithm on a picture



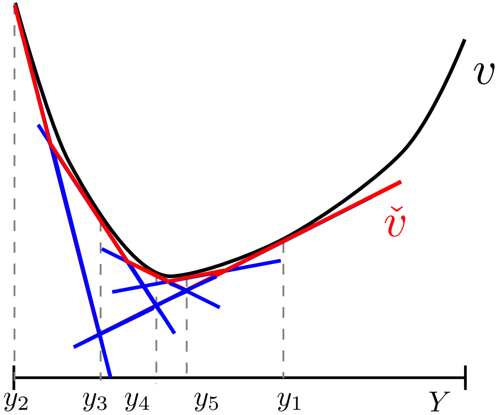
Cutting-plane algorithm on a picture



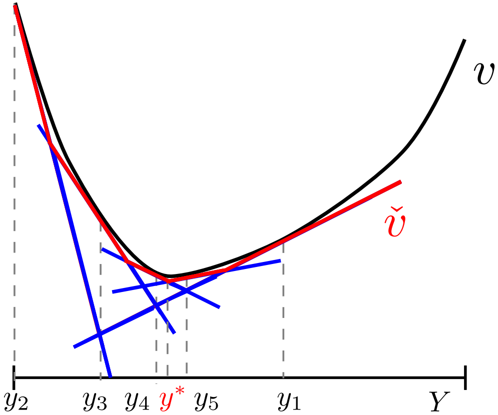
Cutting-plane algorithm on a picture



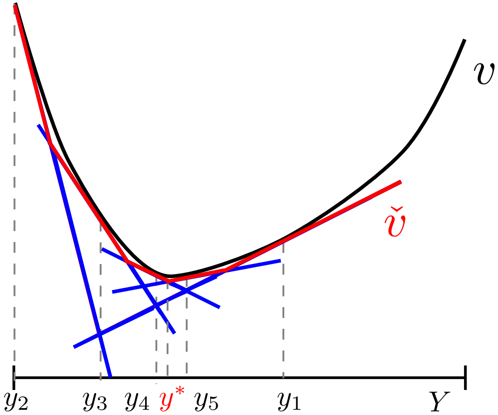
Cutting-plane algorithm on a picture



Cutting-plane algorithm on a picture



Cutting-plane algorithm on a picture



Convergence in 6 iterations

Cutting-plane for stochastic optimization : inexact oracles

In our stochastic optimization examples:

- we do have convex functions
  - example 1 : $v(x) = -\log \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}]$
  - example 2 : $v(x) = \max_{\xi \in \Xi} \ \sum_{t=1}^{T} \psi\big(\sum_i x_i^t - \xi^t\big)$
  - example 3 : $v(x) = \mathbb{E}[c(x, \xi)] = \sum_{s=1}^{S} p_s c(x, \xi_s)$
- we only approximate $v(x)$ and $g \in \partial v(x)$
  - by numerical integration for example 1
  - by maximization for example 2
  - by dual resolution for example 3

Cutting-plane for stochastic optimization : inexact oracles

In our stochastic optimization examples:

- we do have convex functions
  - example 1 : $v(x) = -\log \mathbb{P}[V_{\min} \leqslant V_0 + Ax + \xi \leqslant V_{\max}]$
  - example 2 : $v(x) = \max_{\xi \in \Xi} \ \sum_{t=1}^{T} \psi\big(\sum_i x_i^t - \xi^t \big)$
  - example 3 : $v(x) = \mathbb{E}[c(x, \xi)] = \sum_{s=1}^{S} p_s c(x, \xi_s)$
- we only approximate $v(x)$ and $g \in \partial v(x)$
  - by numerical integration for example 1
  - by maximization for example 2
  - by dual resolution for example 3

Cutting-plane algorithms easily extend to inexact oracles

Inexact oracle provides approximate cutting planes: for given x

$$
\begin{aligned}
v_x &= v(x) - \eta_x & & (\eta_{\max} \geqslant \eta_x \geqslant \eta_{\min}) \\
v(y) &\geqslant v_x + {g_x}^\top (y - x) - \varepsilon_x & & (\varepsilon_{\max} \geqslant \varepsilon_x \geqslant 0),
\end{aligned}
$$

When $\eta^x = \varepsilon^x = 0$, we retrieve $g_x \in \partial v(x)$

Numerical illustration for stochastic unit-commitment

- With real-life EDF model (data from 2013)
  - deterministic problem (1 scenario) : around 50000 continuous variables, 27000 binary variables, and 815000 constraints
  - stochastic problem (50 scenarios) : 1,200,000 continuous variables, 700,000 binary variables, and 20,000,000 constraints

$\rightarrow$ Out of reach for existing (mixed-integer linear) solvers !

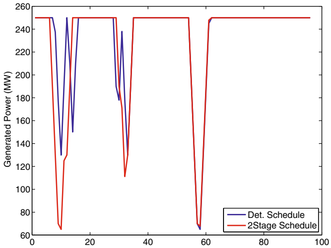
Numerical illustration for stochastic unit-commitment

- With real-life EDF model (data from 2013)
  - deterministic problem (1 scenario) : around 50000 continuous variables, 27000 binary variables, and 815000 constraints
  - stochastic problem (50 scenarios) : 1,200,000 continuous variables, 700,000 binary variables, and 20,000,000 constraints

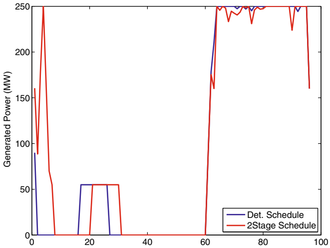
  $\rightarrow$ Out of reach for existing (mixed-integer linear) solvers !

- Cutting-plane allows to solve it – in reasonable time ☺

- Observation : generation is transferred from inflexible but cheap units to expensive but flexible units



inflexible and cheap



expensive and flexible

Further note on numerical experiments

- Emphasis on an interest of inexact cutting-plane : hot-restart
- When increasing the number of scenario:

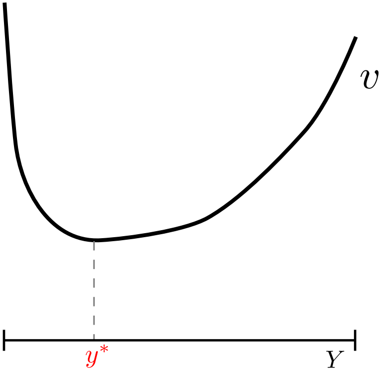
Number of Scenarios	Nb. iter	Sub-pbs calls 1st Stage	Sub-pbs calls 2nd Stage	Average calls
50	4	167	1009	5.88
100	8	360	3461	4.77
250	16	694	14205	3.73

- Hot-restart yields decrease of sub-pbs calls per scenario and iteration
- The number of sub-pbs calls remains within reasonable limits.
  For comparison, using up to 300 calls is common for deterministic real-life unit-commitment
- See more in vanAckooij Malick '15

Outline

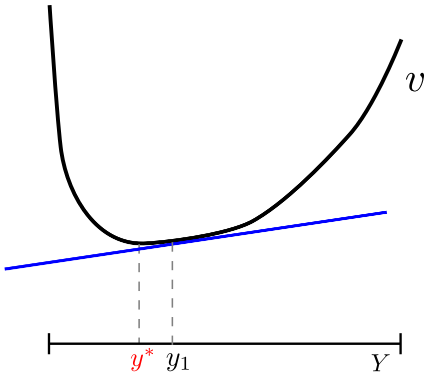
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



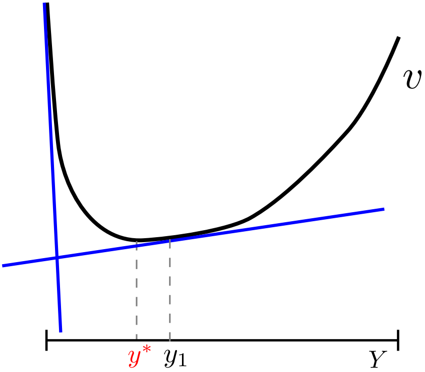
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



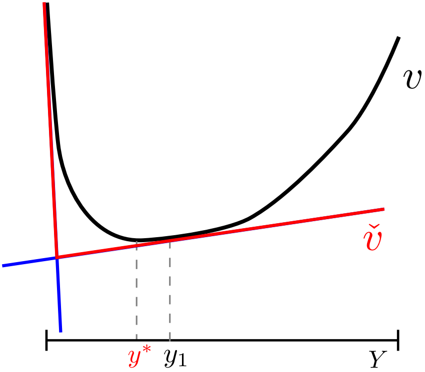
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



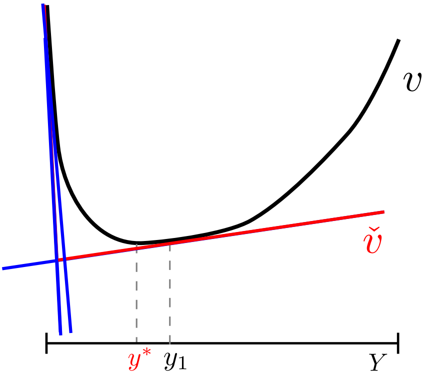
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



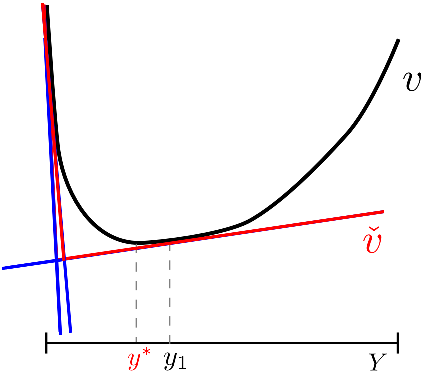
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



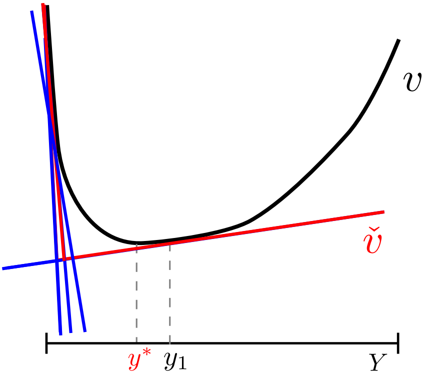
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



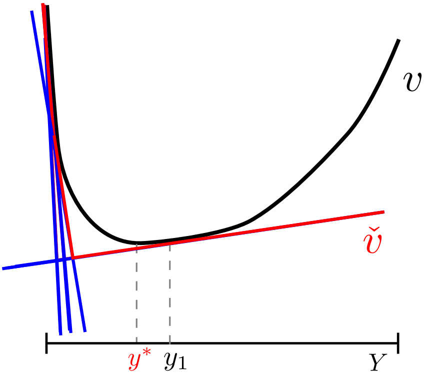
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



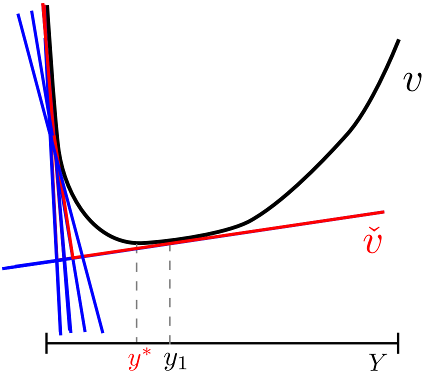
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



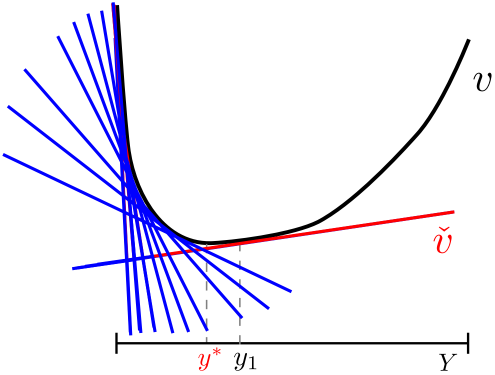
Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



Instability of cutting-plane

- The cutting-plane algorithm is simple... but has several drawbacks
- Among them : inherent instability, see the picture...
- E.g. it is the case for robust unit-commitment ☹
  (vs stochastic unit-commitment ☺)



Stabilized cutting-plane: (level) bundle method

Simple idea : add a quadratic term stabilizing the cutting-plane model around the 'best' current point, called stability center $\hat{x}_k$

- prox-bundle methods Lemaréchal 70s-80s
  (interpreted as inexact proximal algorithm)
- level-bundle methods Lemaréchal Nesterov Nemirovski '95
- many recent improvements and generalizations (around inexactness)

Stabilized cutting-plane: (level) bundle method

Simple idea : add a quadratic term stabilizing the cutting-plane model around the 'best' current point, called stability center $\hat{x}_k$

- prox-bundle methods Lemaréchal 70s-80s
  (interpreted as inexact proximal algorithm)
- level-bundle methods Lemaréchal Nesterov Nemirovski '95
- many recent improvements and generalizations (around inexactness)

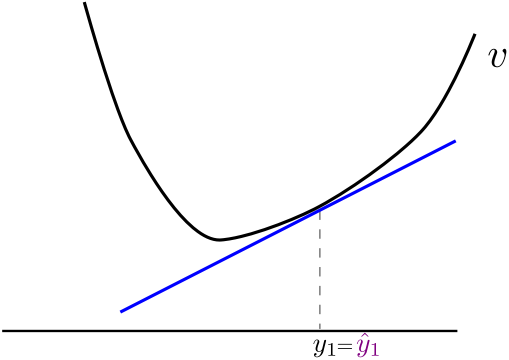
An iteration of level bundle algorithm

$$x_{k+1} := \operatorname{argmin} \left\{ \frac{1}{2}\|x - \hat{x}_k\|^2 : \ \check{v}_k(x) \leqslant L_k, \ x \in X \right\}$$

the level $L_k := (v_k^{\mathtt{up}} + v_k^{\mathtt{low}})/2$ middle of $v_k^{\mathtt{low}}$ lower bound and $v_k^{\mathtt{up}} := \min_{1 \leqslant j \leqslant k} v(x_j)$ upper bound for the minimum of v on X.

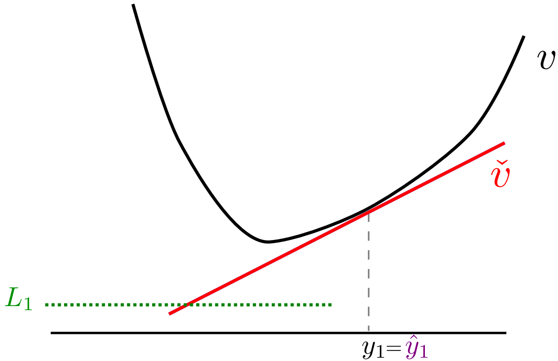
Level bundle algorithm on a picture

Same 1-dimension example :



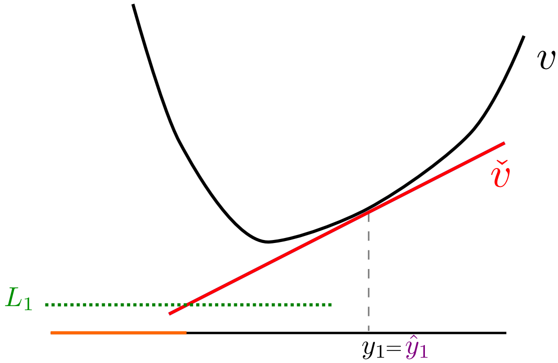
Level bundle algorithm on a picture

Same 1-dimension example :



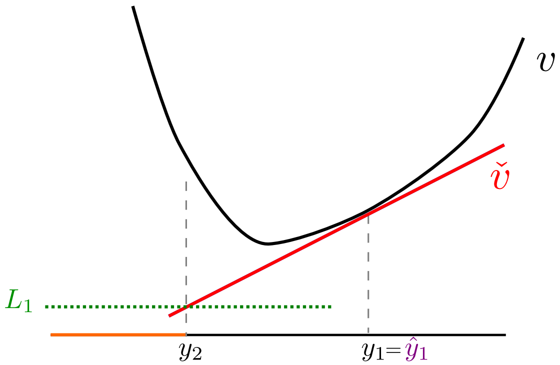
Level bundle algorithm on a picture

Same 1-dimension example :



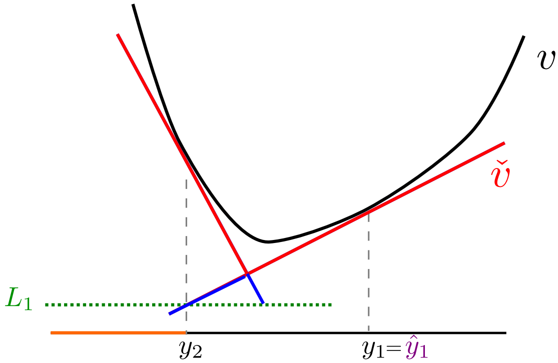
Level bundle algorithm on a picture

Same 1-dimension example :



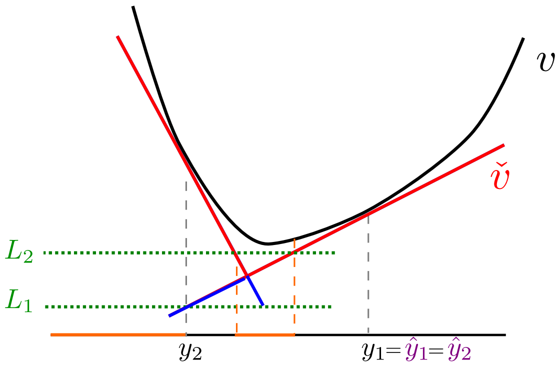
Level bundle algorithm on a picture

Same 1-dimension example :



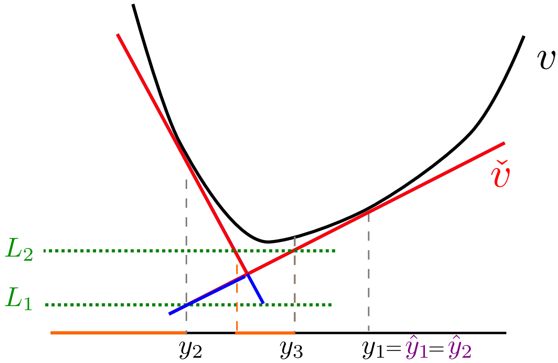
Level bundle algorithm on a picture

Same 1-dimension example :



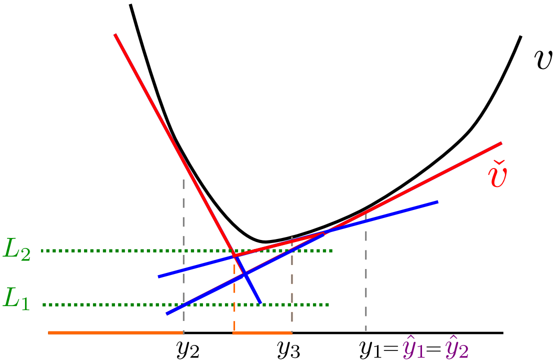
Level bundle algorithm on a picture

Same 1-dimension example :



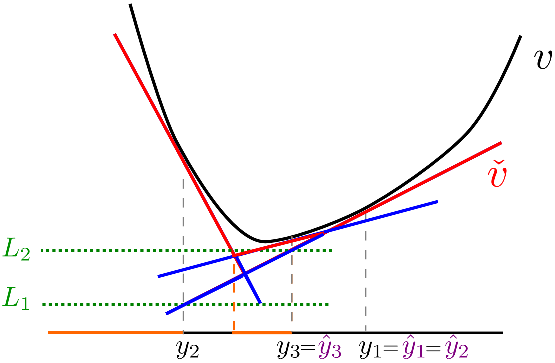
Level bundle algorithm on a picture

Same 1-dimension example :



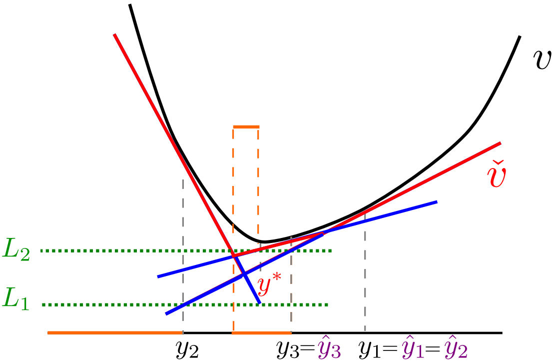
Level bundle algorithm on a picture

Same 1-dimension example :



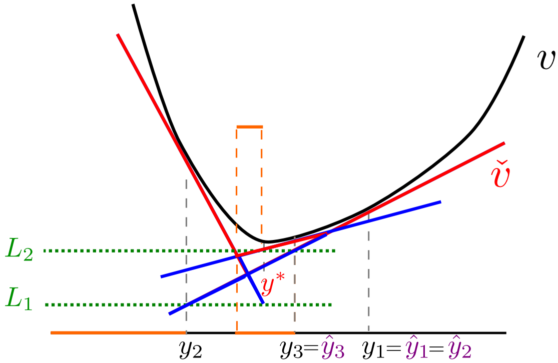
Level bundle algorithm on a picture

Same 1-dimension example :



Level bundle algorithm on a picture

Same 1-dimension example :



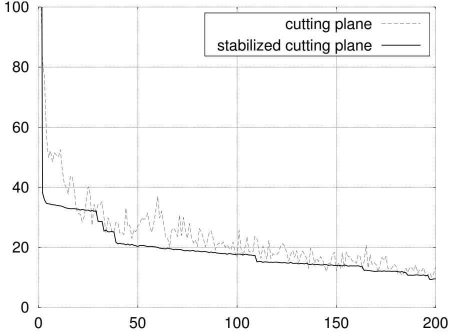
convergence in only 4 iterations (vs 6 for cutting-plane)

Numerical illustration for robust unit-commitment : plot

Using quadratic stabilization does stabilize !

See e.g. on one run for robust unit-commitment on the EDF data set

$$\begin{cases} \min \quad c^\top x + \max_{\xi \in \Xi} \sum_{t=1}^{T} \psi\big(\sum_i x_i^t - \xi^t\big) \\ \quad x \in X \end{cases}$$



cutting-plane alg. vs bundle alg. (relative gap / number of iterations)

Numerical illustration for robust unit-commitment : table

More numerical results on the EDF data set

- comparison of iterations (#it) and sub-pbs calls (#calls)
- cutting-plane algorithm vs bundle algorithm
- easy stopping criteria : 5%
  (the comparison is even more unbalanced for 1%)

Instance	cutting-plane		bundle		improvement	
date	# it.	calls	# it.	calls	# it.	calls
15/01/2013	41	536	26	317	158%	169%
20/03/2013	393	613	103	245	382%	250%
13/05/2013	764	2315	319	974	239%	238%
22/08/2013	2648	3452	356	790	744%	347%
25/10/2013	724	2163	209	1147	346%	189%
10/12/2013	78	578	34	229	229%	252%

Outline

Take-home message

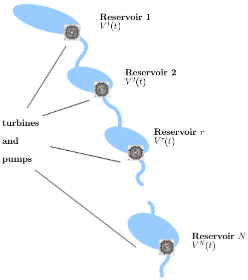
1. Nonsmoothness often appear (e.g. in case max-functions)
2. In industrial energy applications, convex functions are often known implicitly through inexact oracles (vs machine learning)
3. (Stabilized) cutting-plane algorithms are methods of choice
   for large-scale applications showing decomposability

Take-home message

1. Nonsmoothness often appear (e.g. in case max-functions)
2. In industrial energy applications, convex functions are often known implicitly through inexact oracles (vs machine learning)
3. (Stabilized) cutting-plane algorithms are methods of choice for large-scale applications showing decomposability
4. There is active recent research on such methods (inexact computation, faster convergence, new applications...)
5. These nonsmooth algorithms also method of choice for (hard) smooth problems !? e.g. probabilistic constrained optimization

Back to hydro-reservoir

$$\begin{cases} \max & \pi^\top x \\ & \mathbb{P}[V^i_{\text{min}} \leqslant V^i_0 + Ax + \xi \leqslant V^i_{\text{max}}] \geqslant 0.8 \end{cases}$$



When ξ gaussian:

- it is a smooth convex problems
- but the best solution methods are nonsmooth optimization methods !

Comparison of algorithms on the Isère Valley (real-life EDF data)

method	obj. value	$\mathbb{P}$	Nb. Iter. [InfeasQP]	CPU time (mins)
Prepoka '03	175222	0.799511	190	1504.7
Kiwiel '08	175237	0.799394	204	627.3
VanA. Sag. '15	175237	0.799418	188	573.5
LNN '95	175235	0.799854	161	529.6
level	175235	0.799604	165 [3]	423.2

Some references

Material mainly extracted from :

W. van Ackooij and J. Malick
Decomposition algorithm for large-scale two-stage unit-commitment
Annals of Operations Research, 238(1):587-613, 2016

J. Malick, S. Zaourar, and W. Oliveira
Uncontrolled inexact information within bundle methods
to appear in *EURO Journal on Computational Optimization*, 2016

W. van Ackooij, N. Lebbe, and J. Malick
Regularized decomposition of large-scale block-structured robust optimization problems
Submitted to *Computational Management Science*, 2016

W. van Ackooij and J. Malick
Eventual convexity of feasible sets for nonlinear probabilistic constraints with elliptically symmetric distributions
to be submitted, 2016

and also :

W. van Ackooij, R. Henrion, A. Moller, and R. Zorgati
Joint chance constrained programming for hydro reservoir management
Optimization and Engineering, 15:509–531, 2014

Some references

Material mainly extracted from :

W. van Ackooij and J. Malick
Decomposition algorithm for large-scale two-stage unit-commitment
Annals of Operations Research, 238(1):587-613, 2016

J. Malick, S. Zaourar, and W. Oliveira
Uncontrolled inexact information within bundle methods
to appear in *EURO Journal on Computational Optimization*, 2016

W. van Ackooij, N. Lebbe, and J. Malick
Regularized decomposition of large-scale block-structured robust optimization problems
Submitted to *Computational Management Science*, 2016

W. van Ackooij and J. Malick
Eventual convexity of feasible sets for nonlinear probabilistic constraints with elliptically symmetric distributions
to be submitted, 2016

and also :

W. van Ackooij, R. Henrion, A. Moller, and R. Zorgati
Joint chance constrained programming for hydro reservoir management
Optimization and Engineering, 15:509–531, 2014

Some references

Material mainly extracted from :

W. van Ackooij and J. Malick
Decomposition algorithm for large-scale two-stage unit-commitment
Annals of Operations Research, 238(1):587-613, 2016

J. Malick, S. Zaourar, and W. Oliveira
Uncontrolled inexact information within bundle methods
to appear in *EURO Journal on Computational Optimization*, 2016

W. van Ackooij, N. Lebbe, and J. Malick
Regularized decomposition of large-scale block-structured robust optimization problems
Submitted to *Computational Management Science*, 2016

W. van Ackooij and J. Malick
Eventual convexity of feasible sets for nonlinear probabilistic constraints with elliptically symmetric distributions
to be submitted, 2016

and also :

W. van Ackooij, R. Henrion, A. Moller, and R. Zorgati
Joint chance constrained programming for hydro reservoir management
Optimization and Engineering, 15:509–531, 2014

Supplementary

Outline

Counter-example of well-behaviour

Let $\xi \sim N(0,1)$ be given and consider

$$\phi(x) := \mathbb{P}[Qx + L\xi \geqslant b],$$

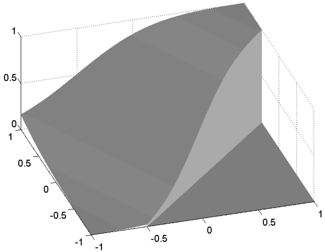
with

$$Q = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}, L = \begin{bmatrix} -1 \\ 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ -\frac{1}{2} \end{bmatrix}$$

Then ϕ is not continuous !

because of the presence of "deterministic" constraints $-1x_1 + x_2 \geqslant -\frac{1}{2}$

the set $\{z \in \mathbb{R}^m \ : g(x,z) = 0\}$ is not of zero measure (at some x's).



Evaluating $\mathbb{P}$

- Let $\xi \sim N(0, R)$ and $R = LL^\top$.
- $\xi = \eta L\zeta$, where η has a chi-distribution with m degrees of freedom and ζ is uniformly distributed over $\mathbb{S}^{m-1}$ euclidian unit-sphere of $\mathbb{R}^m$
- As a consequence, for M Lebesgue-mesurable

$$\mathbb{P}[\xi \in M] = \int_{v \in \mathbb{S}^{m-1}} \mu_\eta \left(\{r \geqslant 0 : rLv \cap M \neq \emptyset\}\right) d\mu_\zeta$$

- Efficient sampling schemes for such integrals Deak 00
- In our case $M(x) = \{z \in \mathbb{R}^m \ : g(x, z) \leqslant 0\}$ is a convex hence Lebesgue measurable

Eventual convexity example

In general $g \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is defined by:

$$g(x,z) := z^\top W(x) z + 2\sum_{i=1}^{n} x_i {w_i}^\top z + b,$$

where W a positive semi-definite matrix valued mapping

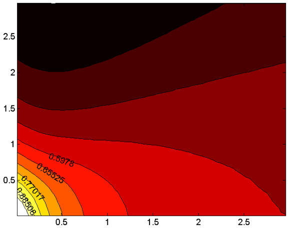
$W(x) = x_1 W_1 + x_2 W_2$, where

$$W_1 = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix} \text{ and } W_2 = \begin{pmatrix} 1 & -0.7 \\ -0.7 & 1 \end{pmatrix}$$

the correlation matrix R:

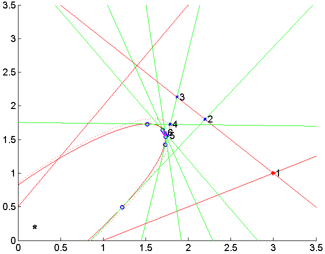
$$R = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}.$$

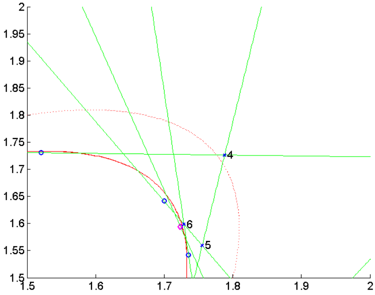
and $w_1 = (-1, 1)$, $w_2 = (2, 3)$ and $b = -3$



Proba constraints are not just plain non-linear constraints

- φ is not known up to arbitrary precision (or would be unreasonably costly). A (sub-)gradient of φ also suffers from numerical imprecision. Explicit formula allowing for computations with a trade-off cost/efficiency.
- An example shows that cutting planes may locally over-estimate the map (or set):





Eventual convexity result

Theorem (Simplified version)

Let $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ convex in the first argument and "non-distorted". Let $\xi \sim E(\mu, \Sigma, \theta)$ be elliptically symmetrically distributed with mean μ, covariance matrix Σ, generator θ and associated radial distribution R. Assume also a "generalized concavity" property of F_R.

Then the set

$$M(p) = \{x \in \mathbb{R}^n \; : \mathbb{P}[g(x,\xi) \leqslant 0] \geqslant p\}$$

is convex provided that

$$p \geqslant p^* := \frac{1}{4} F_R\left(\frac{z_*(m,\alpha,\theta)\delta^{\mathtt{nd}}}{\delta^{\mathtt{vol}}}\right) + \frac{3}{4}$$

If g is only non-distorted for all $\bar{x} \in X$, where X is a convex compact set, then $M(p) \cap X$ is convex for $p \geqslant p^*$, with p^*

Outline

Solution by duality for deterministic case

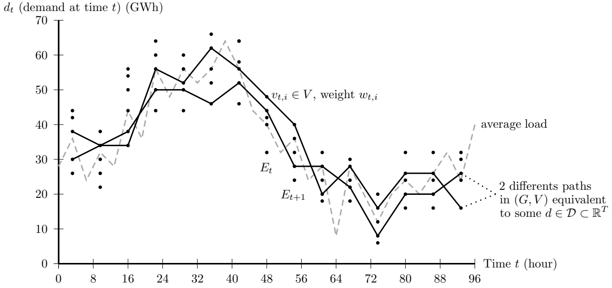
For special case $m = M$

$$\begin{cases} \min & c(x) \\ & x \in X, \quad \sum_i x_i - d = 0 \end{cases}$$

$$\begin{aligned} L(x,\lambda) &:= \sum_i c_i(x_i) + \sum_t \lambda^t \Big(\sum_i x_i^t - d^t \Big) = \sum_i \Big(c_i(x_i) + \sum_t \lambda^t (x_i^t - d^t) \Big) \\ \theta(\lambda) &:= \min_{x \in X} L(x,\lambda) = \sum_i \min_{x_i \in X_i} \Big(c_i(x_i) + \sum_t \lambda^t (x_i^t - d^t) \Big) \end{aligned}$$

dual resolution by proximal bundle method
prima recovery by efficient heuristics

State-space model



- At each time step $t \in \tau$, a set of nodes E_t containing both a value for D_t and a weight w_t.
- Graph (G, V), with $G = \bigcup_t E_t$, such that V connects all nodes in E_t to those in E_{t+1} that are not further apart than H,
- Ξ is the set of all paths in (G, V) satisfying $\sum_t w_t \leqslant W_{\mathtt{max}}$ for a given maximum budget of uncertainty $W_{\mathtt{max}}$.
- Card Ξ is huge, but computing the sup over Ξ by a simple 1-dimensional dynamic programming principle.

Load uncertainty

We generate the uncertain loads as an average load on top of which we add a causal time series model (see, e.g., Bruhns et al 2005).

We consider the Gaussian random variable

$$D(\xi) = \bar{D} + \zeta,$$

where ζ is an $AR(3)$ model with Gaussian innovations

The covariance matrix is $\Sigma^D = (C^D S)(S(C^D))^\top$, where S is a diagonal matrix with element

$$S_{ii} = \mathtt{f}\,\frac{\bar{D}_i}{\frac{1}{T}\sum_{j=1}^{T} \bar{D}_j}$$

and C^D is the nominal covariance matrix with $C_{ji} = \sum_{k=1}^{j-i+1} \phi_k^D$

$$\phi_k^D = \begin{cases} \phi_3 & \text{if } k = 1 \\ \phi_3\phi_1^D + \phi_2 & \text{if } k = 2 \\ \phi_3\phi_2^D + \phi_2\phi_1^D + \phi_1 & \text{if } k = 3 \\ \sum_{k=1}^{3} \phi_{4-k}\phi_{j-k}^D & \text{otherwise.} \end{cases}$$