Revisitant les méthodes dérivées de la décomposition d'opérateurs

Claudia Sagastizábal
IMECC-UNICAMP Brazil
CeMEAI
CEPID - Center for Mathematical Sciences Applied to Industry

SMAI 2023, Pointe à Pître
23 Mai 2023

Motivation: energy sources in Brazil

Motivation: what is the value of water for the energy business?

Decision today

Consequences in the future

•Minimize immediate cost by emptying reservoirs	rain	Decision ok
	drought	Deficit

Motivation: what is the value of water for the energy business?

Decision today | **Consequences in the future**

•Minimize immediate cost by emptying reservoirs	{	rain	Decision ok
		drought	Deficit

or

•Keep water, more $$$ by thermal generation	{	rain	Excess water
		drought	Decision ok

The value of water is an opportunity/substitution cost

Given by the value function of a linear stochastic program

Depends on

- the initial reservoir volumes
- the uncertainty representation
- how uncertainty is handled in the optimization problem
- how the optimization problem is solved
- environmental constraints

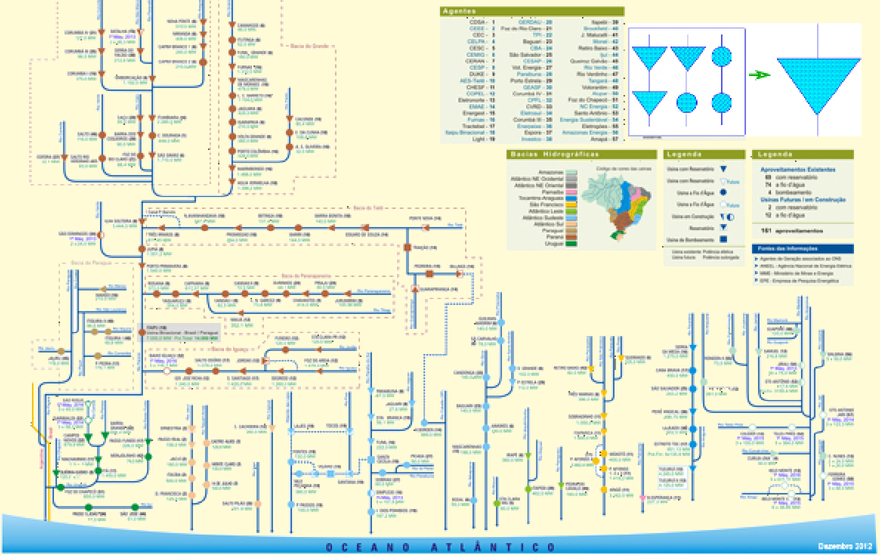
Drives guvernamental policies and business decisions of (+400) agents in the energy sector of Brazil

Diagrama Esquemático das Usinas Hidroelétricas do SIN

ONS Operador Nacional do Sistema Elétrico

Usinas Hidroelétricas Despachadas pelo ONS na Otimização da Operação Eletroenergética do Sistema Interligado Nacional

Horizonte: 2013 - 2017



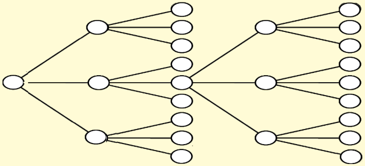
Future cost of water: piecewise linear function

$$v(x_0) = \begin{cases} \min\limits_{u,x} & \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{NS}\sum_{j=1}^{T_t^i} c_t^j g_t^{i,j}\Big] \\ & x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i & \textbf{(BAL)} \\ & gh_t^i + \sum_{j \le T_t^i} g_t^{i,j} + \sum_{\ell \in \mathcal{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \ge dem_t^i - (1-\gamma_t^i)\xi_t^i & \textbf{(DEM)} \\ & u^{\min} \le u = (gh, spill, gt, f) \le u^{\max} \\ & x_t^{i,\min} \le x_t^i \le x_t^{i,\max} & \textbf{(BOX)} \end{cases}$$

Future cost of water: piecewise linear function

$$v(x_0) = \begin{cases} \min\limits_{u,x} & \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{NS}\sum_{j=1}^{T_t^i} c_t^j g_t^{i,j}\Big] \\ & x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i \quad \textbf{(BAL)} \\ & gh_t^i + \sum_{j \le T_t^i} g_t^{i,j} + \sum_{\ell \in \mathscr{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \ge dem_t^i - (1-\gamma_t^i)\xi_t^i \quad \textbf{(DEM)} \\ & u^{\min} \le u = (gh, spill, gt, f) \le u^{\max} \\ & x_t^{i,\min} \le x_t^i \le x_t^{i,\max} \quad \textbf{(BOX)} \end{cases}$$

With 3 hydro conditions, $\{normal, wet, dry\}$ for $T = 4$ months, there are 3^3 scenarios

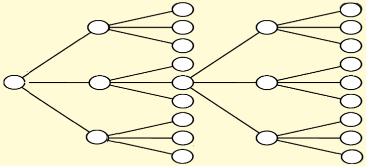


Future cost of water: piecewise linear function

$$v(x_0) = \begin{cases} \min\limits_{u,x} \quad \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{NS} \sum_{j=1}^{T_t^i} c_t^j g_t^{i,j} \right] \\ x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i & \textbf{(BAL)} \\ gh_t^i + \sum_{j \le T_t^i} g_t^{i,j} + \sum_{\ell \in \mathcal{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \ge dem_t^i - (1-\gamma_t^i)\xi_t^i & \textbf{(DEM)} \\ u^{\min} \le u = (gh, spill, gt, f) \le u^{\max} \\ x_t^{i,\min} \le x_t^i \le x_t^{i,\max} & \textbf{(BOX)} \end{cases}$$

With 3 hydro conditions, $\{normal, wet, dry\}$ for $T = 4$ months, there are 3^3 scenarios

real problem considers 10 years
($T = 120$ months)
it has 20^{119} scenarios!!!



Future cost of water: piecewise linear function

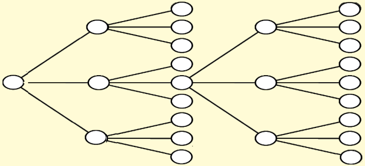
$$\left\{\begin{array}{ll}\min_{u,x} & \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{NS}\sum_{j=1}^{T_t^i} c_t^j g_t^{i,j}\Big] \\ & x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i \qquad \textbf{(BAL)} \\ & gh_t^i + \sum_{j \le T_t^i} g_t^{i,j} + \sum_{\ell \in \mathcal{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \ge dem_t^i - (1-\gamma_t^i)\xi_t^i \qquad \textbf{(DEM)} \\ & u^{\min} \le u = (gh, spill, gt, f) \le u^{\max} \\ & x_t^{i,\min} \le x_t^i \le x_t^{i,\max} \qquad \textbf{(BOX)}\end{array}\right.$$

With 3 hydro conditions, $\{normal, wet, dry\}$ for $T = 4$ months, there are 3^3 scenarios

real problem considers 10 years

($T = 120$ months)

it has 20^{119} scenarios!!!

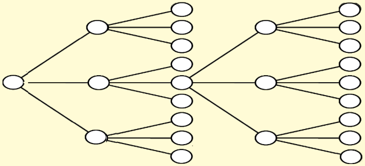


Future cost of water: piecewise linear function (parallel computation!)

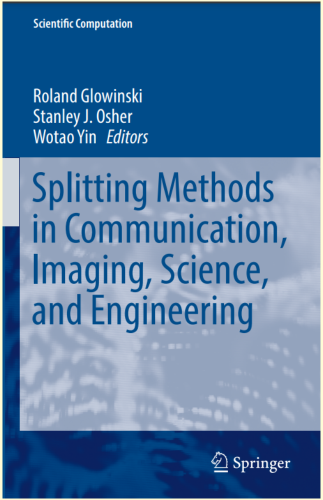
$$
\left\{
\begin{array}{ll}
\min_{u,x} & \mathbb{E}\Big[\sum_{t=1}^{T}\sum_{i=1}^{NS}\sum_{j=1}^{T_t^i} c_t^j g_t^{i,j}\Big] \\
 & x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i \quad \textbf{(BAL)} \\
 & gh_t^i + \sum_{j \le T_t^i} g_t^{i,j} + \sum_{\ell \in \mathscr{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \ge dem_t^i - (1-\gamma_t^i)\xi_t^i \quad \textbf{(DEM)} \\
 & u^{\min} \le u = (gh, spill, gt, f) \le u^{\max} \\
 & x_t^{i,\min} \le x_t^i \le x_t^{i,\max} \quad \textbf{(BOX)}
\end{array}
\right.
$$

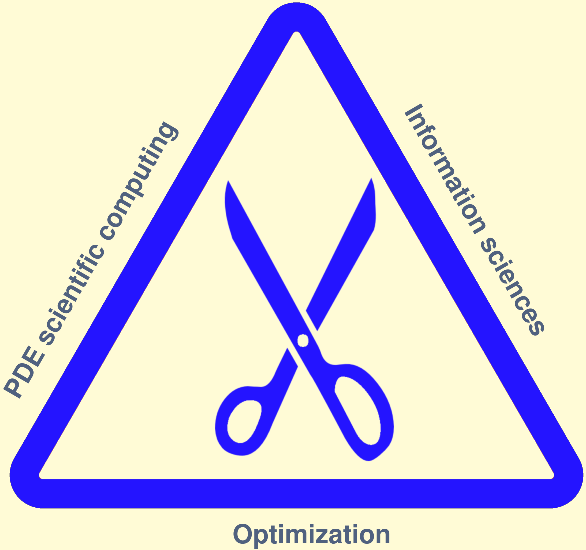
With 3 hydro conditions, $\{normal, wet, dry\}$ for $T = 4$ months, there are 3^3 scenarios

real problem considers 10 years
($T = 120$ months)
it has 20^{119} scenarios!!!

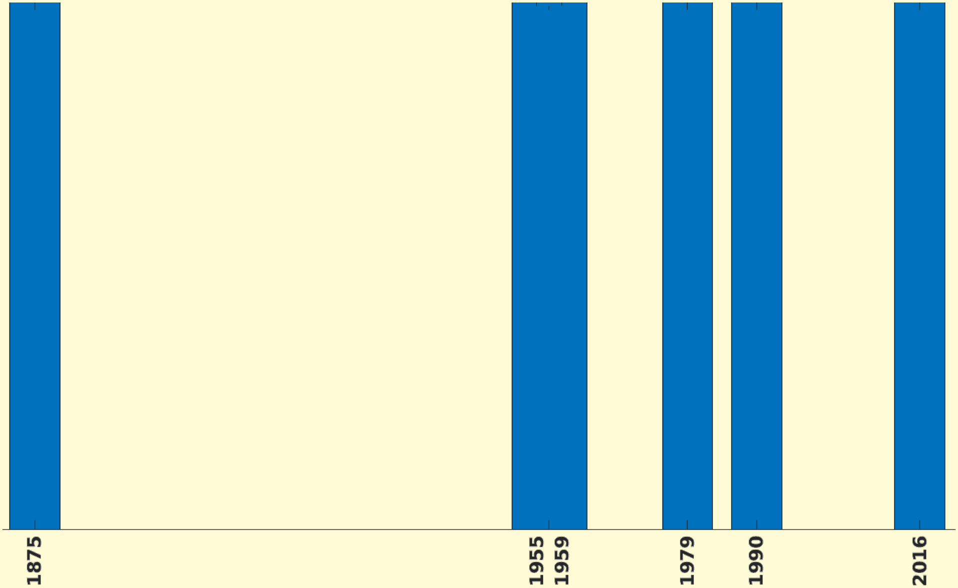


The Triangle of Splitting Methods

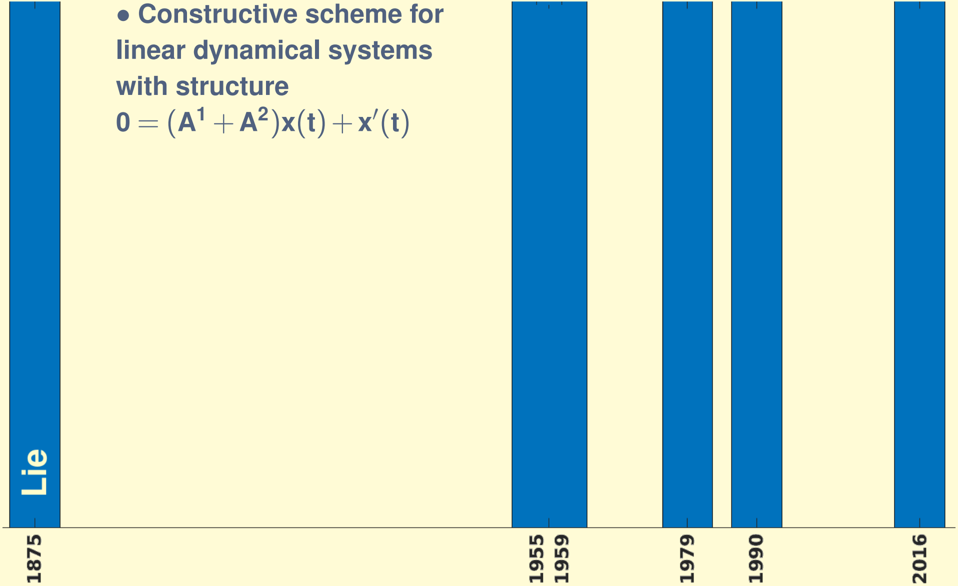




Splitting Timeline - some milestones



Splitting Timeline - some milestones

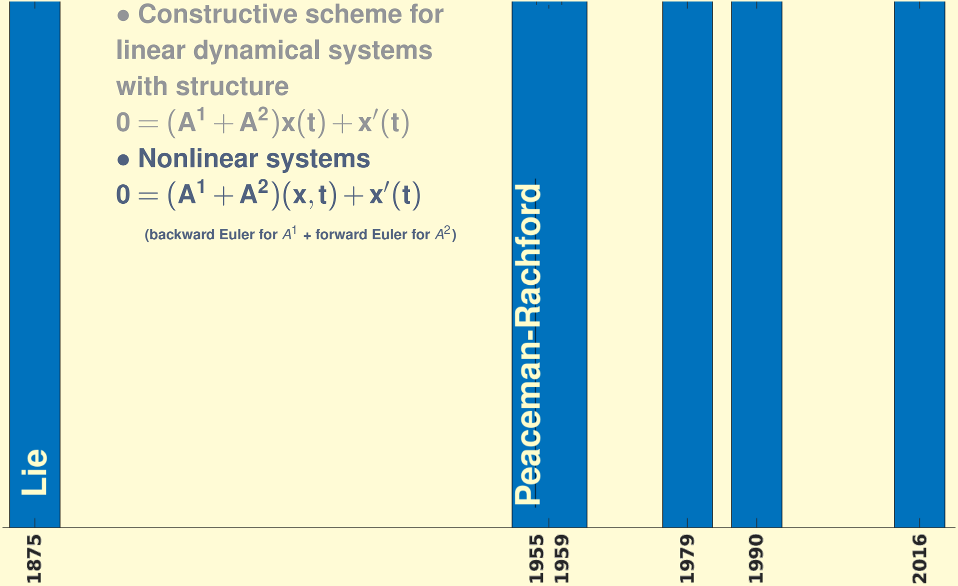


- Constructive scheme for linear dynamical systems with structure $0 = (A^1 + A^2)x(t) + x'(t)$

Splitting Timeline - some milestones

- Constructive scheme for linear dynamical systems with structure $0 = (A^1 + A^2)x(t) + x'(t)$
- **Nonlinear systems** $0 = (A^1 + A^2)(x, t) + x'(t)$

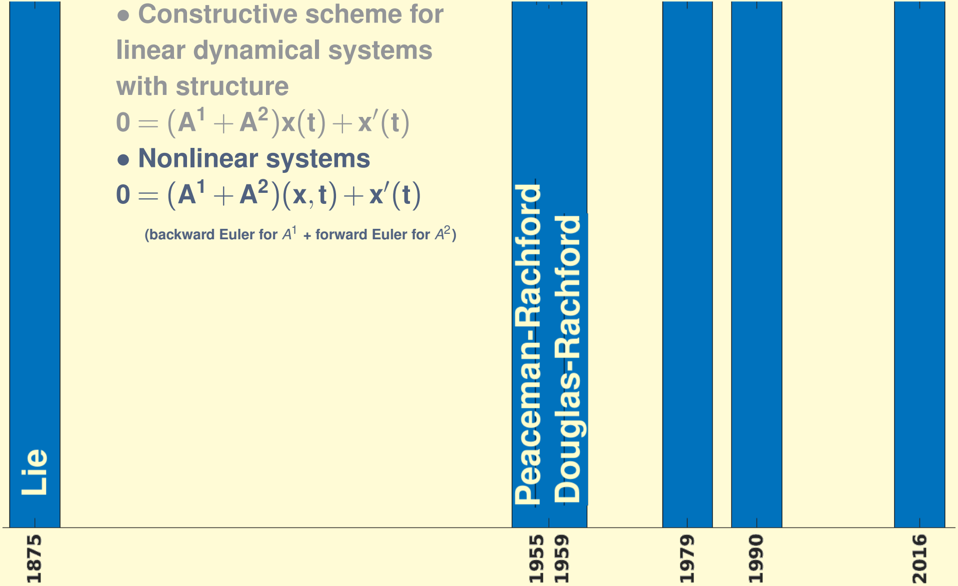
(backward Euler for A^1 + forward Euler for A^2)



Splitting Timeline - some milestones

- Constructive scheme for linear dynamical systems with structure $0 = (A^1 + A^2)x(t) + x'(t)$
- Nonlinear systems $0 = (A^1 + A^2)(x, t) + x'(t)$

(backward Euler for A^1 + forward Euler for A^2)

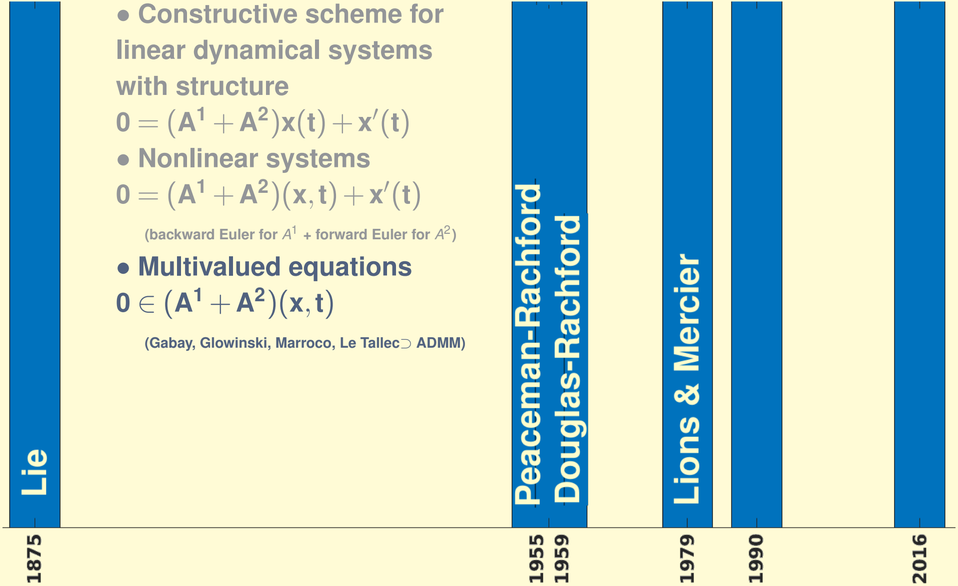


Splitting Timeline - some milestones

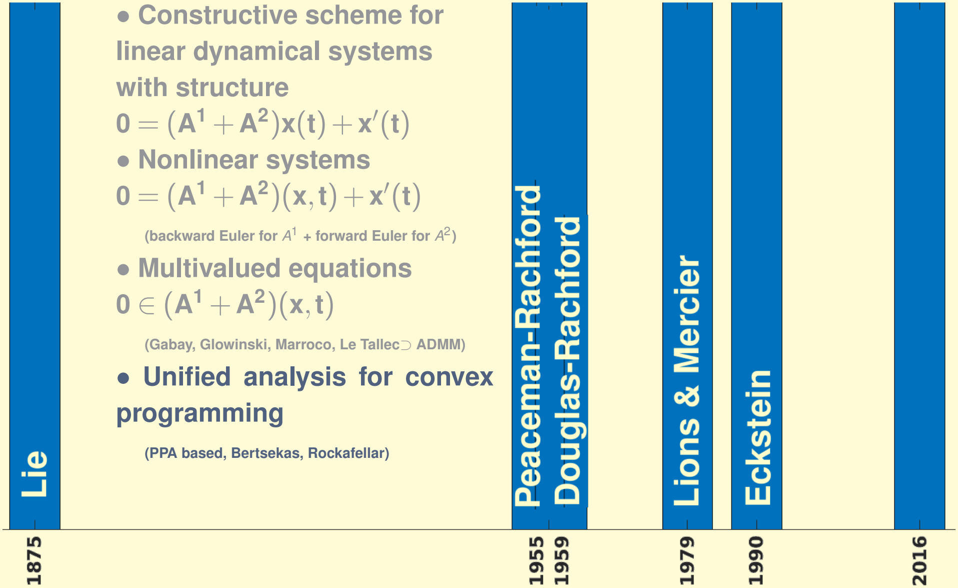
- Constructive scheme for linear dynamical systems with structure
  $0 = (A^1 + A^2)x(t) + x'(t)$
- Nonlinear systems
  $0 = (A^1 + A^2)(x, t) + x'(t)$

  (backward Euler for A^1 + forward Euler for A^2)
- **Multivalued equations**
  $0 \in (A^1 + A^2)(x, t)$

  (Gabay, Glowinski, Marroco, Le Tallec $\supset$ ADMM)



Splitting Timeline - some milestones



- Constructive scheme for linear dynamical systems with structure
  $0 = (A^1 + A^2)x(t) + x'(t)$
- Nonlinear systems
  $0 = (A^1 + A^2)(x, t) + x'(t)$

  (backward Euler for A^1 + forward Euler for A^2)
- Multivalued equations
  $0 \in (A^1 + A^2)(x, t)$

  (Gabay, Glowinski, Marroco, Le Tallec$\supset$ ADMM)
- **Unified analysis for convex programming**

  (PPA based, Bertsekas, Rockafellar)

Splitting Timeline - some milestones

- Constructive scheme for linear dynamical systems with structure
  $0 = (A^1 + A^2)x(t) + x'(t)$
- Nonlinear systems
  $0 = (A^1 + A^2)(x, t) + x'(t)$

  (backward Euler for A^1 + forward Euler for A^2)
- Multivalued equations
  $0 \in (A^1 + A^2)(x, t)$

  (Gabay, Glowinski, Marroco, Le Tallec$\supset$ ADMM)
- Unified analysis for convex programming

  (PPA based, Bertsekas, Rockafellar)
- **Plethora!**

Lie — 1875

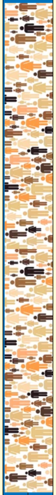
Peaceman-Rachford — 1955

Douglas-Rachford — 1959

Lions & Mercier — 1979

Eckstein — 1990

2016



Splitting cloud: dynamical systems, multivalued equations, distributed optimization, statistical learning

Splitting cloud: dynamical systems, multivalued equations, distributed optimization, statistical learning

$$0 \in A(x) \text{ for } A = A^1 + A^2$$

Splitting cloud: dynamical systems, multivalued equations, distributed optimization, statistical learning

$$0 \in A(x) \text{ for } A = A^1 + A^2$$

akin to Newton's method

$$0 = \nabla f(x)$$

Splitting cloud: dynamical systems, multivalued equations, distributed optimization, statistical learning

$0 \in A(x)$ for $A = A^1 + A^2$

akin to Newton's method

$$0 = \nabla f(x)$$

Can we exploit further knowing $A = \partial f$?

Un petit détour...

source: Sarah Dry's wordpress

Newton's method for nonlinear systems

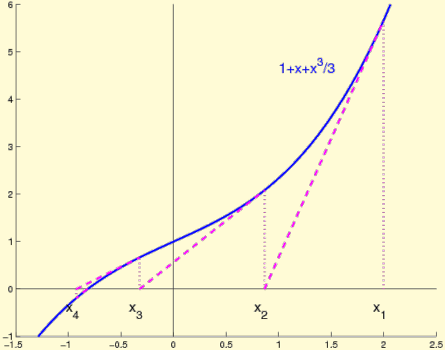
$$
\begin{aligned}
0 &= G(x^*) \\
&\approx G(x^k) + G'(x^k)d^k
\end{aligned}
$$

fast convergence $x^{k+1} = x^k + d^k$

Newton's method for nonlinear systems

$$
\begin{aligned}
0 &= G(x^*) \\
&\approx G(x^k) + G'(x^k) d^k
\end{aligned}
$$

fast convergence $x^{k+1} = x^k + d^k$



Newton method is accurate

$$G(x) = 1 + x + x^3/3$$

1	2.00000000000000	0
2	0.86666666666667	1
3	-0.32323745064862	1
4	-0.92578663808031	1
5	-0.82332584261905	2
6	-0.81774699537697	5
7	-0.81773167400186	9
8	-0.81773167388682	15

Newton

What about Newton's method for optimization?

$$\begin{aligned} 0 &= G(x^*) \\ &\approx G(x^k) + G'(x^k)d^k \end{aligned}$$

fast convergence

What about Newton's method for optimization?

$$
\begin{aligned}
0 &= G(x^*) \\
&\approx G(x^k) + G'(x^k)d^k
\end{aligned}
$$

fast convergence

In optimization

$G(x) = \nabla f(x)$ for an objective f

What about Newton's method for optimization?

$$
\begin{aligned}
0 &= \nabla f(x^*) \\
&\approx \nabla f(x^k) + \nabla^2 f(x^k) d^k
\end{aligned}
$$

fast convergence

In optimization

$$G(x) = \nabla f(x)$$

$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$$

for an objective f

What about Newton's method for optimization?

$$0 = \nabla f(x^*)$$
$$\approx \nabla f(x^k) + \nabla^2 f(x^k) d^k$$

fast convergence

In optimization

$$G(x) = \nabla f(x)$$

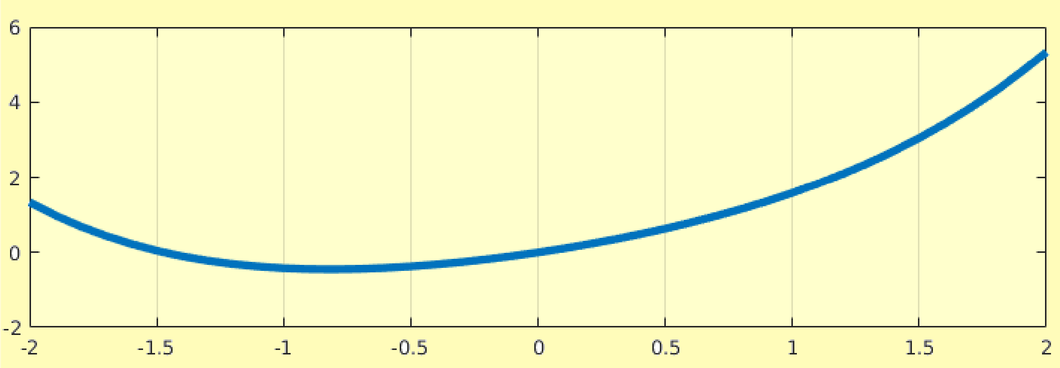
$$x^{k+1} = x^k - [\nabla^2 f(x^k)]^{-1} \nabla f(x^k)$$

for an objective f

$\min f \approx \min f$-model

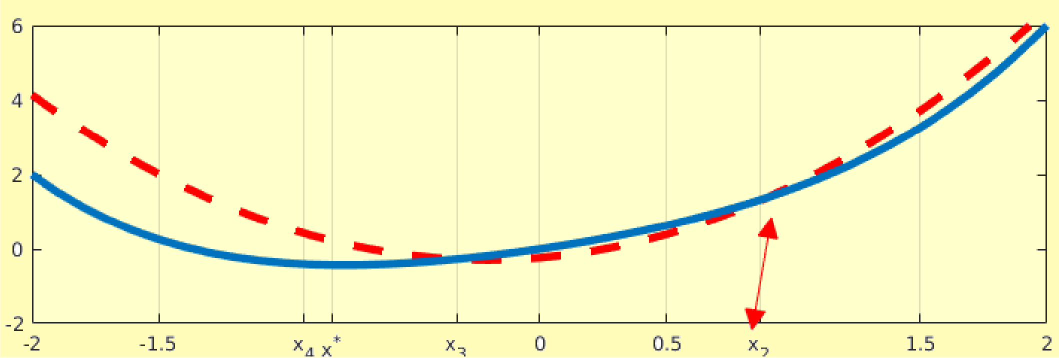
$$\min_d f(x^k) + \langle \nabla f(x^k), d \rangle + \frac{1}{2} \langle \nabla^2 f(x^k) d, d \rangle$$

Newton iterates for optimization

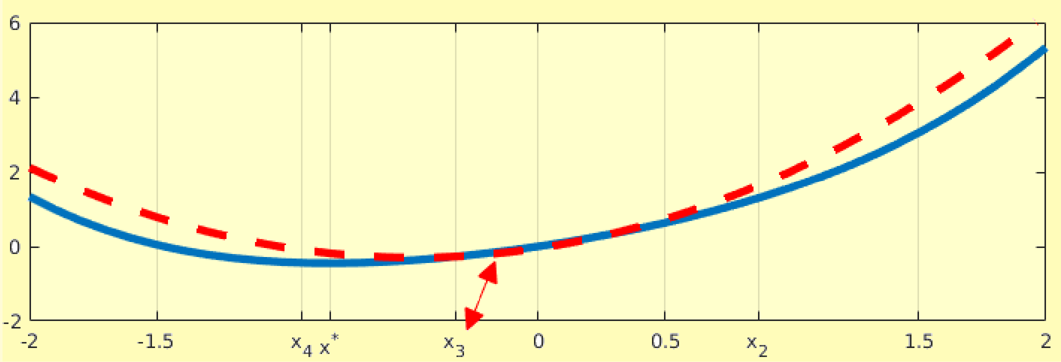


$$G(x) = 1 + x + x^3/3 \quad \Longrightarrow f(x) = x + x^2/2 + x^4/12$$

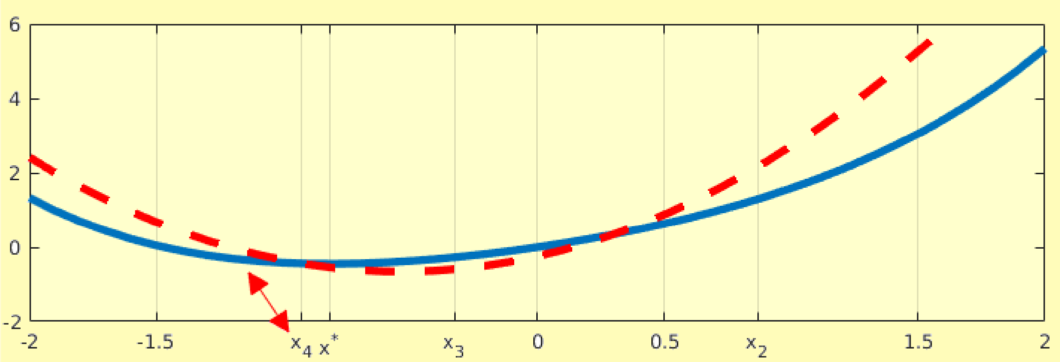
Newton iterates for optimization



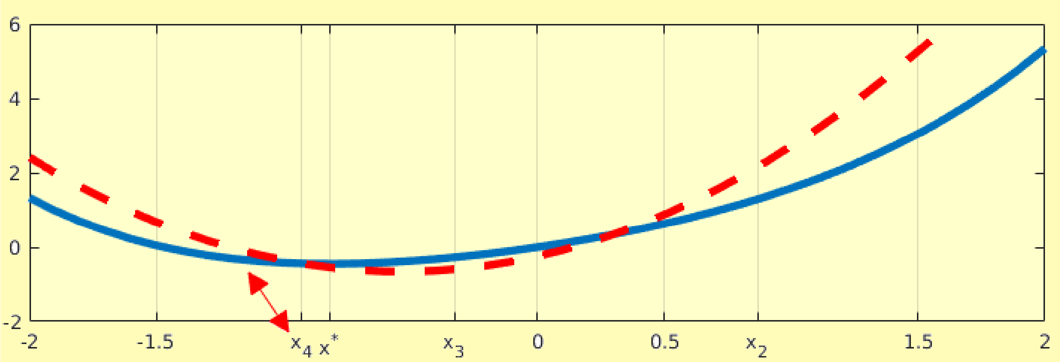
Newton iterates for optimization



Newton iterates for optimization

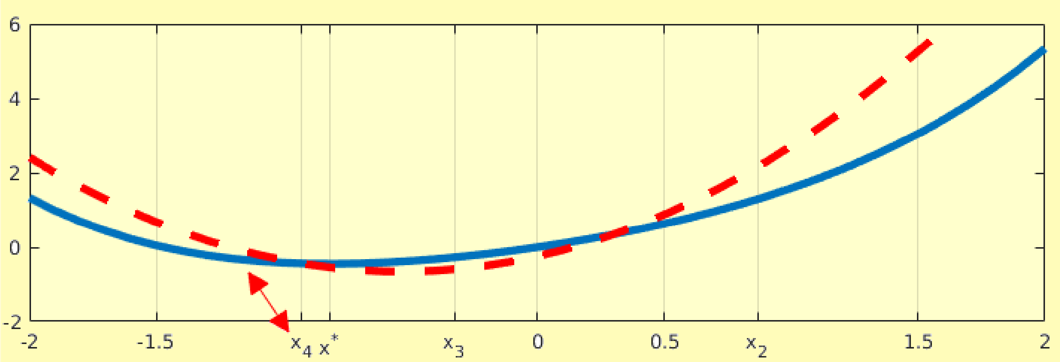


Newton iterates for optimization



Can we avoid computing the Hessian matrix?

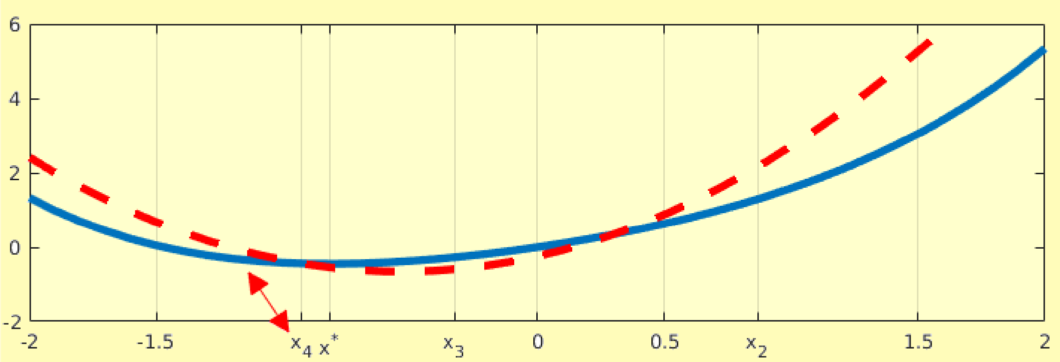
Newton iterates for optimization



Can we avoid computing the Hessian matrix? **YES!**

$$\min_d f(x^k) + \left\langle \nabla f(x^k), d \right\rangle + \frac{1}{2} \left\langle M^k d, d \right\rangle$$

Newton iterates for optimization

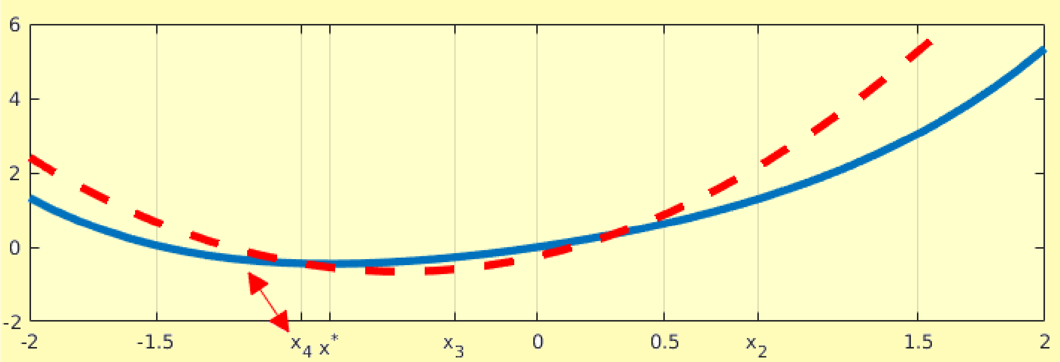


Can we avoid computing the Hessian matrix? **YES!**

$$\min_d f(x^k) + \left\langle \nabla f(x^k), d \right\rangle + \frac{1}{2} \left\langle M^k d, d \right\rangle$$

quasi-Newton matrix

Newton iterates for optimization

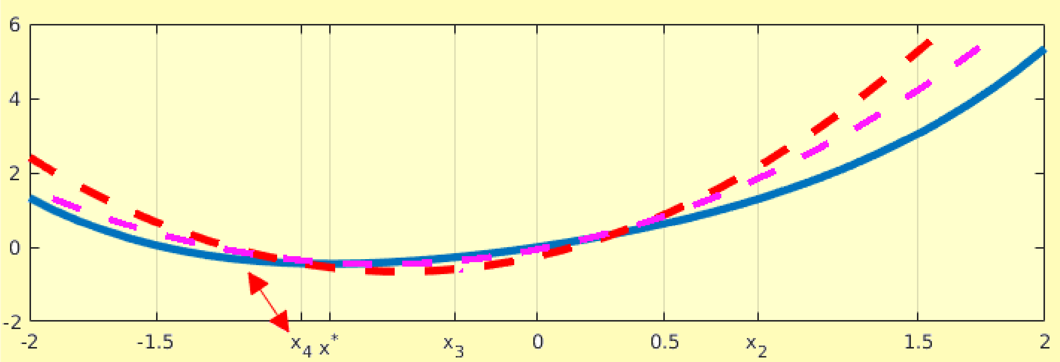


Can we avoid computing the Hessian matrix? **YES!**

$$\min_d f(x^k) + \langle \nabla f(x^k), d \rangle + \frac{1}{2} \langle M^k d, d \rangle$$

quasi-Newton matrix $0 = \nabla f(x^k) + M^k d^k$

quasi-Newton iterates for optimization



Eventually, the true Hessian curvature is estimated **only** along the generated directions

quasi-Newton methods are accurate too!

quasi-Newton

1	2.00000000000000	0
2	1.50000000000000	0
3	0.61224489795918	1
4	-0.16202797536640	1
5	-0.92209500449059	1
6	-0.78540447895661	1
7	-0.81609056319699	3
8	-0.81775774021392	5
9	-0.81773165292101	8
10	-0.81773167388656	13
11	-0.81773167388682	15

Newton

1	2.00000000000000	0
2	0.86666666666667	1
3	-0.32323745064862	1
4	-0.92578663808031	1
5	-0.82332584261905	2
6	-0.81774699537697	5
7	-0.81773167400186	9
8	-0.81773167388682	15

quasi-Newton methods are **accurate** too!

1	2.00000000000000	0
2	1.50000000000000	0
3	0.61224489795918	1
4	-0.16202797536640	1
5	-0.92209500449059	1
6	-0.78540447895661	1
7	-0.81609056319699	3
8	-0.81775774021392	5
9	-0.81773165292101	8
10	-0.81773167388656	13
11	-0.81773167388682	15

quasi-Newton

1	2.00000000000000	0
2	0.86666666666667	1
3	-0.32323745064862	1
4	-0.92578663808031	1
5	-0.82332584261905	2
6	-0.81774699537697	5
7	-0.81773167400186	9
8	-0.81773167388682	15

Newton

. . . fin du détour

Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

$\min_x f^1(x) + f^2(Mx)$

$$0 \in \partial f^1(x) + M^\top \partial f^2(Mx)$$

$$\begin{cases} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{cases}$$

Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

$\min_x f^1(x) + f^2(Mx)$ $\quad\quad$ $0 \in \partial f^1(x) + M^\top \partial f^2(Mx)$

$\begin{cases} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{cases}$

or its dual:

$\max_w -f^{1*}(-M^\top w) - f^{2*}(w)$ $\quad\quad$ $0 \in M\partial f^{1*}(-M^\top w) + \partial f^{2*}(w)$

$\begin{cases} k\text{th subproblem on } \partial f^{1*} \\ k\text{th subproblem on } \partial f^{2*} \end{cases}$

Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

$\min_x f^1(x) + f^2(Mx)$ $\qquad 0 \in \partial f^1(x) + M^\top \partial f^2(Mx)$

$$\begin{cases} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{cases}$$

or its dual:

$\max_w -f^{1*}(-M^\top w) - f^{2*}(w)$ $\qquad 0 \in M\partial f^{1*}(-M^\top w) + \partial f^{2*}(w)$

$$\begin{cases} k\text{th subproblem on } \partial f^{1*} \\ k\text{th subproblem on } \partial f^{2*} \end{cases}$$

Considering $x = (x^1, x^2)$

$$\min_{x^1, x^2} f^1(x^1) + f^2(x^2) \quad \text{s.t.} \quad A^1 x^1 + A^2 x^2 = 0$$

Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

$\min_x f^1(x) + f^2(Mx)$ $\qquad$ $0 \in \partial f^1(x) + M^\top \partial f^2(Mx)$

$$\begin{cases} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{cases}$$

or its dual:

$\max_w -{f^1}^*(-M^\top w) - {f^2}^*(w)$ $\qquad$ $0 \in M \partial {f^1}^*(-M^\top w) + \partial {f^2}^*(w)$

$$\begin{cases} k\text{th subproblem on } \partial {f^1}^* \\ k\text{th subproblem on } \partial {f^2}^* \end{cases}$$

Considering $x = (x^1, x^2)$

$$\min_{x^1, x^2} f^1(x^1) + f^2(x^2) \quad \text{s.t.} \quad A^1 x^1 + A^2 x^2 = 0$$

write Lagrangian to make kth-subproblems **easy**

Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{x^1,x^2} f^1(x^1) + f^2(x^2) \quad \text{s.t.} \quad A^1 x^1 + A^2 x^2 = 0$$

Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{x^1, x^2} f^1(x^1) + f^2(x^2) \quad \text{s.t.} \quad A^1 x^1 + A^2 x^2 = 0$$

Lagrangian

$$\begin{aligned} L(x, w) &= \sum_{s=1}^{2} \Big(f^s(x^s) + \langle A^{s\top} w, x^s \rangle \Big) \\ &= \sum_{s=1}^{2} L^s(x^s, w) \end{aligned}$$

Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{x^1,x^2} f^1(x^1) + f^2(x^2) \quad \text{s.t.} \quad A^1x^1 + A^2x^2 = 0$$

Lagrangian

$$\begin{aligned} L(x,w) &= \sum_{s=1}^{2} \left(f^s(x^s) + \langle A^{s\top} w, x^s \rangle \right) \\ &= \sum_{s=1}^{2} L^s(x^s, w) \end{aligned}$$

Lagrangian relaxation approach

Primal step Having dual iterate w^k, solve primal subproblems

$$\min_x L(x, w^k) = \sum_s \min_{\mathbf{x^s}} L(\mathbf{x^s}, w^k)$$

Dual step Use primal output to update dual variable

Rewriting often involves constraint with a simple subspace

Compressed sensing

$$\min_{x^1} \quad \frac{1}{2}\|RTx^1 - a\|_2^2 + \lambda\|A^1x^1\|_1$$

Rewriting often involves constraint with a simple subspace

Compressed sensing

$$\begin{cases} \min_{x^1,x^2} & \frac{1}{2}\|RTx^1 - a\|_2^2 + \lambda\| \quad x^2 \quad \|_1 \\ \text{s.t.} & A^1x^1 - x^2 = 0 \end{cases}$$

Rewriting often involves constraint with a simple subspace

Compressed sensing

$$\begin{cases} \min_{x^1, x^2} & \frac{1}{2}\|RTx^1 - a\|_2^2 + \lambda \| \quad x^2 \quad \|_1 \\ \text{s.t.} & A^1 x^1 - x^2 = 0 \end{cases}$$

$$x \in N := \{(x^1, x^2) : \begin{bmatrix} A^1 \\ -I \end{bmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = 0\}$$

Rewriting often involves constraint with a simple subspace

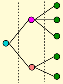
Compressed sensing

$$\begin{cases} \min_{x^1, x^2} & \frac{1}{2}\|RTx^1 - a\|_2^2 + \lambda\| \quad x^2 \quad \|_1 \\ \text{s.t.} & A^1 x^1 - x^2 = 0 \end{cases}$$

$$x \in N := \{(x^1, x^2) : \begin{bmatrix} A^1 \\ -I \end{bmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = 0\}$$

Progressive hedging

$$\begin{cases} \min_x & \mathbb{E}[f^s(x)] \\ \text{s.t.} & x \in X^s \qquad s \in S \end{cases}$$



Rewriting often involves constraint with a simple subspace

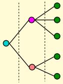
Compressed sensing

$$\begin{cases} \min_{x^1,x^2} & \frac{1}{2}\|RTx^1 - a\|_2^2 + \lambda\| \; x^2 \; \|_1 \\ \text{s.t.} & A^1x^1 - x^2 = 0 \end{cases}$$

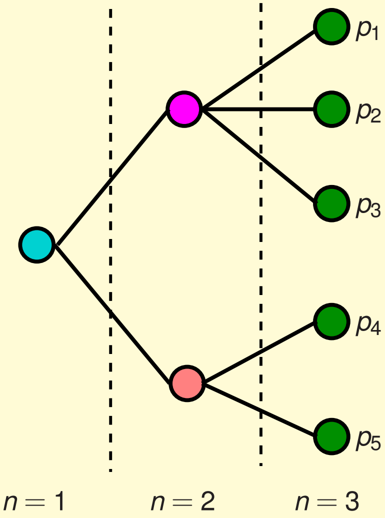
$$x \in N := \{(x^1, x^2) : \begin{bmatrix} A^1 \\ -I \end{bmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = 0\}$$

Progressive hedging

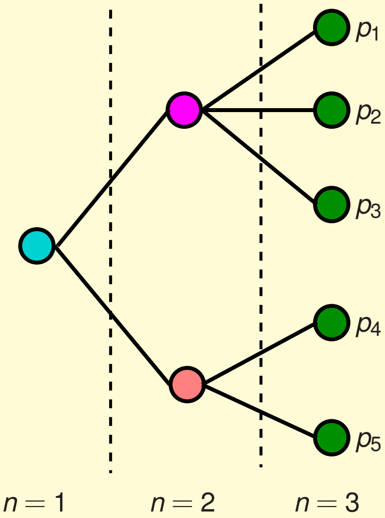
$$\begin{cases} \min_x & \mathbb{E}[f^s(x)] \\ \text{s.t.} & x \in X^s \qquad s \in S \end{cases} \iff \begin{cases} \min\limits_{x^s : s \in S} & \sum\limits_s p^s f^s(x^s) \\ \text{s.t.} & x^s \in X^s \qquad s \in S \\ & \sum_s A^s x^s = 0 \\ & \iff x \in N = ? \end{cases}$$

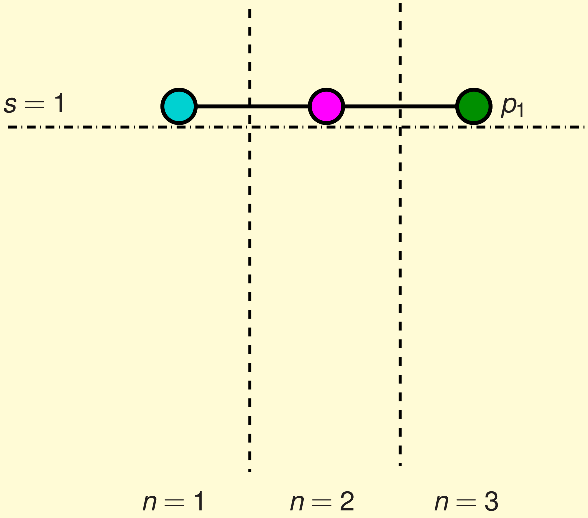


Progressive hedging rewriting

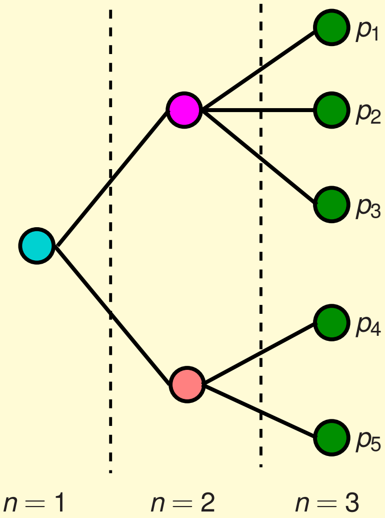


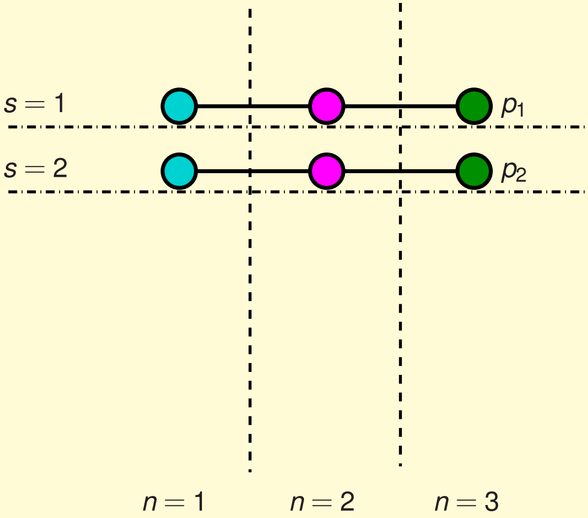
Progressive hedging rewriting



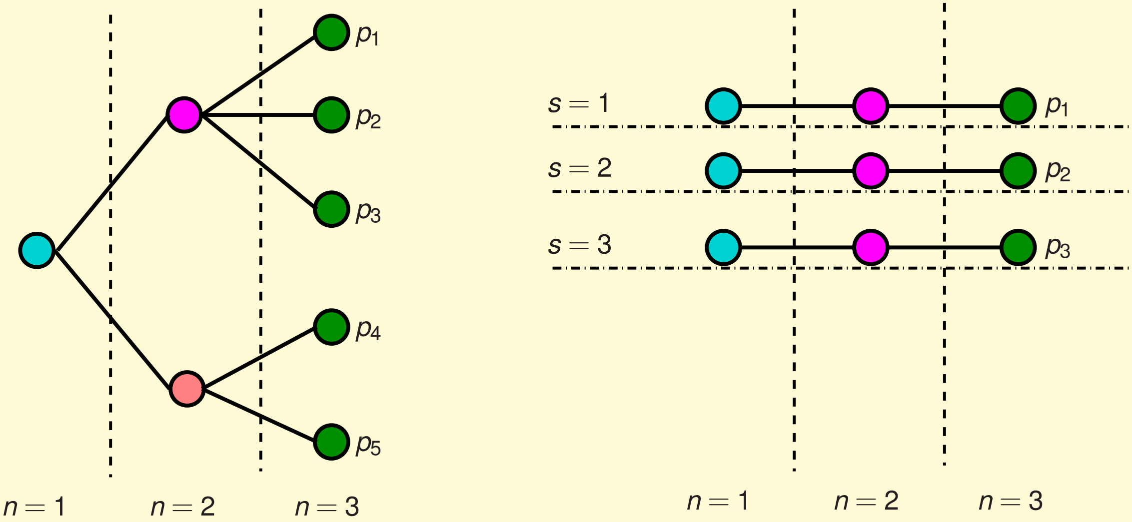


Progressive hedging rewriting

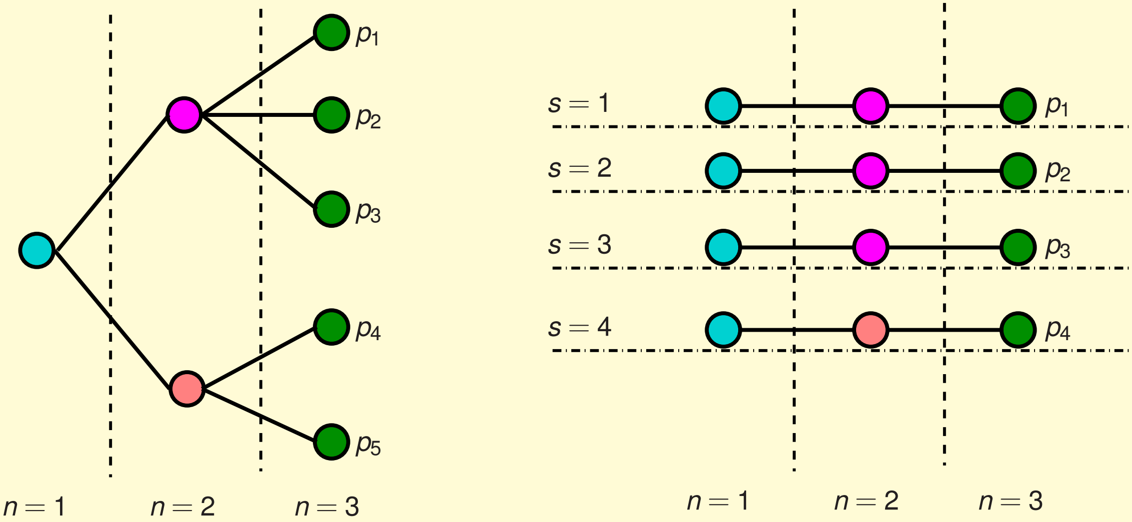




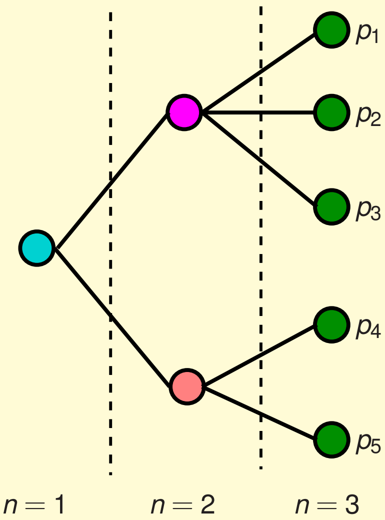
Progressive hedging rewriting

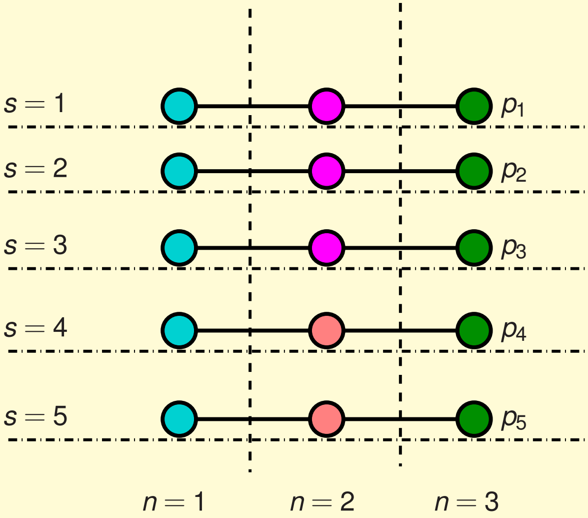


Progressive hedging rewriting

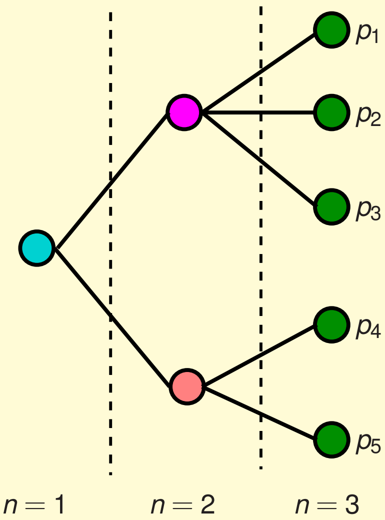


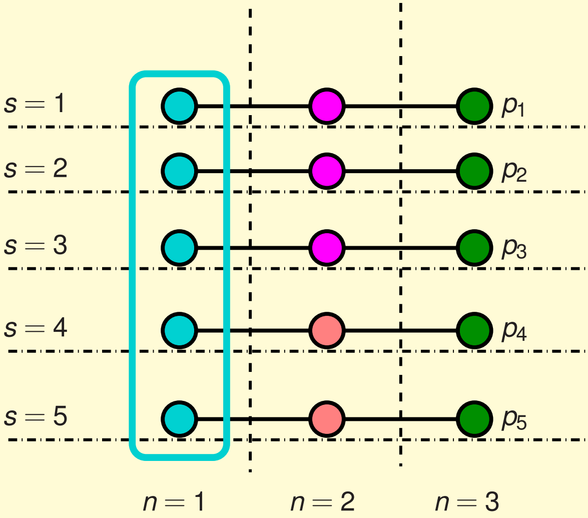
Progressive hedging rewriting



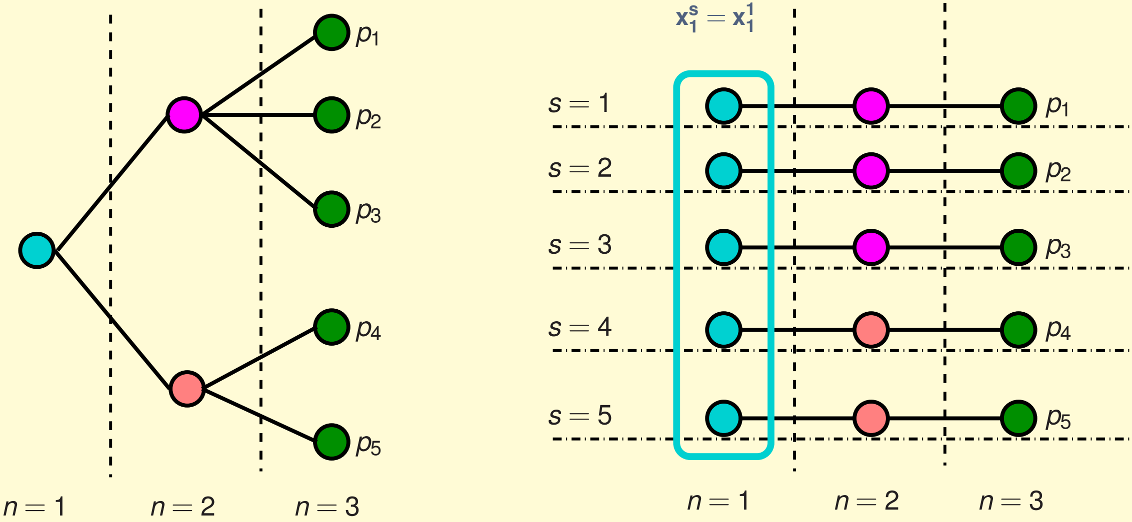


Progressive hedging rewriting

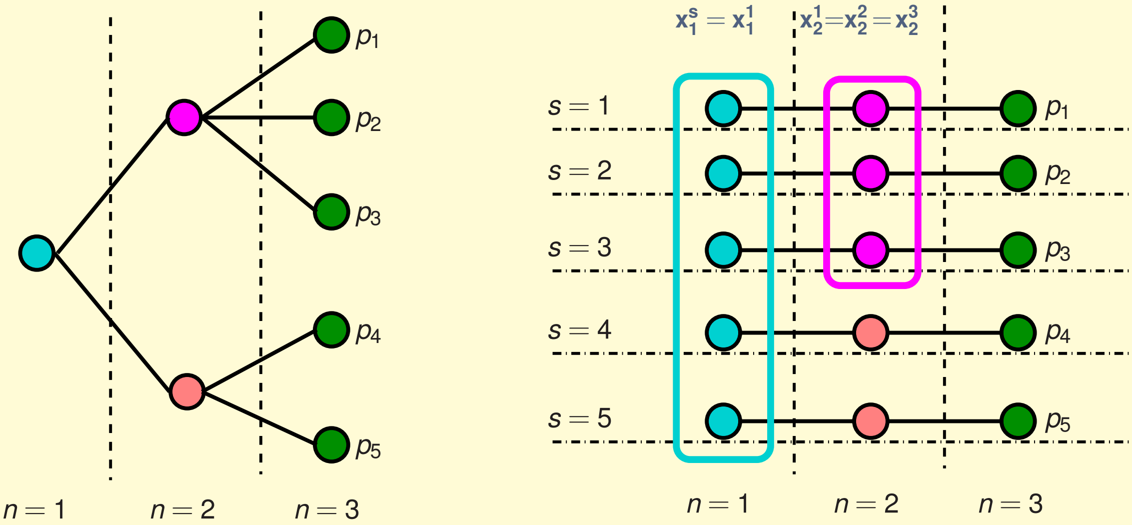


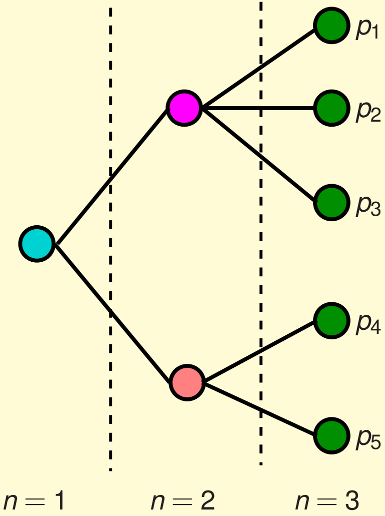


Progressive hedging rewriting

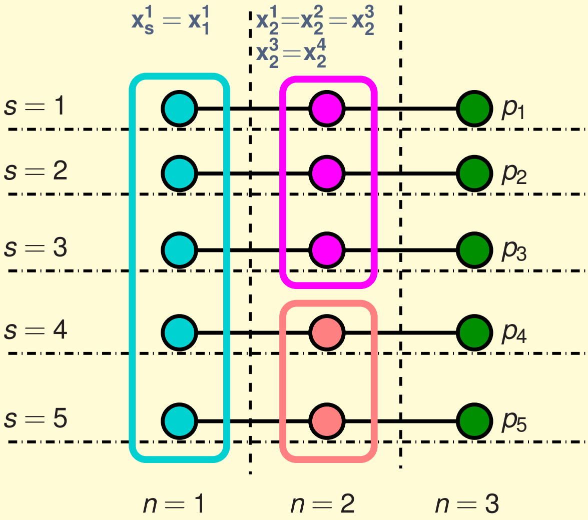


Progressive hedging rewriting



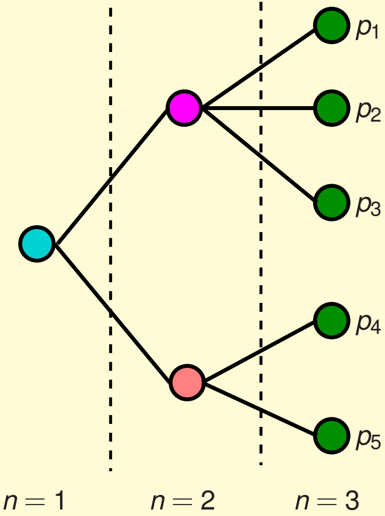


p_1
p_2
p_3
p_4
p_5
$n = 1$
$n = 2$
$n = 3$

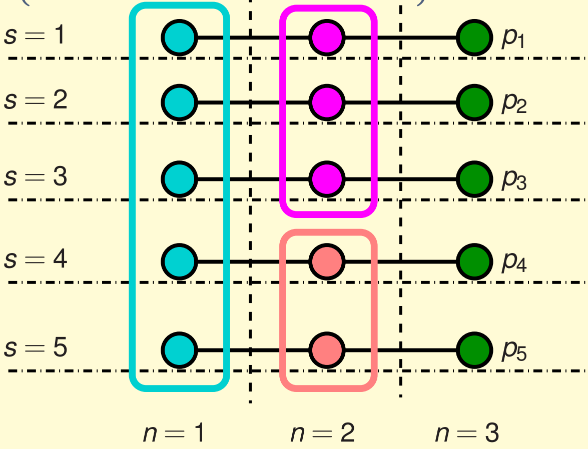


$\mathbf{x}_s^1 = \mathbf{x}_1^1$
$\mathbf{x}_2^1 = \mathbf{x}_2^2 = \mathbf{x}_2^3$
$\mathbf{x}_2^3 = \mathbf{x}_2^4$
$s = 1$
$s = 2$
$s = 3$
$s = 4$
$s = 5$
p_1
p_2
p_3
p_4
p_5
$n = 1$
$n = 2$
$n = 3$

Progressive hedging rewriting



$$\mathbf{N} := \left\{ \mathbf{x} = \mathbf{x}_n^s : \begin{array}{l} \mathbf{x}_s^1 = \mathbf{x}_1^1 \end{array} \; \begin{array}{l} \mathbf{x}_2^1 = \mathbf{x}_2^2 = \mathbf{x}_2^3 \\ \mathbf{x}_2^3 = \mathbf{x}_2^4 \end{array} \right\}$$



Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{(x^1,x^2,\ldots,x^s,\ldots)} \sum_s f^s(x^s) \quad \text{s.t.} \quad \sum_s A^s x^s = 0$$

Lagrangian

$$L_{t_0}(x,w) = \sum_s \left(f^s(x^s) + \langle A^{s\top} w, x^s \rangle \right) = \sum_s L^s(x^s, w)$$

Lagrangian relaxation approach

Primal step Having dual iterate w^k, solve primal subproblems

$$\min_x L\ (x, w^k) = \sum_s \min_{x^s} L\ (x^s, w^k)$$

Dual step Use primal output to update dual variable

Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{(x^1,x^2,\dots,x^s,\dots)} \sum_s f^s(x^s) \quad \text{s.t.} \quad \sum_s A^s x^s = 0$$

Augmented Lagrangian

$$L_{t_0}(x,w) \;=\; \sum_s \Big(f^s(x^s) + \langle A^{s\top} w, x^s \rangle \Big) + \frac{t_0}{2} \| \sum_s A^s x^s \|^2 = \sum_s L^s(x^s, w) + \frac{t_0}{2} \| \sum_s A^s x^s \|^2$$

Augmented Lagrangian relaxation approach (1st try, naïve)

Primal step Having dual iterate w^k, solve primal subproblems

$$\min_x L_{t_0}(x, w^k) \neq \sum_s \min_{\mathbf{x^s}} L^s_{t_0}(\mathbf{x^s}, w^k)$$

Dual step Use primal output to update dual variable

Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make kth-subproblems **easy**, for

$$\min_{(x^1,x^2,\ldots,x^s,\ldots)} \sum_s f^s(x^s) \quad \text{s.t.} \quad \sum_s A^s x^s = 0$$

Augmented Lagrangian is not separable

$$\begin{aligned} L_{t_0}(x,w) &= \sum_s \left(f^s(x^s) + \langle A^{s\top} w, x^s \rangle \right) + \frac{t_0}{2} \| \sum_s A^s x^s \|^2 = \sum_s L^s(x^s, w) + \frac{t_0}{2} \| \sum_s A^s x^s \|^2 \\ &\approx \mathbb{L}^k(x,w) \textit{separable} \end{aligned}$$

Augmented Lagrangian relaxation approach

Primal step Having dual iterate w^k, solve approximate primal subproblems

$$\min_x \mathbb{L}^k(x, w^k) = \sum_s \min_{\mathbf{x^s}} L_{t_0}\left((\mathbf{x^s}, x^{k,-s}), w^k \right)$$

Project primal output onto **N**

Dual step Use primal output to update dual variable

The progressive hedging algorithm (RW91)

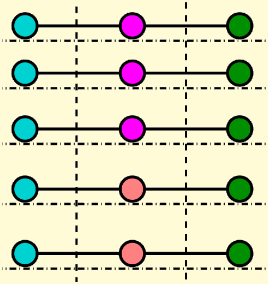
Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$
and a fixed prox-parameter $t_0 > 0$

The progressive hedging algorithm (RW91)

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$

and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$

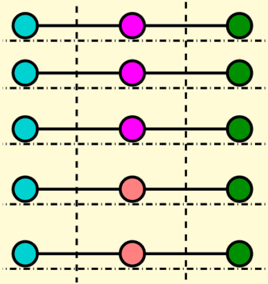


The progressive hedging algorithm (RW91)

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$

and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$

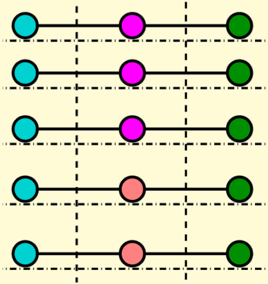


2. projects onto $\mathbf{N}$ to define x^{k+1}

The progressive hedging algorithm (RW91)

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$
and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$



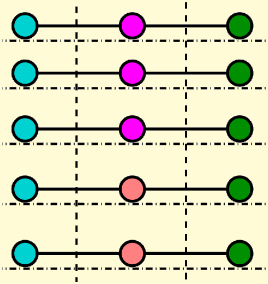
2. projects onto $\mathbf{N}$ to define x^{k+1}
3. computes $w^{k+1} = w^k + t_0(x^{k+\frac{1}{2}} - x^k)$

The progressive hedging algorithm (RW91)

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$

and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$



BUT convergence relies on DR for OC

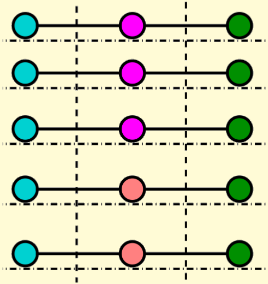
cannot vary t_0, it remains fixed!

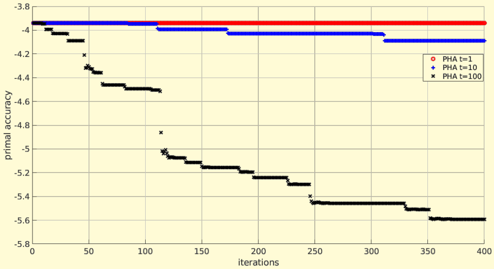
2. projects onto $\mathbf{N}$ to define x^{k+1}
3. computes $w^{k+1} = w^k + t_0(x^{k+\frac{1}{2}} - x^k)$

The progressive hedging algorithm (RW91)

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^\perp$
and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$





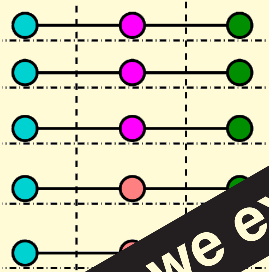
$t_0 = 1$
$t_0 = 10$
$t_0 = 100$

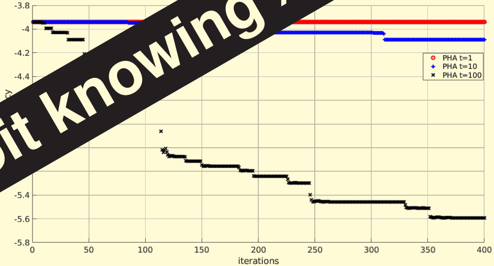
2. projects onto $\mathbf{N}$ to define x^{k+1}
3. computes $w^{k+1} = w^k + t_0(x^{k+\frac{1}{2}} - x^k)$

Given $x^k = (x^{k,s} : s \in S) \in \mathbf{N}$ and $w^k = (w^{k,s} : s \in S) \in \mathbf{N}^{\perp}$

and a fixed prox-parameter $t_0 > 0$

1. solves in parallel $x^{k+\frac{1}{2},s} = \arg\min_{\mathbf{x^s}}$ … $s), w^{k,s})$





$t_0 = 1$

$t_0 = 10$

$t_0 = 100$

Can we exploit knowing $A = \partial f$?

2. …ts onto **N** to define x^{k+1}

3. computes $w^{k+1} = w^k + t_0(x^{k+\frac{1}{2}} - x^k)$

The projective trick

1. Find primal intermediate points in parallel

$$x^{k+\frac{1}{2},s} \;=\; \arg\min_{\mathbf{x^s}} L_{t_0}((\mathbf{x^s}, x^{k,-s}), w^{k,s})$$

2. Project primal iterate onto **N**
3. Update dual iterate

The projective trick

1. Find primal intermediate points in parallel

$$
\begin{aligned}
x^{k+\frac{1}{2},s} &= \arg\min_{x^s} L_{t_0}((x^s, x^{k,-s}), w^{k,s}) \\
x^{k+\frac{1}{2},s} &= \mathtt{prox}^{t_0}_{\mathbb{L}^{k,s}}(\cdot, w^{k,s})(x^{k,s})
\end{aligned}
$$

2. Project primal iterate onto **N**
3. Update dual iterate

The projective trick

1. Find primal intermediate points in parallel

$$x^{k+\frac{1}{2},s} = \arg\min_{x^s} L_{t_0}((x^s, x^{k,-s}), w^{k,s})$$
$$x^{k+\frac{1}{2},s} = \mathtt{prox}^{t_0}_{\mathbb{L}^{k,s}}(\cdot, w^{k,s})(x^{k,s})$$

2. Project primal iterate onto **N**
3. Update dual iterate

$\Longrightarrow$ let's interpret primal update in terms of the dual problem

The projective trick

1. Find primal intermediate points in parallel

$$x^{k+\frac{1}{2},s} = \arg\min_{x^s} L_{t_0}((x^s, x^{k,-s}), w^{k,s})$$
$$x^{k+\frac{1}{2},s} = \mathtt{prox}^{t_0}_{\mathbb{L}^{k,s}}(\cdot, w^{k,s})(x^{k,s})$$

2. Project primal iterate onto **N**
3. Update dual iterate

$\Longrightarrow$ let's interpret primal update in terms of the dual problem

$$\text{DUAL solves iteratively} \begin{cases} \min & H(w) = \sum_s H^s(w^s) \\ \text{s.t.} & w \in \mathbf{N}^\perp \end{cases}$$

by computing proximal points for individual models $\mathbb{H}^{k,s}$

The projective trick in the dual

1. Find **dual** intermediate points in parallel

$$w^{k+\frac{1}{2},s} \;=\; \mathtt{prox}_{\mathbb{H}^{k,s}}^{\frac{1}{t_0}}(w^{k,s})$$

2. Project **dual** iterate onto $\mathbf{N}^\perp$
3. Update **primal** iterate

The projective trick in the dual

1. Find **dual** intermediate points in parallel

$$w^{k+\frac{1}{2},s} \;=\; \mathtt{prox}^{\frac{1}{t_0}}_{\mathbb{H}^{k,s}}(w^{k,s})$$

2. Project **dual** iterate onto $\mathbf{N}^{\perp}$
3. Update **primal** iterate

$\Longrightarrow$ by computing proximal points for individual models $\mathbb{H}^{k,s}$ in the dual problem we can now compare

the dual models $\mathbb{H}^{k,s}$ with the dual functions H^s

The projective trick in the dual

1. Find **dual** intermediate points in parallel

$$w^{k+\frac{1}{2},s} \;=\; \mathtt{prox}^{\frac{1}{t_0}}_{\mathbb{H}^{k,s}}(w^{k,s})$$

2. Project **dual** iterate onto $\mathbf{N}^{\perp}$
3. Update **primal** iterate

$\Longrightarrow$ by computing proximal points for individual models $\mathbb{H}^{k,s}$ in the dual problem we can now compare

the dual models $\mathbb{H}^{k,s}$ with the dual functions H^s

– Bad fit: decrease t_0

– Good fit: ok! t_0 can increase

The projective trick in the dual

1. Find **dual** intermediate points in parallel

$$w^{k+\frac{1}{2},s} \;=\; \mathtt{prox}^{\frac{1}{t_0}}_{\mathbb{H}^{k,s}}(w^{k,s})$$

2. Project **dual** iterate onto $\mathbf{N}^{\perp}$
3. Update **primal** iterate

$\Longrightarrow$ by computing proximal points for individual models $\mathbb{H}^{k,s}$ in the dual problem we can now compare

the dual models $\mathbb{H}^{k,s}$ with the dual functions H^s

– Bad fit: decrease t_0

– Good fit: ok! t_0 can increase

Bad/good fit dichotomy $\equiv$ null/serious steps in bundle methods

Goodies of Bundle PH - convergence

$\Longrightarrow$ allows to increase/decrease t_k

fits model-based descent theory (Atenas, Sagastizábal, Silva, Solodov, SiOPT 2023), extended to handle projective step(Atenas, Sagastizábal, JoCA 2023)

Goodies of Bundle PH - convergence

$\Longrightarrow$ allows to increase/decrease t_k

fits model-based descent theory (Atenas, Sagastizábal, Silva, Solodov, SiOPT 2023), extended to handle projective step(Atenas, Sagastizábal, JoCA 2023)

If $t_{k+1} \in [t_{\min}, t_{\max}]$

- Convergence for infinite subsequence of serious steps

–Global convergence with linear rate if error bound

- Convergence for infinite tail of null steps

–Last generated serious iterate was optimal

–The null tail converges to last serious

–no rate

Goodies of Bundle PH - convergence

$\Longrightarrow$ allows to increase/decrease t_k

fits model-based descent theory (Atenas, Sagastizábal, Silva, Solodov, SiOPT 2023), extended to handle projective step(Atenas, Sagastizábal, JoCA 2023)

If $t_{k+1} \in [t_{\min}, t_{\max}]$

- Convergence for infinite subsequence of serious steps

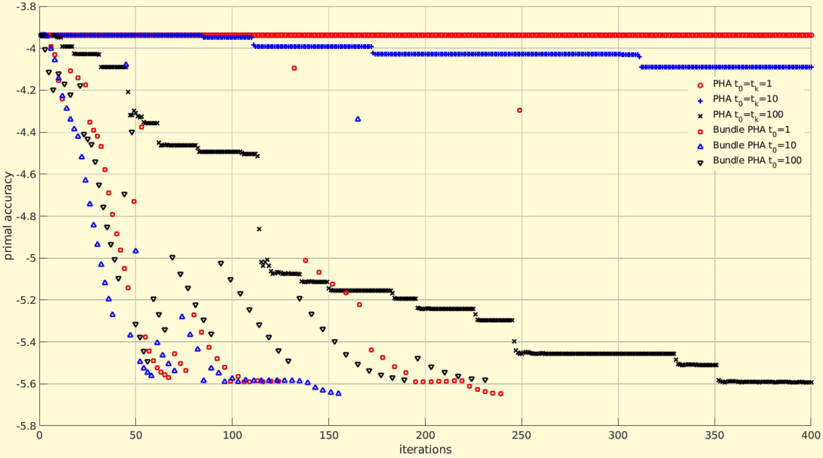
–Global convergence with linear rate if error bound

- Convergence for infinite tail of null steps

–Last generated serious iterate was optimal
–The null tail converges to last serious
–no rate

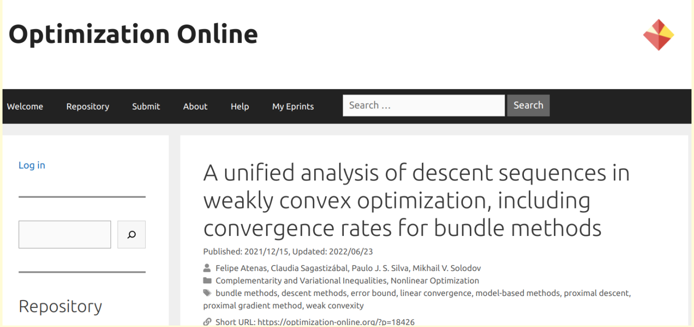
$\Longrightarrow$ implementable stopping test!

Goodies of Bundle PH - Performance



To know more: related references

JoCA
vol 30(2), 2023



SiOPT
vol 33(1), 2023

Bonus track

- The family of proximal decomposition methods (separable augmented Lagrangians by Philippe Mahey, Adam Ouorou, Jean-Pierre Dussault, co-authors)

$$\begin{cases} \min_x & \sum_j f_j(x_j) & \\ \text{s.t.} & x_j \in S_j & \forall j \\ & \sum_j g_j(x_j) = 0 & \end{cases}$$

Bonus track

- The family of proximal decomposition methods (separable augmented Lagrangians by Philippe Mahey, Adam Ouorou, Jean-Pierre Dussault, co-authors)

$$\left\{\begin{array}{lll} \min_{x,y} & \sum_j f_j(x_j) & \\ \text{s.t.} & x_j \in S_j & \forall j \\ & g_j(x_j) + y_j = 0 & \forall j \\ & \sum_j y_j = 0 & \end{array}\right.$$

Bonus track

- The family of proximal decomposition methods (separable augmented Lagrangians by Philippe Mahey, Adam Ouorou, Jean-Pierre Dussault, co-authors)

$$\begin{cases} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in S_j \quad \forall j \\ & g_j(x_j) + y_j = 0 \quad \forall j \\ & \sum_j y_j = 0 \end{cases} \iff (y_1, \dots, y_j, \dots) \in \mathbf{N}$$

Bonus track

- The family of proximal decomposition methods (separable augmented Lagrangians by Philippe Mahey, Adam Ouorou, Jean-Pierre Dussault, co-authors)

$$\begin{cases} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in S_j & \forall j \\ & g_j(x_j) + y_j = 0 & \forall j \\ & \sum_j y_j = 0 & \iff (y_1, \ldots, y_j, \ldots) \in \mathbf{N} \end{cases}$$

- The unified theory extends those methods to weakly convex problems + linear rate + stopping test + varying prox-parameter!

Bonus track

- The family of proximal decomposition methods (separable augmented Lagrangians by Philippe Mahey, Adam Ouorou, Jean-Pierre Dussault, co-authors)

$$\begin{cases} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in S_j \qquad \forall j \\ & g_j(x_j) + y_j = 0 \quad \forall j \\ & \sum_j y_j = 0 \qquad \iff (y_1, \dots, y_j, \dots) \in \mathbf{N} \end{cases}$$

- The unified theory extends those methods to weakly convex problems + linear rate + stopping test + varying prox-parameter!

- What about Decomposition-Coordination Methods ?
  (separable augmented Lagrangians by Pierre Carpentier, Guy Cohen, Jean-Christophe Culioli, co-authors `https://doi.org/10.1007/978-3-642-46823-0_6`