

# 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**

- Minimize immediate cost by emptying reservoirs

**Consequences**

**in the future**

{	rain	Decision ok
	drought	Deficit

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

## Decision

### today

- Minimize immediate cost by emptying reservoirs

{	rain	Decision ok
	drought	Deficit

### or

- Keep water, more \$\$\$ by thermal generation

{	rain	Excess water
	drought	Decision ok

## Consequences

### in the future

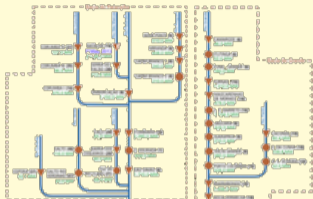
## The value of water is an opportunity/substitution cost

Given by the value function of a linear stochastic program

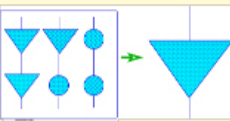
Depends on

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

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



Agenças	
CEASA - 3	GERMUN - 28
CEEL - 2	For do Rio Claro - 29
CEEL - 3	ITP - 30
CELPLA - 4	Itaipu - 31
CEISO - 5	Itaipu - 32
CEISUL - 6	Itaipu - 33
CEISUL - 7	Itaipu - 34
CEISUL - 8	Itaipu - 35
CEISUL - 9	Itaipu - 36
CEISUL - 10	Itaipu - 37
CEISUL - 11	Itaipu - 38
CEISUL - 12	Itaipu - 39
CEISUL - 13	Itaipu - 40
CEISUL - 14	Itaipu - 41
CEISUL - 15	Itaipu - 42
CEISUL - 16	Itaipu - 43
CEISUL - 17	Itaipu - 44
CEISUL - 18	Itaipu - 45
CEISUL - 19	Itaipu - 46
CEISUL - 20	Itaipu - 47
CEISUL - 21	Itaipu - 48
CEISUL - 22	Itaipu - 49
CEISUL - 23	Itaipu - 50
CEISUL - 24	Itaipu - 51
CEISUL - 25	Itaipu - 52
CEISUL - 26	Itaipu - 53
CEISUL - 27	Itaipu - 54
CEISUL - 28	Itaipu - 55
CEISUL - 29	Itaipu - 56
CEISUL - 30	Itaipu - 57
CEISUL - 31	Itaipu - 58
CEISUL - 32	Itaipu - 59
CEISUL - 33	Itaipu - 60
CEISUL - 34	Itaipu - 61
CEISUL - 35	Itaipu - 62
CEISUL - 36	Itaipu - 63
CEISUL - 37	Itaipu - 64
CEISUL - 38	Itaipu - 65
CEISUL - 39	Itaipu - 66
CEISUL - 40	Itaipu - 67
CEISUL - 41	Itaipu - 68
CEISUL - 42	Itaipu - 69
CEISUL - 43	Itaipu - 70
CEISUL - 44	Itaipu - 71
CEISUL - 45	Itaipu - 72
CEISUL - 46	Itaipu - 73
CEISUL - 47	Itaipu - 74
CEISUL - 48	Itaipu - 75
CEISUL - 49	Itaipu - 76
CEISUL - 50	Itaipu - 77
CEISUL - 51	Itaipu - 78
CEISUL - 52	Itaipu - 79
CEISUL - 53	Itaipu - 80
CEISUL - 54	Itaipu - 81
CEISUL - 55	Itaipu - 82
CEISUL - 56	Itaipu - 83
CEISUL - 57	Itaipu - 84
CEISUL - 58	Itaipu - 85
CEISUL - 59	Itaipu - 86
CEISUL - 60	Itaipu - 87
CEISUL - 61	Itaipu - 88
CEISUL - 62	Itaipu - 89
CEISUL - 63	Itaipu - 90
CEISUL - 64	Itaipu - 91
CEISUL - 65	Itaipu - 92
CEISUL - 66	Itaipu - 93
CEISUL - 67	Itaipu - 94
CEISUL - 68	Itaipu - 95
CEISUL - 69	Itaipu - 96
CEISUL - 70	Itaipu - 97
CEISUL - 71	Itaipu - 98
CEISUL - 72	Itaipu - 99
CEISUL - 73	Itaipu - 100

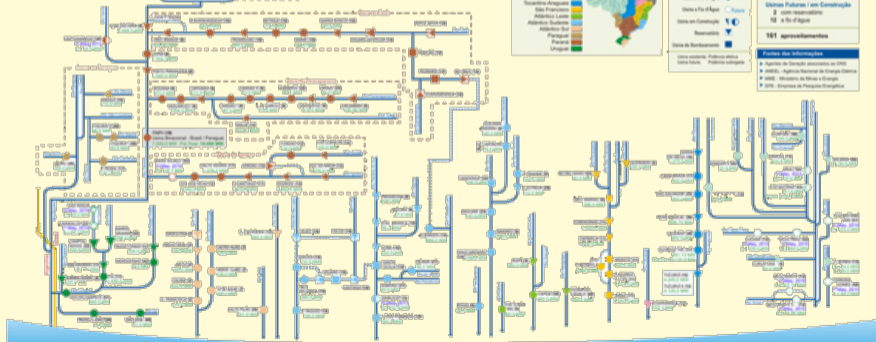


Bacias Hidrográficas	
Araguari	Amarelo
Atibaia NE	Verde
Atibaia NW	Verde
Paranaíba	Verde
Tocantins	Verde
São Francisco	Verde
Atibaia Leste	Verde
Atibaia Sudeste	Verde
Atibaia Sul	Verde
Paranaíba	Verde
Paranaíba	Verde
Uruguai	Verde

Legenda	
Usina com Reservatório	▲
Usina sem Reservatório	●
Usina a Fio d'Água	○
Usina em Construção	▲
Reservatório	■
Sítio de Bombamento	■
Usina existente	▲
Usina futura	●

Legenda	
Aproveitamentos Existentes	▲
com reservatório	■
a fio d'água	○
em construção	▲
Usinas Futuras / em Construção	●
com reservatório	■
a fio d'água	○
161 aproveitamentos	

Fontes das Informações	
▲ Agência de Energia Interligada do SIN	
● ANEL - Agência Nacional de Energia Elétrica	
■ ANEEL - Ministério de Minas e Energia	
○ ONS - Empresa de Pesquisa Energética	



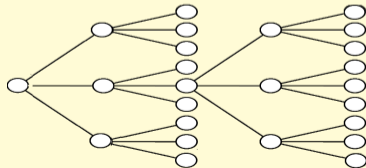
# Future cost of water: piecewise linear function

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

# Future cost of water: piecewise linear function

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

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



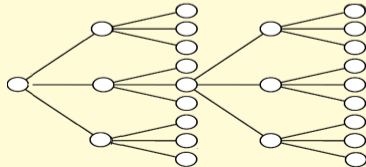


# Future cost of water: piecewise linear function

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

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



$$\min_{u,x} \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^{NS} \sum_{j=1}^{T_t^i} c_t^j g_t^{i,j} \right]$$

$$x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i \quad \text{(BAL)}$$

$$gh_t^i + \sum_{j \leq T_t^i} g_t^{i,j} + \sum_{\ell \in \mathcal{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \geq dem_t^i - (1 - \gamma_t^i) \xi_t^i \quad \text{(DEM)}$$

$$u^{\min} \leq u = (gh, spill, gt, f) \leq u^{\max}$$

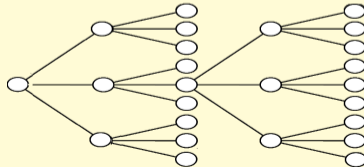
$$x_t^{i,\min} \leq x_t^i \leq x_t^{i,\max} \quad \text{(BOX)}$$

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!)



$$\min_{u,x} \mathbb{E} \left[ \sum_{t=1}^T \sum_{i=1}^{NS} \sum_{j=1}^{T_i^i} c_t^j g_t^{i,j} \right]$$

$$x_t^i + gh_t^i + spill_t^i = x_{t-1}^i + \gamma_t^i \xi_t^i - evap_t^i \quad \text{(BAL)}$$

$$gh_t^i + \sum_{j \leq T_t^i} g_t^{i,j} + \sum_{\ell \in \mathcal{L}^i} (f_t^{\ell,i} - f_t^{i,\ell}) \geq dem_t^i - (1 - \gamma_t^i) \xi_t^i \quad \text{(DEM)}$$

$$u^{\min} \leq u = (gh, spill, gt, f) \leq u^{\max}$$

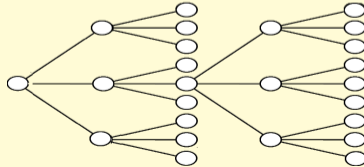
$$x_t^{i,\min} \leq x_t^i \leq x_t^{i,\max} \quad \text{(BOX)}$$

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

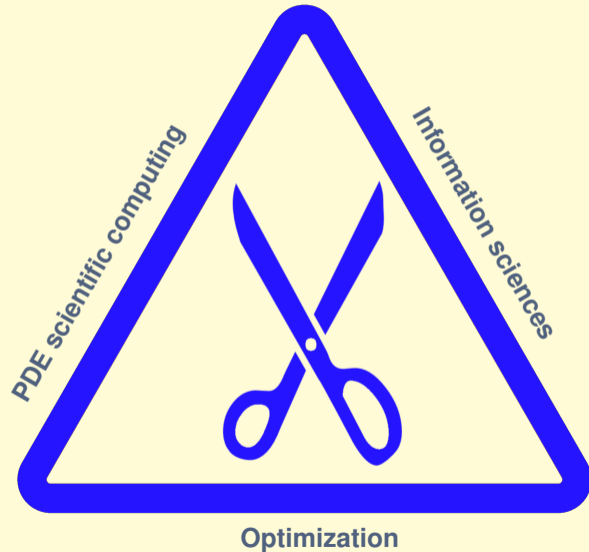
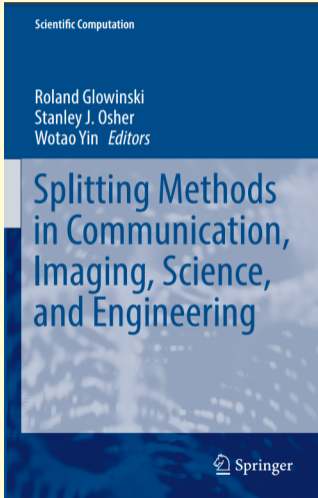
real problem considers 10 years

( $T = 120$  months)

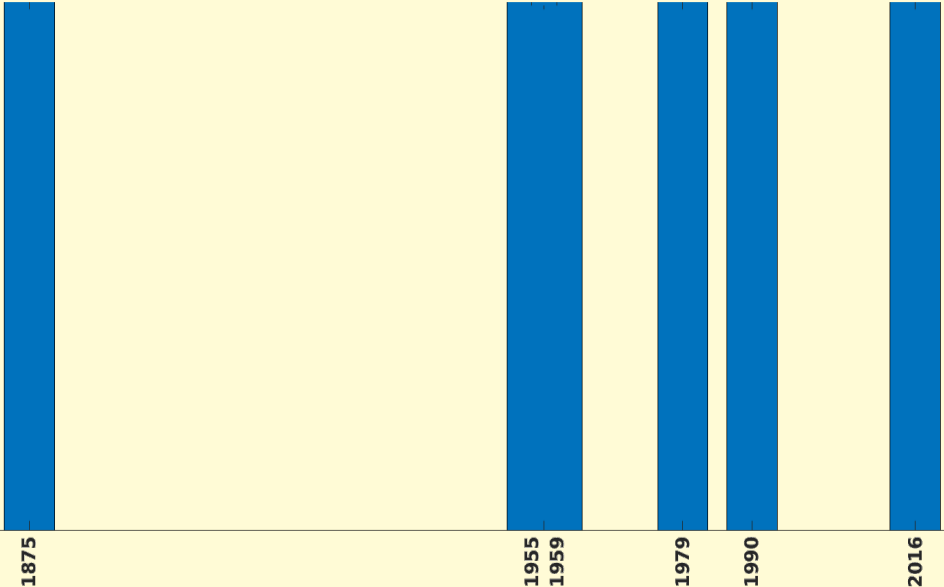
it has  $20^{119}$  scenarios!!!



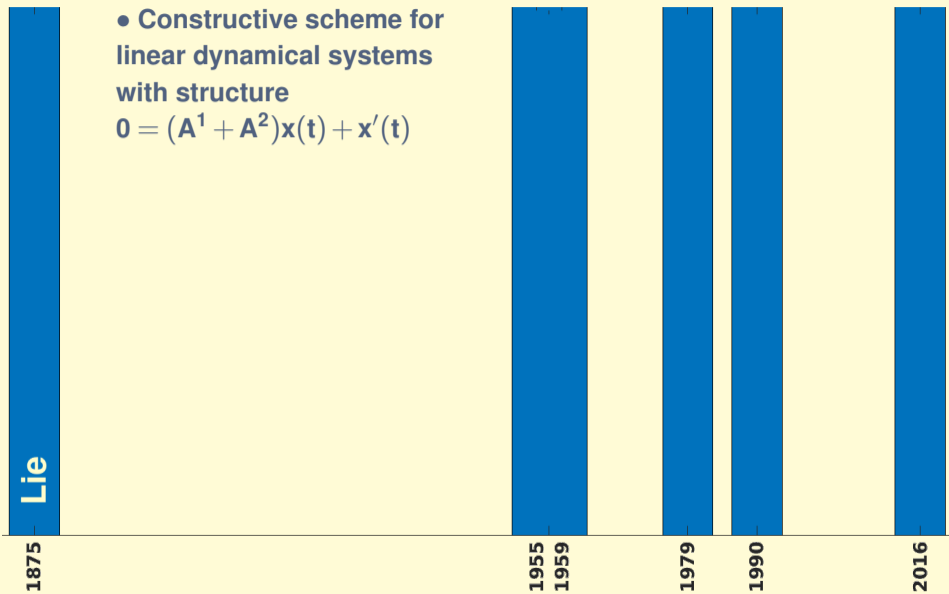
# The Triangle of Splitting Methods



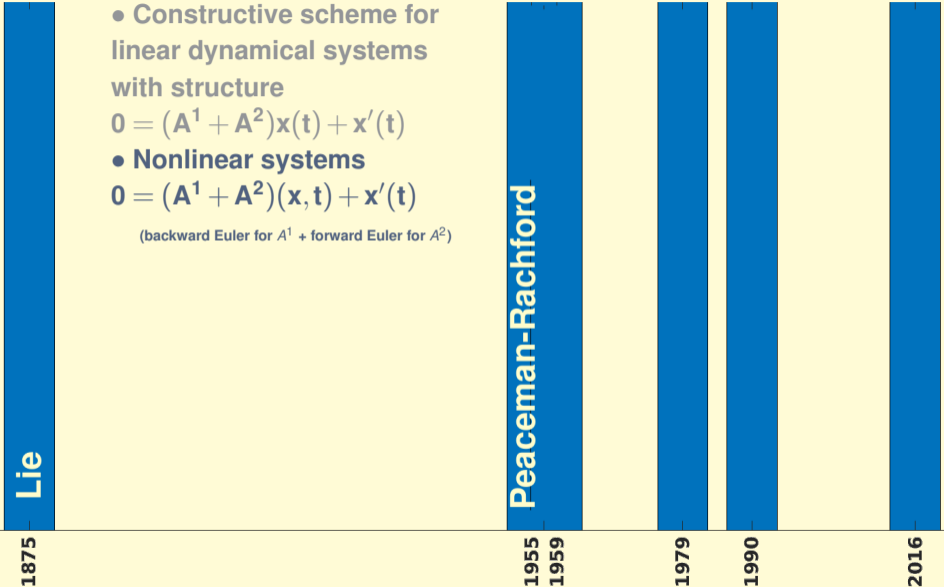
# Splitting Timeline - some milestones



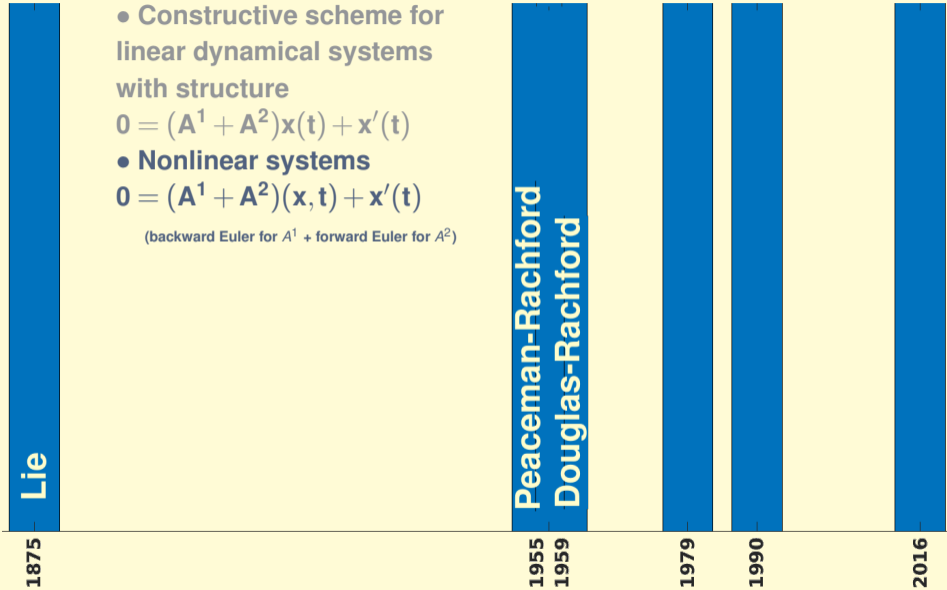
# Splitting Timeline - some milestones



# Splitting Timeline - some milestones

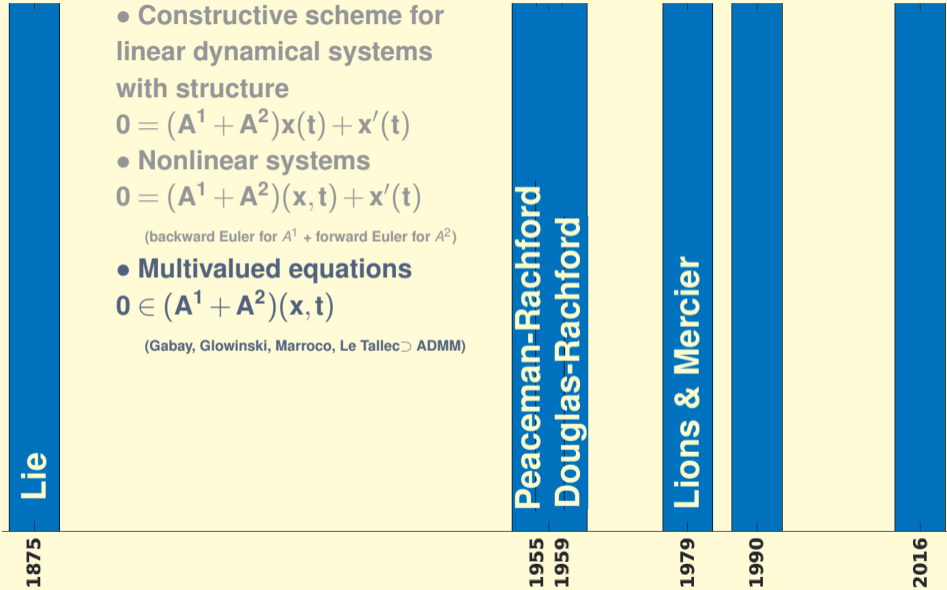


# Splitting Timeline - some milestones

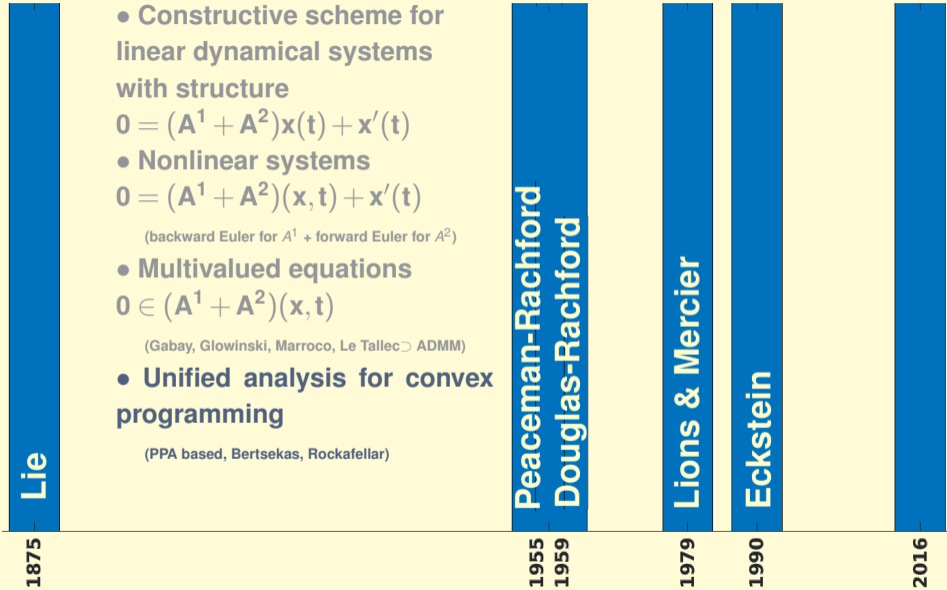




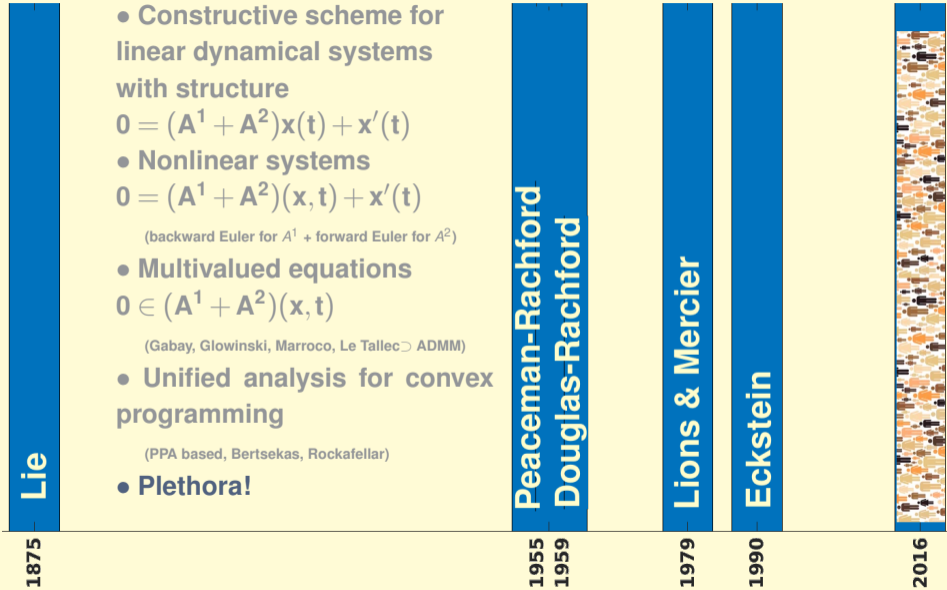
# Splitting Timeline - some milestones



# Splitting Timeline - some milestones



# Splitting Timeline - some milestones





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



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





## Un petit détour...



source: Sarah Dry's wordpress



# Newton's method for nonlinear systems

$$0 = G(x^*)$$

$$\approx G(x^k) + G'(x^k)d^k$$

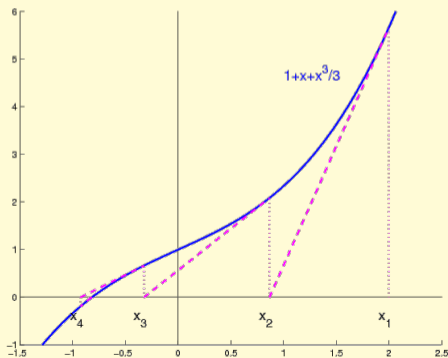
**fast** convergence  $x^{k+1} = x^k + d^k$

# Newton's method for nonlinear systems

$$0 = G(x^*)$$

$$\approx G(x^k) + G'(x^k)d^k$$

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



# Newton method is **accurate**

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

1	2.000000000000000	0
2	<b>0.866666666666667</b>	1
3	<b>-0.32323745064862</b>	1
4	<b>-0.92578663808031</b>	1
5	<b>-0.82332584261905</b>	2
6	<b>-0.81774699537697</b>	5
7	<b>-0.81773167400186</b>	9
8	<b>-0.81773167388682</b>	15

**Newton**

# What about Newton's method for optimization?


$$0 = G(x^*)$$

$$\approx G(x^k) + G'(x^k)d^k$$

 **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?

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

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

 **fast** convergence

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

for an objective  $f$

In optimization

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

# 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

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

In optimization

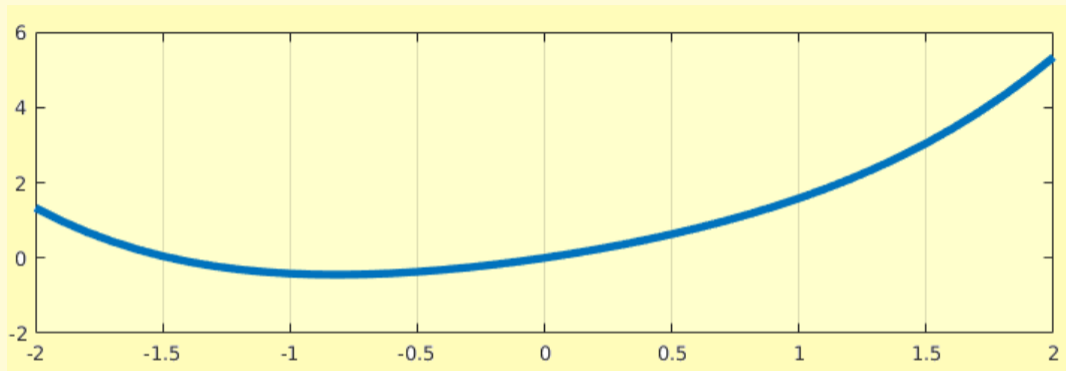
$$G(x) = \nabla f(x)$$

for an objective  $f$

$$\min f \approx \min \mathbf{f\text{-model}}$$

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

# Newton iterates for optimization

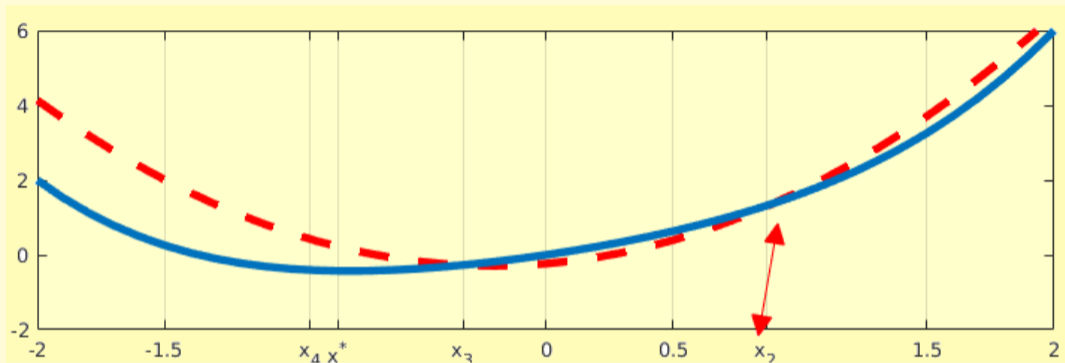


$$G(x) = 1 + x + x^3/3$$

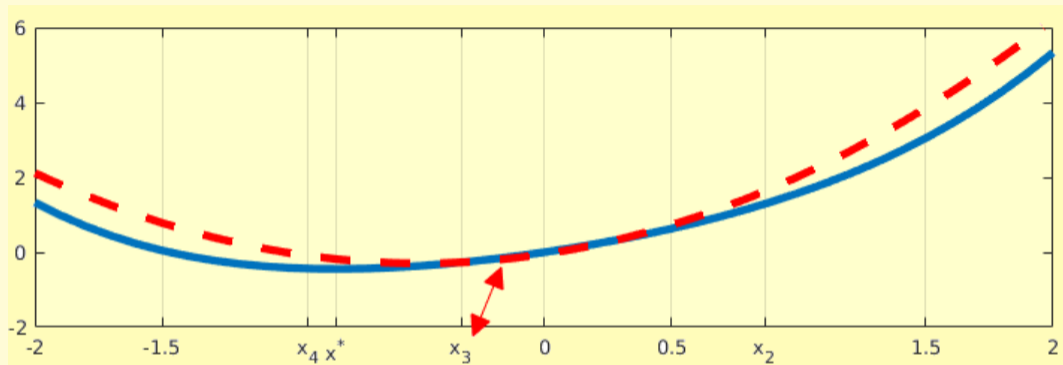
$$\implies f(x) = x + x^2/2 + x^4/12$$



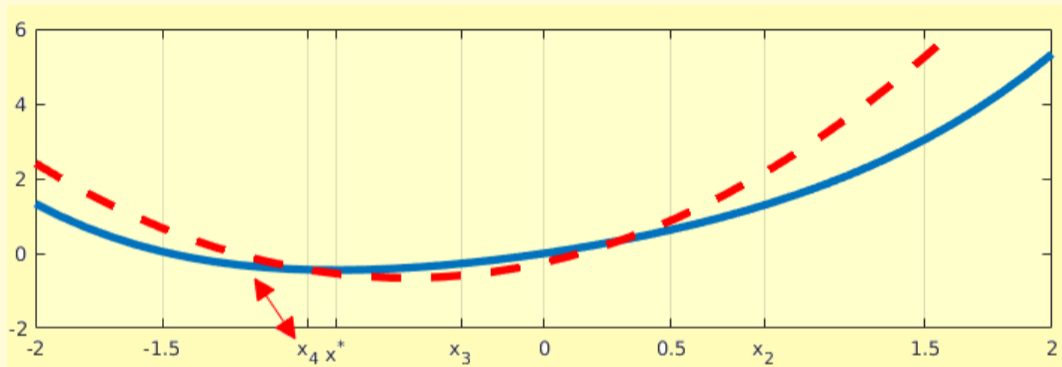
# Newton iterates for optimization



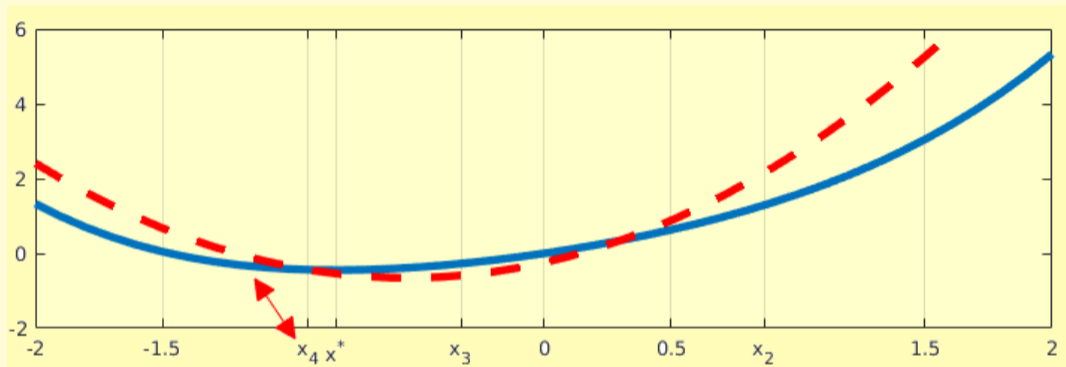
# Newton iterates for optimization



# Newton iterates for optimization

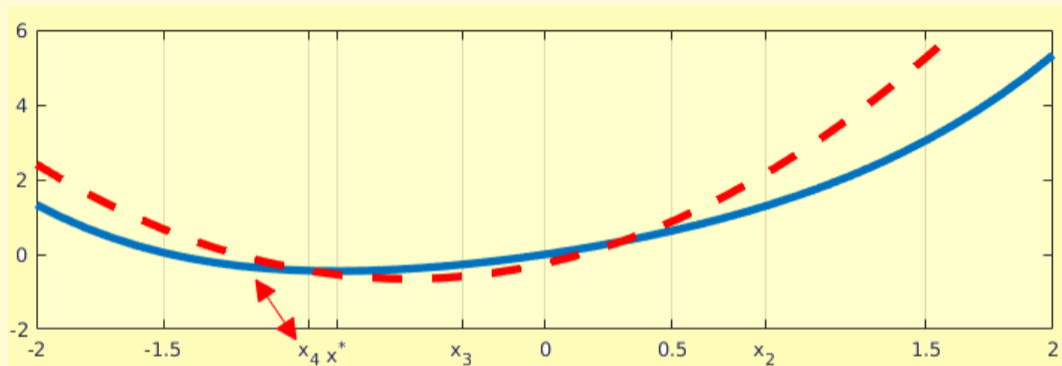


# Newton iterates for optimization



Can we avoid computing the Hessian matrix?

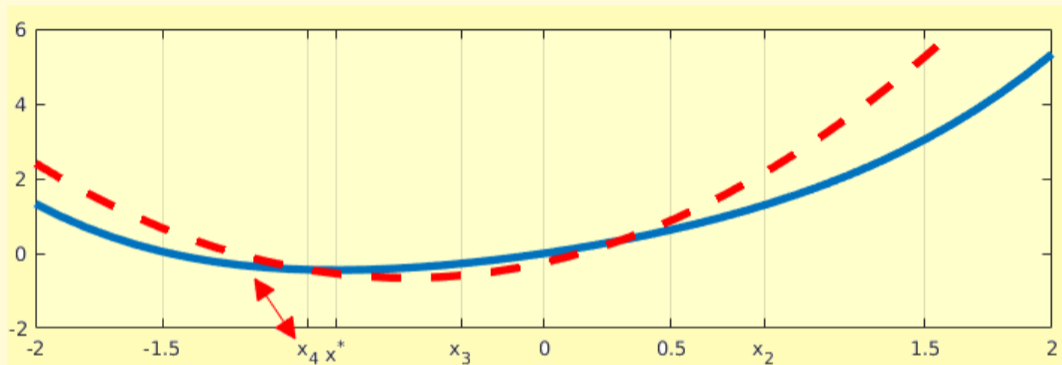
# Newton iterates for optimization



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

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

# Newton iterates for optimization



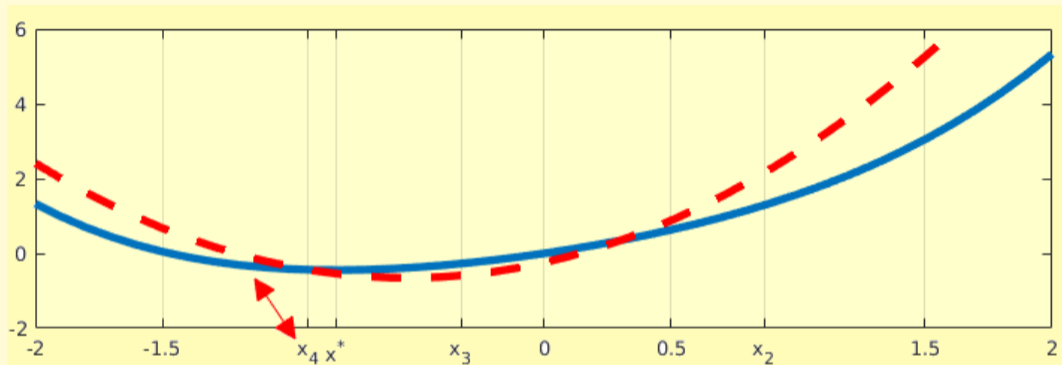
Can we avoid computing the Hessian matrix?

**YES!**

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

**quasi-Newton matrix**

# Newton iterates for optimization



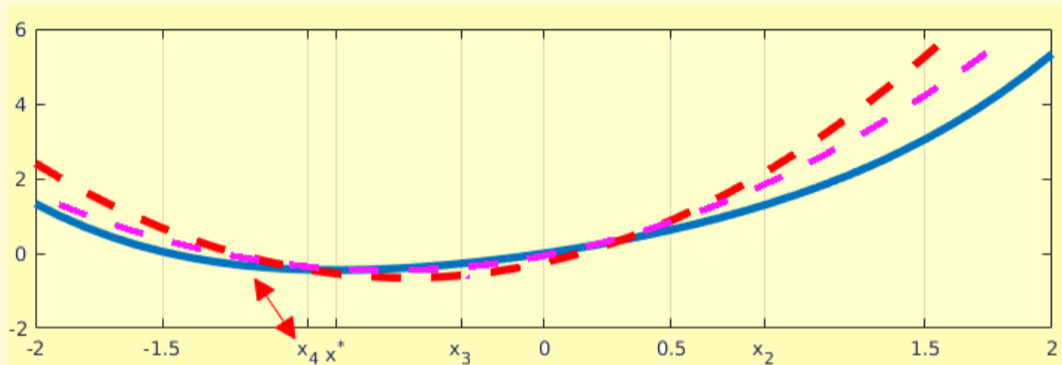
Can we avoid computing the Hessian matrix? **YES!**

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

**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!

1	2.000000000000000	0
2	1.500000000000000	0
3	<b>0.61224489795918</b>	1
4	<b>-0.16202797536640</b>	1
5	<b>-0.92209500449059</b>	1
6	<b>-0.78540447895661</b>	1
7	<b>-0.81609056319699</b>	3
8	<b>-0.81775774021392</b>	5
9	<b>-0.81773165292101</b>	8
10	<b>-0.81773167388656</b>	13
11	<b>-0.81773167388682</b>	15

**quasi-Newton**

1	2.000000000000000	0
2	<b>0.866666666666667</b>	1
3	<b>-0.32323745064862</b>	1
4	<b>-0.92578663808031</b>	1
5	<b>-0.82332584261905</b>	2
6	<b>-0.81774699537697</b>	5
7	<b>-0.81773167400186</b>	9
8	<b>-0.81773167388682</b>	15

**Newton**

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

1	2.000000000000000	0
2	1.500000000000000	0
3	<b>0.61224489795918</b>	1
4	<b>-0.16202797536640</b>	1
5	<b>-0.92209500449059</b>	1
6	<b>-0.78540447895661</b>	1
7	<b>-0.81609056319699</b>	3
8	<b>-0.81775774021392</b>	5
9	<b>-0.81773165292101</b>	8
10	<b>-0.81773167388656</b>	13
11	<b>-0.81773167388682</b>	15

**quasi-Newton**

1	2.000000000000000	0
2	<b>0.866666666666667</b>	1
3	<b>-0.32323745064862</b>	1
4	<b>-0.92578663808031</b>	1
5	<b>-0.82332584261905</b>	2
6	<b>-0.81774699537697</b>	5
7	<b>-0.81773167400186</b>	9
8	<b>-0.81773167388682</b>	15

**Newton**

**... fin du détour**

## Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

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

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

$$\left\{ \begin{array}{l} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{array} \right.$$

## Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

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

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

$$\left\{ \begin{array}{l} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{array} \right.$$

or its dual:

$$\max_w -f^{1*}(-M^\top w) - f^{2*}(w)$$

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

$$\left\{ \begin{array}{l} k\text{th subproblem on } \partial f^{1*} \\ k\text{th subproblem on } \partial f^{2*} \end{array} \right.$$

## Splitting variants stem from primal and dual formulations

Applied to optimality conditions of

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

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

$$\left\{ \begin{array}{l} k\text{th subproblem on } \partial f^1 \\ k\text{th subproblem on } \partial f^2 \end{array} \right.$$

or its dual:

$$\max_w -f^{1*}(-M^\top w) - f^{2*}(w)$$

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

$$\left\{ \begin{array}{l} k\text{th subproblem on } \partial f^{1*} \\ k\text{th subproblem on } \partial f^{2*} \end{array} \right.$$

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

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

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

$$\left\{ \begin{array}{l} \text{kth subproblem on } \partial f^1 \\ \text{kth subproblem on } \partial f^2 \end{array} \right.$$

or its dual:

$$\max_w -f^{1*}(-M^\top w) - f^{2*}(w)$$

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

$$\left\{ \begin{array}{l} \text{kth subproblem on } \partial f^{1*} \\ \text{kth subproblem on } \partial f^{2*} \end{array} \right.$$

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  $k$ th-subproblems **easy**

## Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make  $k$ th-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  $k$ th-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

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



# Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make  $k$ th-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

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

## Lagrangian relaxation approach

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

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

**Dual step** Use primal output to update dual variable

## Compressed sensing

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

## Compressed sensing

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

## Compressed sensing

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

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

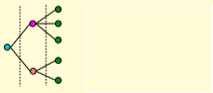
# Rewriting often involves constraint with a simple subspace

## Compressed sensing

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

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

## Progressive hedging

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


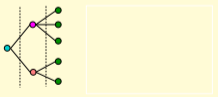
# Rewriting often involves constraint with a simple subspace

## Compressed sensing

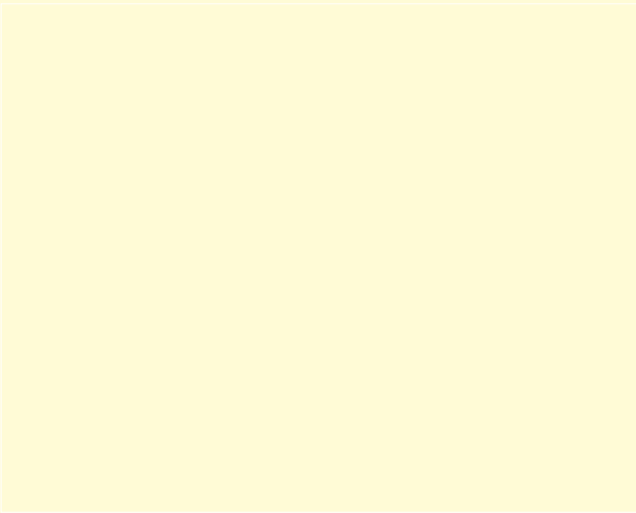
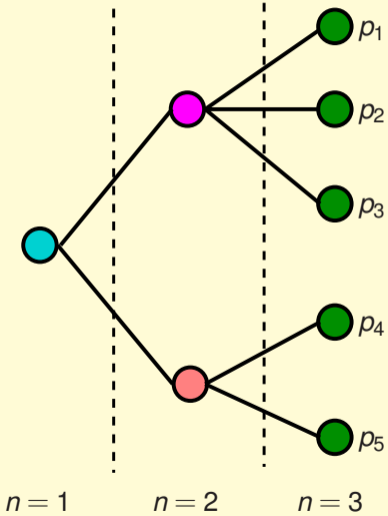
$$\begin{cases} \min_{x^1, x^2} & \frac{1}{2} \|RTx^1 - a\|_2^2 + \lambda \|x^2\|_1 \\ \text{s.t.} & A^1 x^1 - x^2 = 0 \end{cases}$$

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

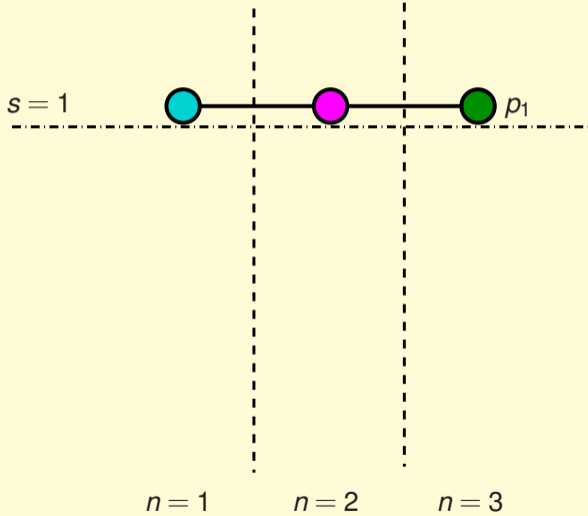
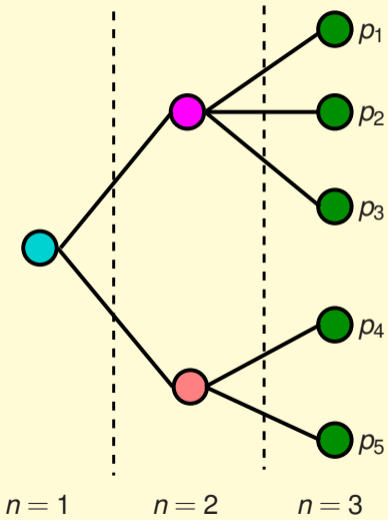
## Progressive hedging

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


# Progressive hedging rewriting

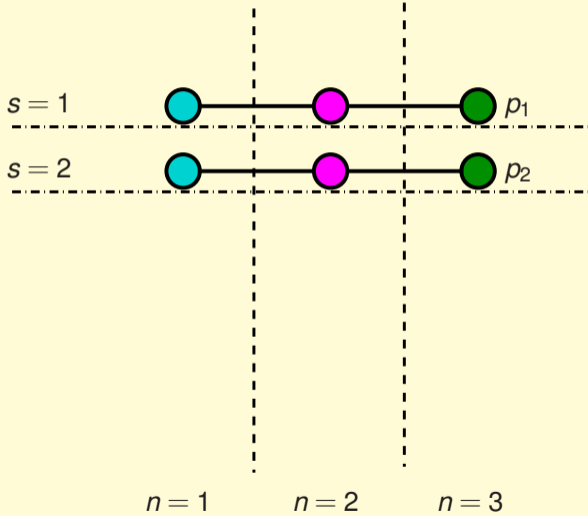
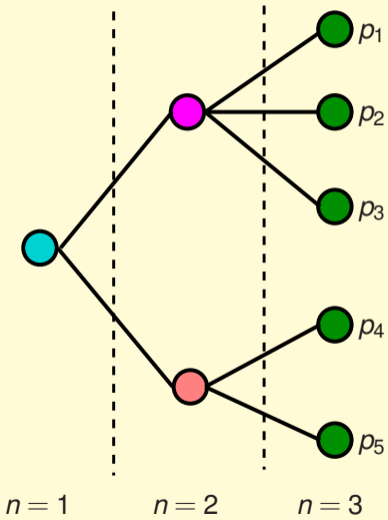


# Progressive hedging rewriting

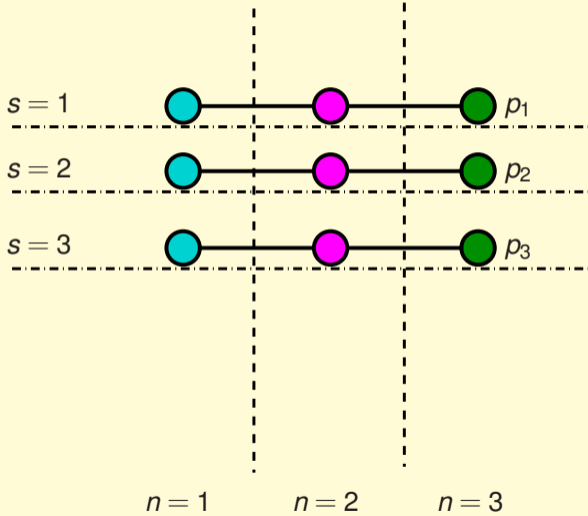
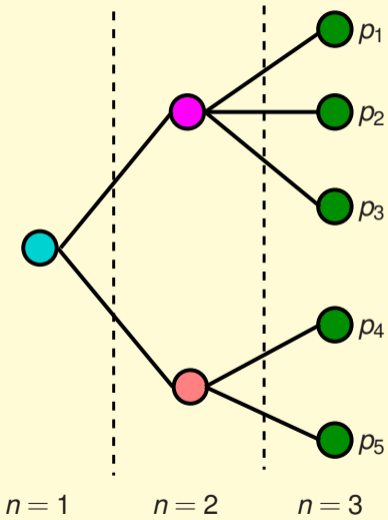




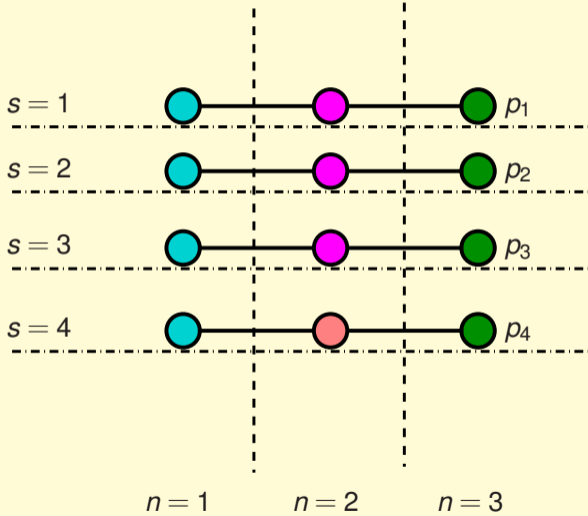
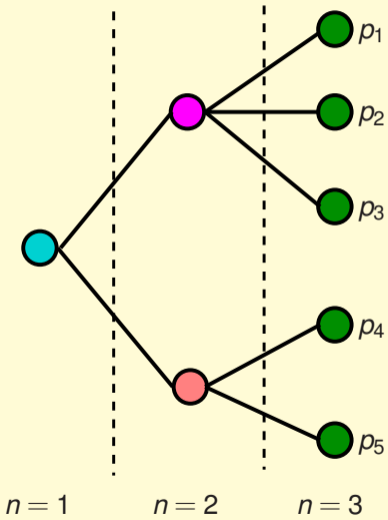
# Progressive hedging rewriting



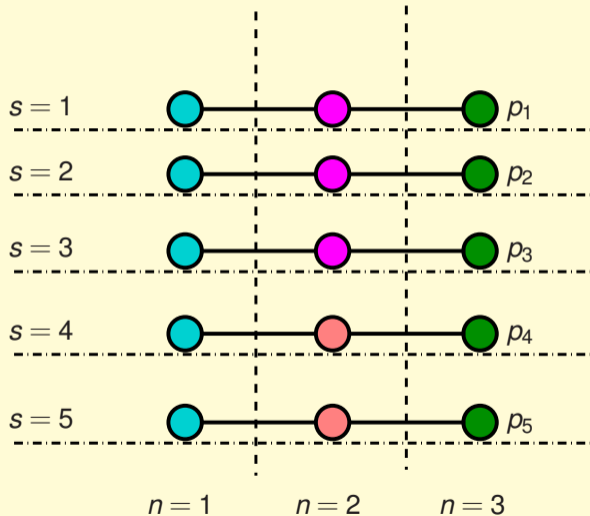
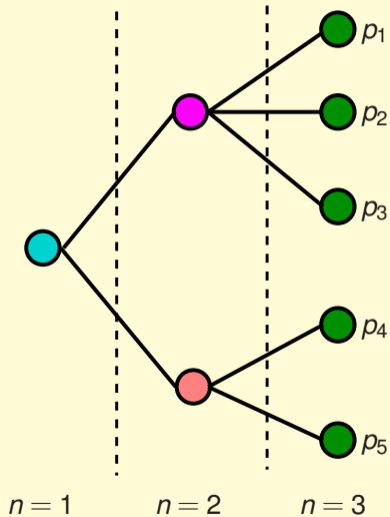
# Progressive hedging rewriting



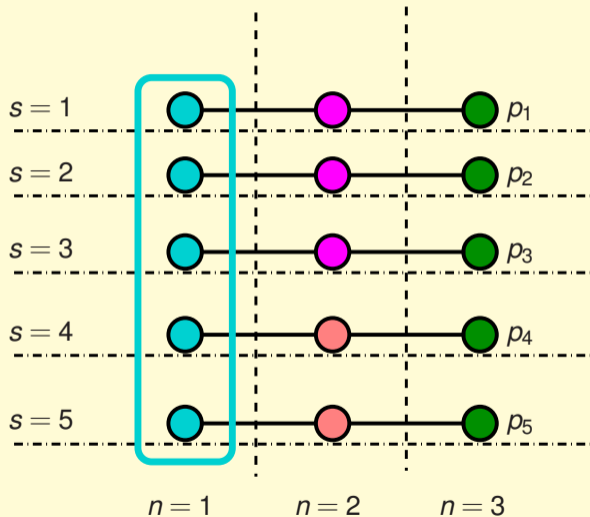
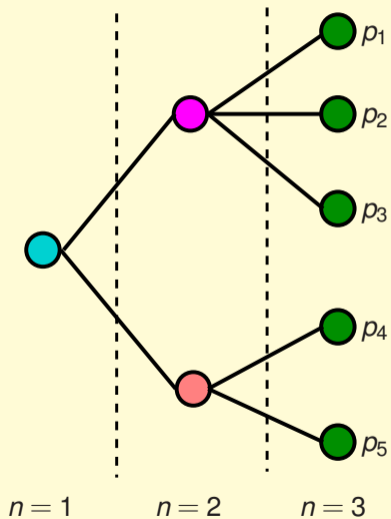
# Progressive hedging rewriting



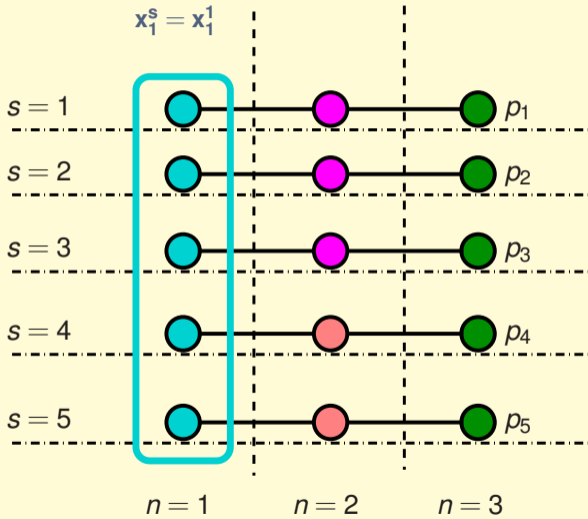
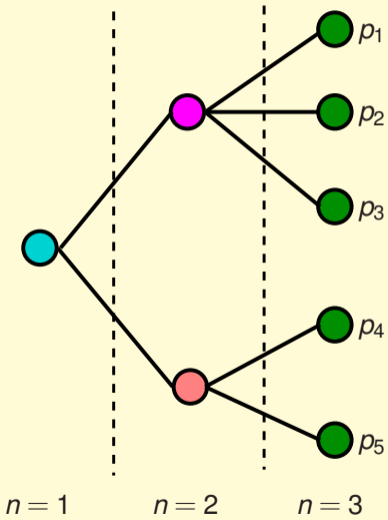
# Progressive hedging rewriting



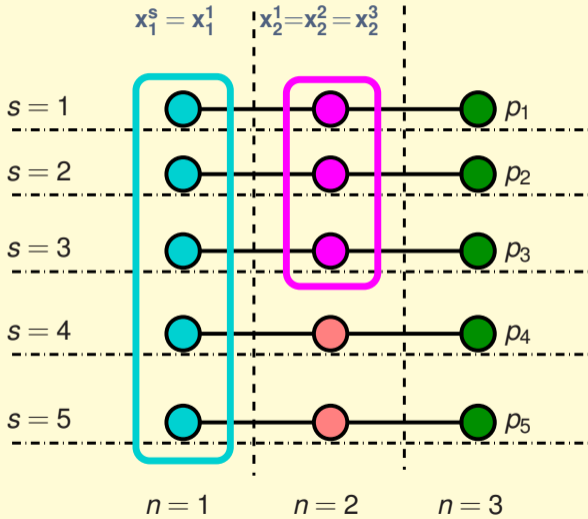
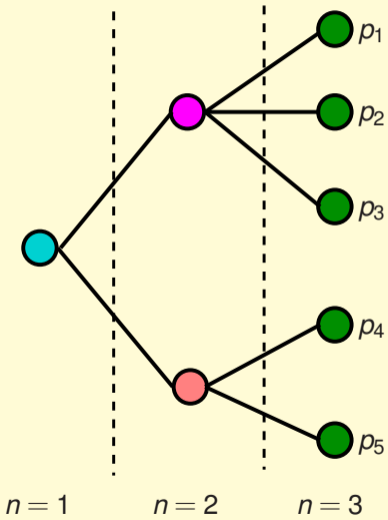
# Progressive hedging rewriting



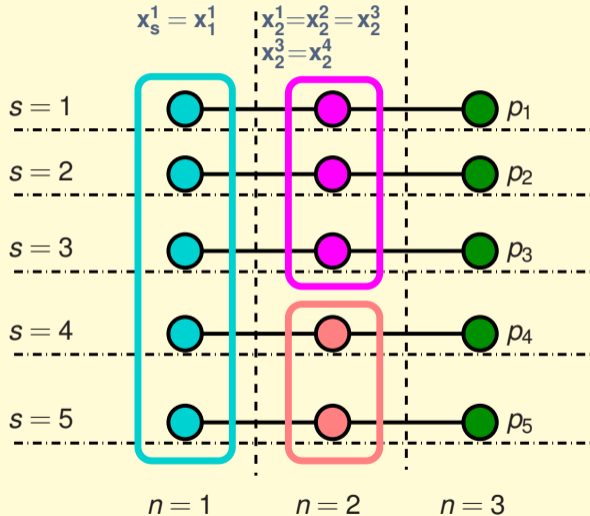
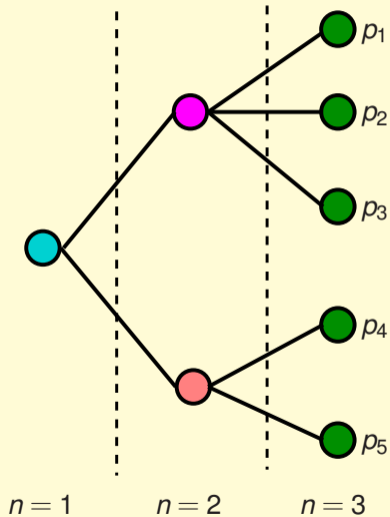
# Progressive hedging rewriting



# Progressive hedging rewriting

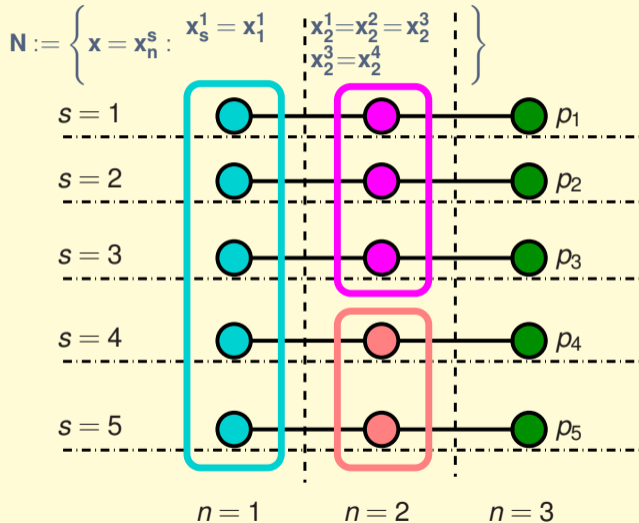
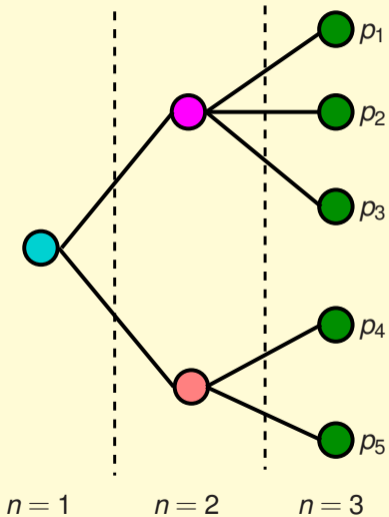


# Progressive hedging rewriting





# Progressive hedging rewriting



# Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make  $k$ th-subproblems **easy**, for

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

## 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  $k$ th-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

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

## Augmented Lagrangian relaxation approach (1st try, naïve)

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

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

**Dual step** Use primal output to update dual variable

# Splitting variants stem from primal and dual formulations

Rewriting often results from efforts to make  $k$ th-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 is not separable

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

Augmented Lagrangian relaxation approach

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

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

**Project** primal output onto **N**

**Dual step** Use primal output to update dual variable

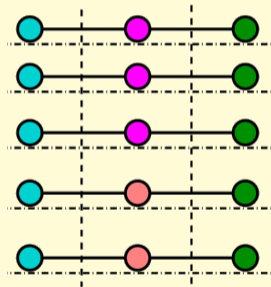
## The progressive hedging algorithm (RW91)

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  $t_0 > 0$

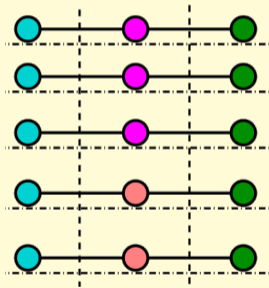
1. solves in parallel  $x^{k+\frac{1}{2},s} = \arg \min_{x^s} L_{t_0}((\mathbf{x}^s, x^{k,-s}), w^{k,s})$



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

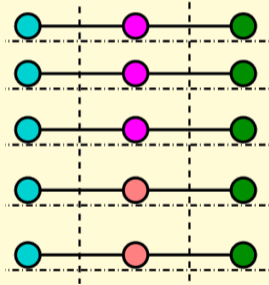


2. projects onto  $\mathbf{N}$  to define  $x^{k+1}$

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



2. projects onto

$\mathbf{N}$  to define  $x^{k+1}$

3. computes

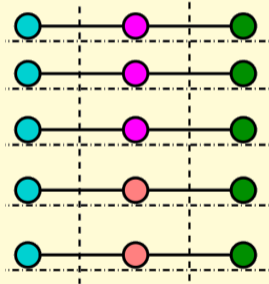
$$w^{k+1} = w^k + t_0(x^{k+\frac{1}{2}} - x^k)$$



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



**BUT** convergence relies on DR for OC  
cannot vary  $t_0$ , it remains fixed!

2. projects onto

$\mathbf{N}$  to define  $x^{k+1}$

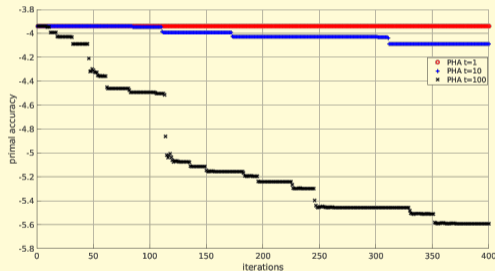
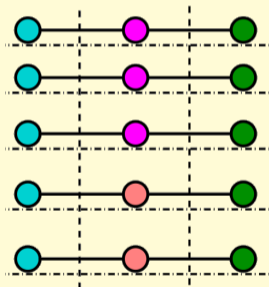
3. computes

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

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



$t_0 = 1$

$t_0 = 10$

$t_0 = 100$

2. projects onto

3. computes

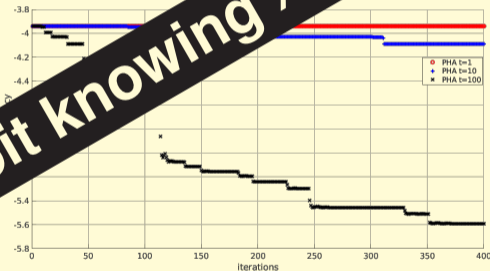
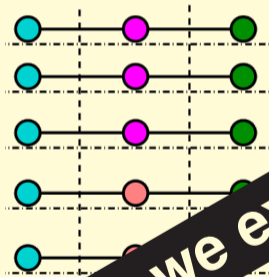
$\mathbf{N}$  to define  $x^{k+1}$

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

# 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  $x^{k+\frac{1}{2},s} = \arg \min_{x^s} L(x^s, w^{k,s})$



$t_0 = 1$

$t_0 = 10$

$t_0 = 100$

2. projects onto

$\mathbf{N}$  to define  $x^{k+1}$

3. computes

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

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

## 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  $\mathbf{N}$
3. Update dual iterate

## The projective trick

1. Find primal intermediate points in parallel

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

$$x^{k+\frac{1}{2},s} = \text{prox}_{\mathbb{L}^{k,s}}^{t_0}(\cdot, w^{k,s})(x^{k,s})$$

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

## The projective trick

1. Find primal intermediate points in parallel

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

$$x^{k+\frac{1}{2},s} = \text{prox}_{\mathbb{L}^{k,s}}^{t_0}(\cdot, w^{k,s})(x^{k,s})$$

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

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

# The projective trick

1. Find primal intermediate points in parallel

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

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

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

⇒ 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

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

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

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

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

3. Update **primal** iterate

⇒ 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

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

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

3. Update **primal** iterate

⇒ 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

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

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

3. Update **primal** iterate

⇒ 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

⇒ 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

⇒ 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

⇒ 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

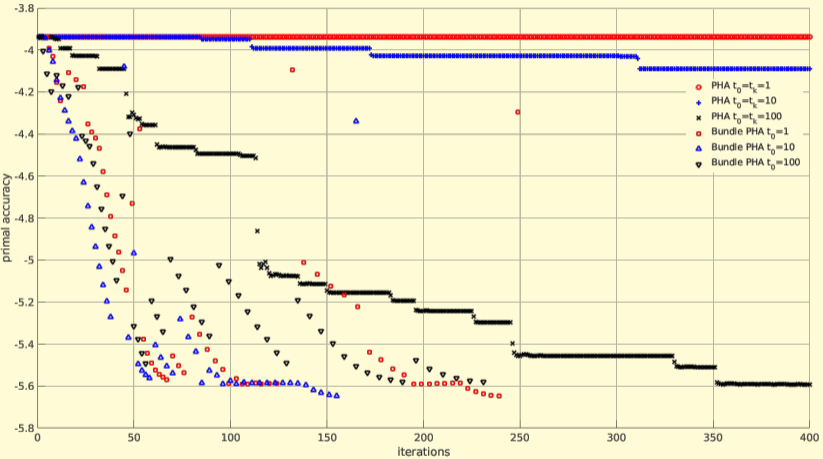
–Last generated serious iterate was optimal

–The null tail converges to last serious

–no rate

⇒ implementable stopping test!

# Goodies of Bundle PH - Performance



# To know more: related references



Free a

Home | Submit | Browse - | Search | Document

hal-03738298, version 1 | Preprints, Working Papers, ...

## A bundle-like progressive hedging algorithm

Felipe Atenas<sup>1</sup>, Claudia Sagastizábal<sup>2</sup> [Details](#)

- IMECC - Instituto de Matemática, Estatística e Computação Científica [Brésil]
- UNICAMP - Universidade Estadual de Campinas = University of Campinas

**Abstract :** For convex multistage programming problems, we propose a variant for the Progressive Hedging algorithm inspired from bundle methods. Like in the original algorithm, iterates are generated by first solving separate problems for each scenario, and then performing a projective step to ensure non-anticipativity. An additional test checks the quality of the approximation, splitting iterates into two subsequences, akin to the dichotomy between bundle serious and null steps. The method is shown to converge in both cases, and the convergence rate is linear for the serious subsequence. Our bundle-like approach endows the Progressive Hedging algorithm with an implementable stopping test. Moreover, it is possible to vary the augmentation parameter along iterations without impacting convergence. Such enhancements with respect to the original Progressive Hedging algorithm are obtained at the expense of the solution of additional subproblems at each iteration, one per scenario.

JoCA  
vol 30(2), 2023

## Optimization Online



Welcome | Repository | Submit | About | Help | My Eprints

Search ...

Search

[Log in](#)



Repository

## A unified analysis of descent sequences in weakly convex optimization, including convergence rates for bundle methods

Published: 2021/12/15, Updated: 2022/06/23

Felipe Atenas, Claudia Sagastizábal, Paulo J. S. Silva, Mikhail V. Solodov

Complementarity and Variational Inequalities, Nonlinear Optimization

bundle methods, descent methods, error bound, linear convergence, model-based methods, proximal descent, proximal gradient method, weak convexity

Short URL: <https://optimization-online.org/?p=18426>

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)

$$\left\{ \begin{array}{ll} \min_x & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in \mathcal{S}_j \quad \forall j \\ & \sum_j g_j(x_j) = 0 \end{array} \right.$$

## Bonus track

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

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

## Bonus track

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

$$\left\{ \begin{array}{ll} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in \mathcal{S}_j \quad \forall j \\ & g_j(x_j) + y_j = 0 \quad \forall j \\ & \sum_j y_j = 0 \quad \iff (y_1, \dots, y_j, \dots) \in \mathbf{N} \end{array} \right.$$

## Bonus track

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

$$\left\{ \begin{array}{ll} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in \mathcal{S}_j \quad \forall j \\ & g_j(x_j) + y_j = 0 \quad \forall j \\ & \sum_j y_j = 0 \quad \iff (y_1, \dots, y_j, \dots) \in \mathbf{N} \end{array} \right.$$

- ▶ 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)

$$\left\{ \begin{array}{ll} \min_{x,y} & \sum_j f_j(x_j) \\ \text{s.t.} & x_j \in \mathcal{S}_j \quad \forall j \\ & g_j(x_j) + y_j = 0 \quad \forall j \\ & \sum_j y_j = 0 \quad \iff (y_1, \dots, y_j, \dots) \in \mathbf{N} \end{array} \right.$$

- ▶ 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](https://doi.org/10.1007/978-3-642-46823-0_6))