Divide to conquer:

DECOMPOSITION METHODS

FOR ENERGY OPTIMIZATION

Claudia Sagastizábal

mailto:sagastiz@impa.br, http://www.impa.br/~sagastiz

21st ISMP

Berlin, August 2012

With thanks to:

AFOSR Grant FA9550-08-1-0370, NSF Grant DMS 0707205, and CNPq & Faperj from Brazil

Specific context: energy problems

Optimization helps in decision making:

Generation and dispatch problems: how to operate -daily, weekly, or yearly- a mix of power plants or an individual plant, taking into account specific technological characteristics

Transmission problems: how to deliver power through the electric network in a reliable manner

Expansion problems: how to decide which plants are to be built in a five, ten, or thirty years future, as well as their location and network connections.

Competitive market problems: how to understand competitive interaction between several agents (GenCo, DisCo, etc) seeking to maximize profit

Why bother in decomposing?

All the models exhibit some intrinsic separable structure:

– different technologies define feasible sets with different features

– operating costs are often given by sums

– a large system has different geographical regions

– some decisions can be taken along time steps

– uncertainty is often represented by sequential stochastic processes (time series)

Why bother in decomposing?

All the models exhibit some intrinsic separable structure:

– different technologies define feasible sets with different features

– operating costs are often given by sums

– a large system has different geographical regions

– some decisions can be taken along time steps

– uncertainty is often represented by sequential stochastic processes (time series)

and, without decomposing, the problem cannot be solved

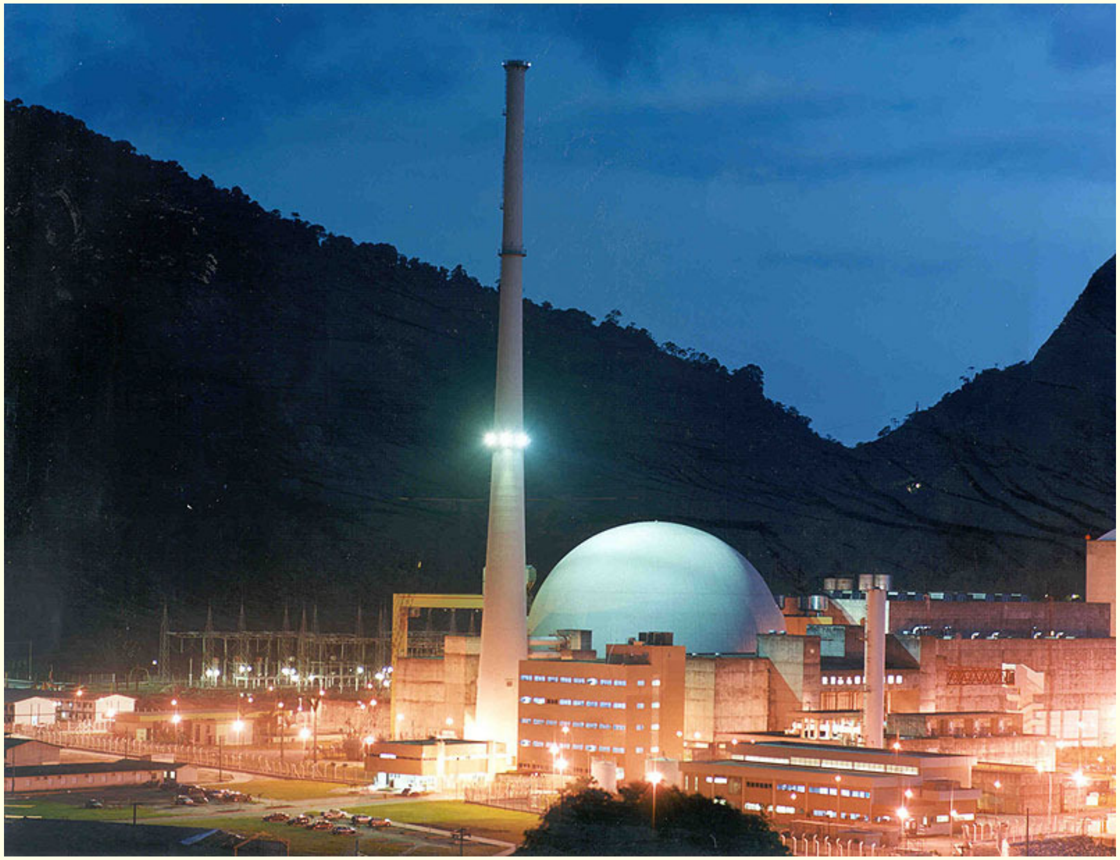
in the desired time frame, or with the desired accuracy

Algorithms need to be fast and good

Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies



Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies

Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies

Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies

Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies

– There is a network, to be operated in a reliable manner

– Energy cannot be stored, but water can

– Availability of hydropower is limited and uncertain

Specific features of energy systems

For a given power system, perform the **optimal management of the resources**, given that

– Power can be generated by different technologies

– There is a network, to be operated in a reliable manner

– Energy cannot be stored, but water can

– Availability of hydropower is limited and uncertain

– Same for eolic

Specific features of energy systems

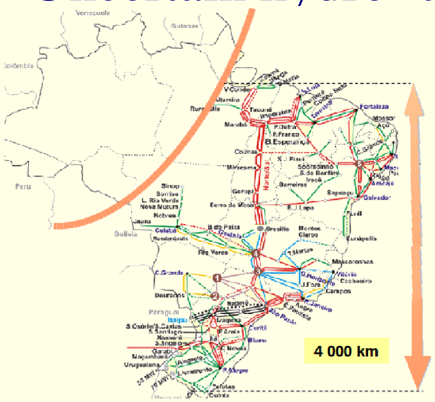
For a given power system, perform the **optimal management of the resources**, given that

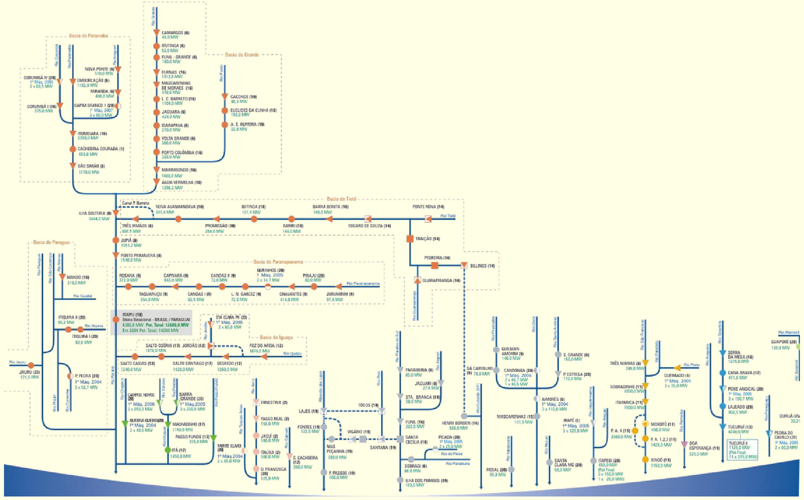
– Power can be generated by different technologies

– There is a network, to be operated in a reliable manner

– Energy cannot be stored, but water can

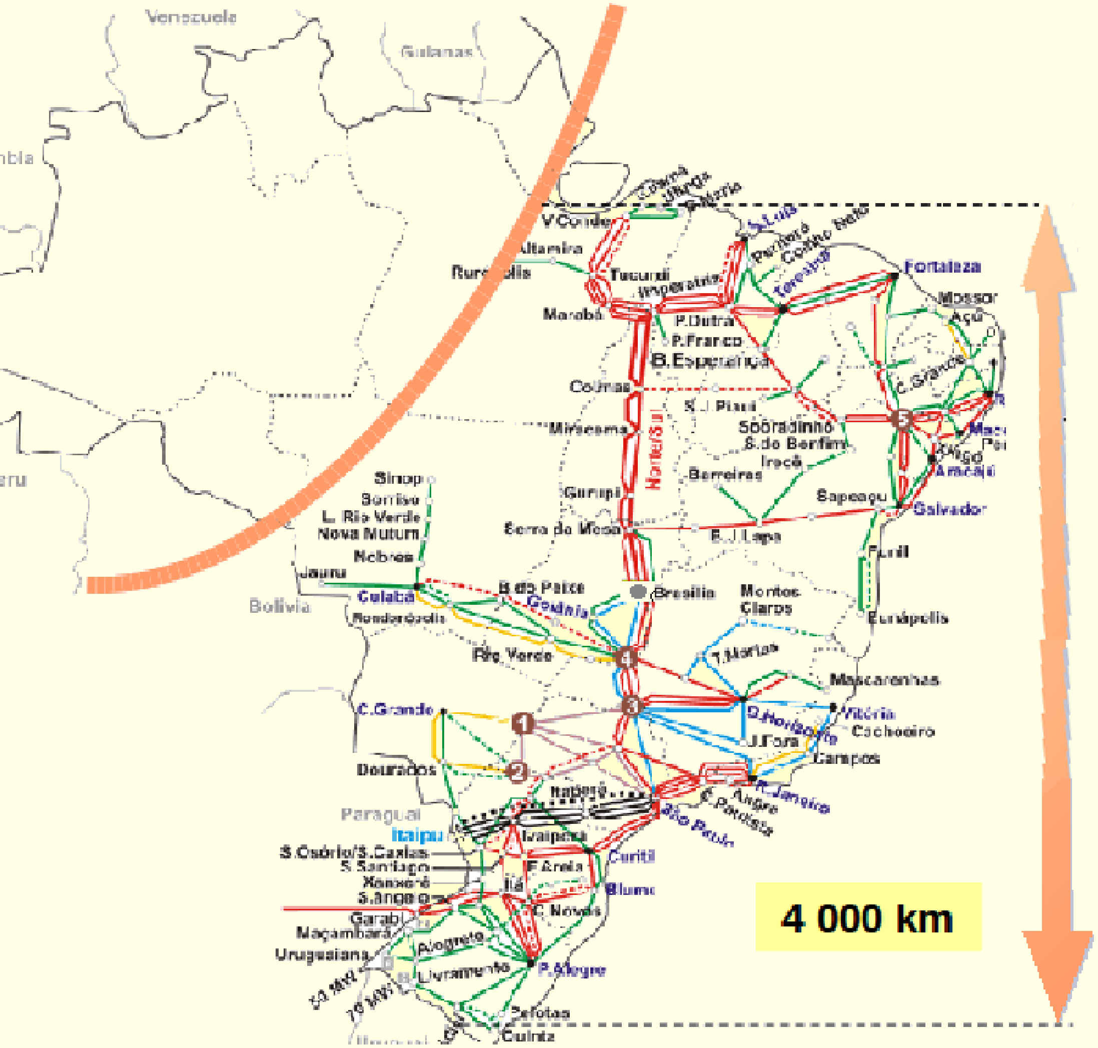
– Uncertain hydro- and wind power

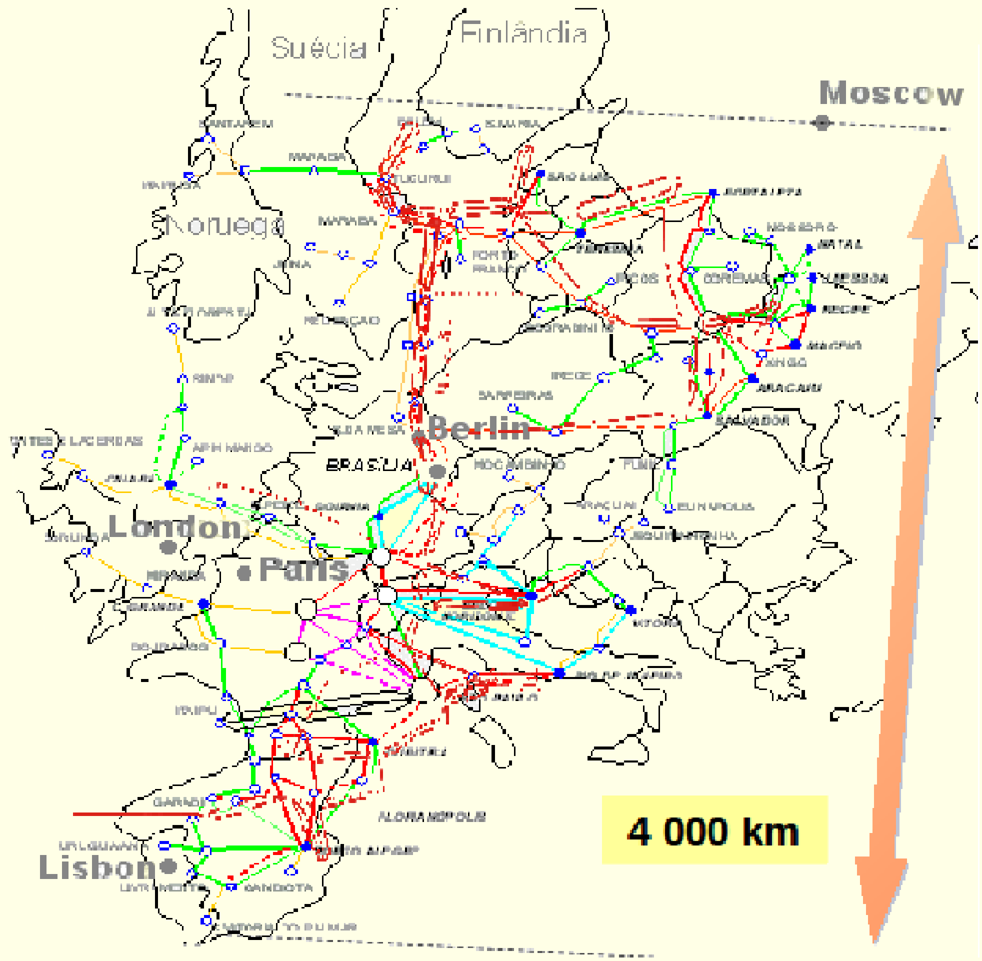




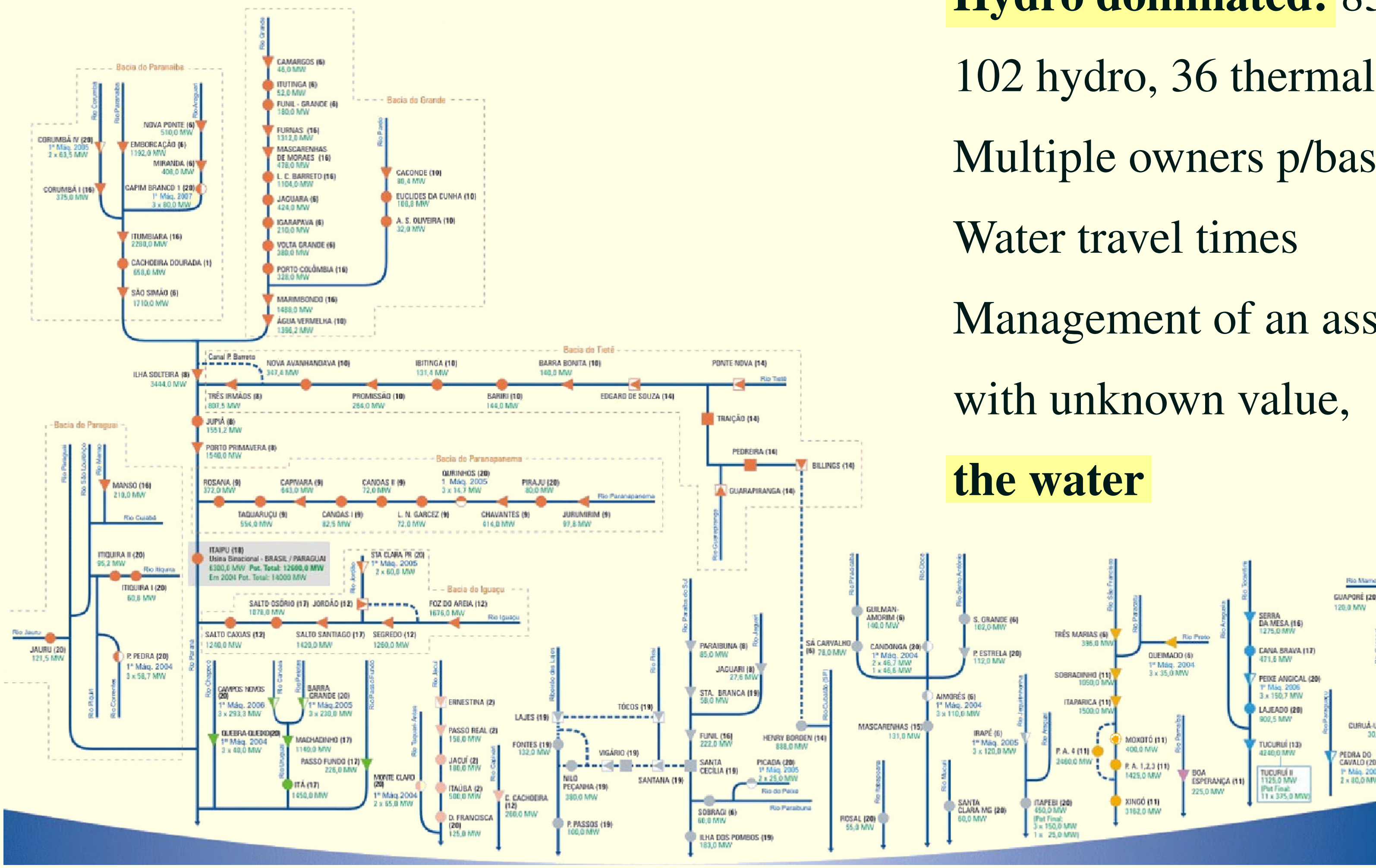
Network and system can be **large**

Network can have continental dimensions





Huge hydrothermal system



Hydro dominated: 83%

102 hydro, 36 thermal (> 200 total)

Multiple owners p/basin

Water travel times

Management of an asset

with unknown value,

the water

Output with high socio-economic impact: **electricity prices**

DIVIDING TO CONQUER:

Lagrange, Benders,

Bundle,

and friends

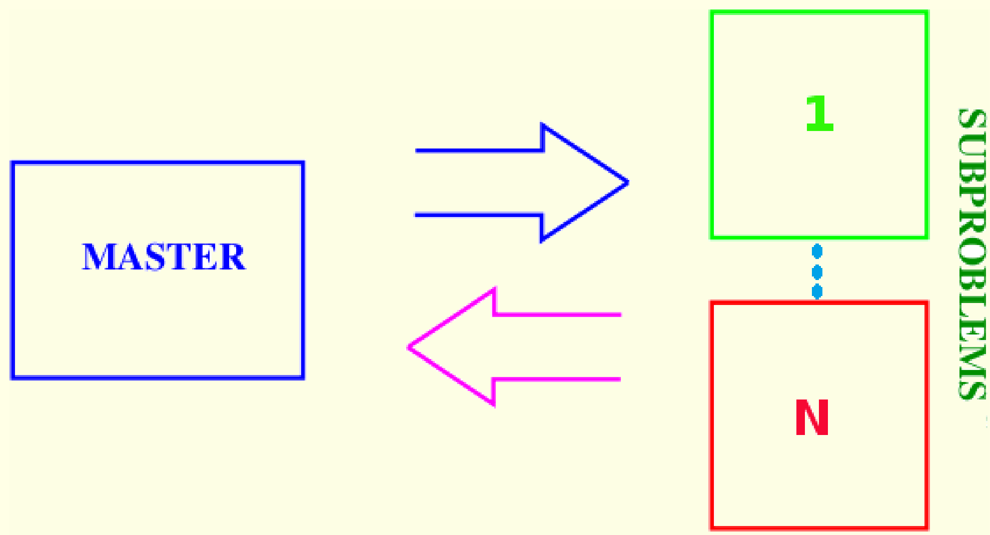
(a few representative examples of the power of decomposition)

Energy problems have intrinsic structure, made explicit by some **decomposition** method

by Lagrangian relaxation

by Benders decomposition

Principle: if a problem is difficult to solve directly,
solve instead a sequence of easier subproblems.

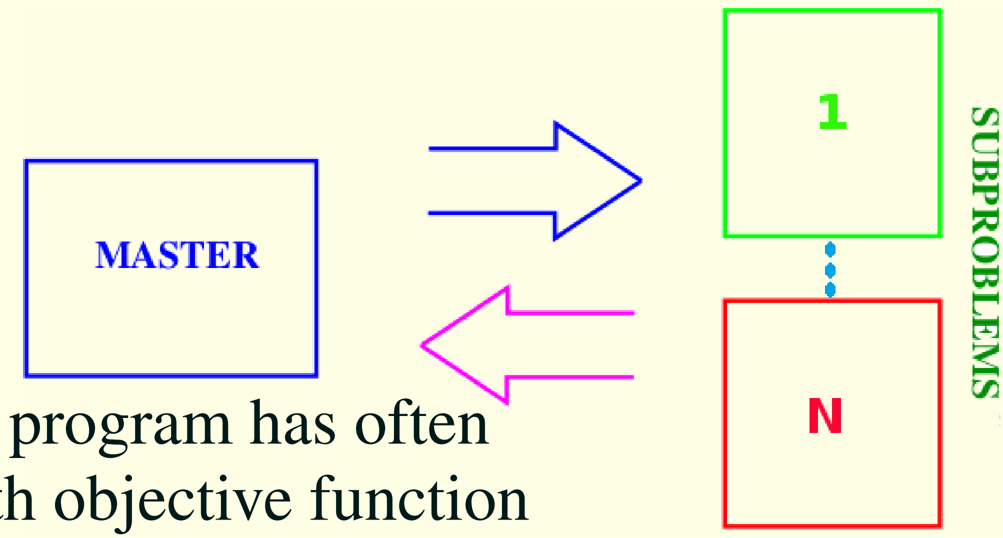


Energy problems have intrinsic structure, made explicit by some **decomposition** method

by Lagrangian relaxation

by Benders decomposition

Principle: if a problem is difficult to solve directly,
solve instead a sequence of easier subproblems.



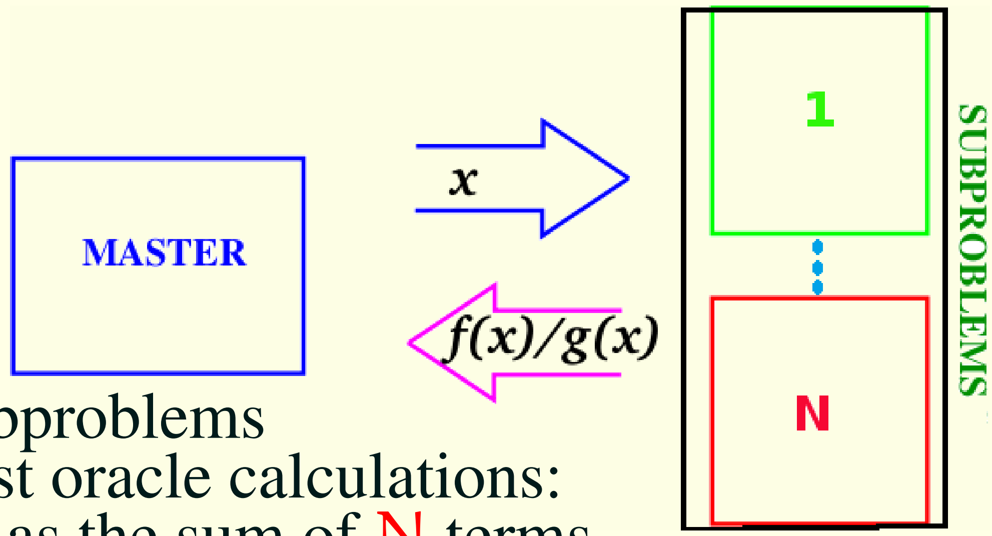
The master program has often a nonsmooth objective function

Energy problems have intrinsic structure, made explicit by some **decomposition** method

by Lagrangian relaxation

by Benders decomposition

Principle: if a problem is difficult to solve directly,
solve instead a sequence of easier subproblems.



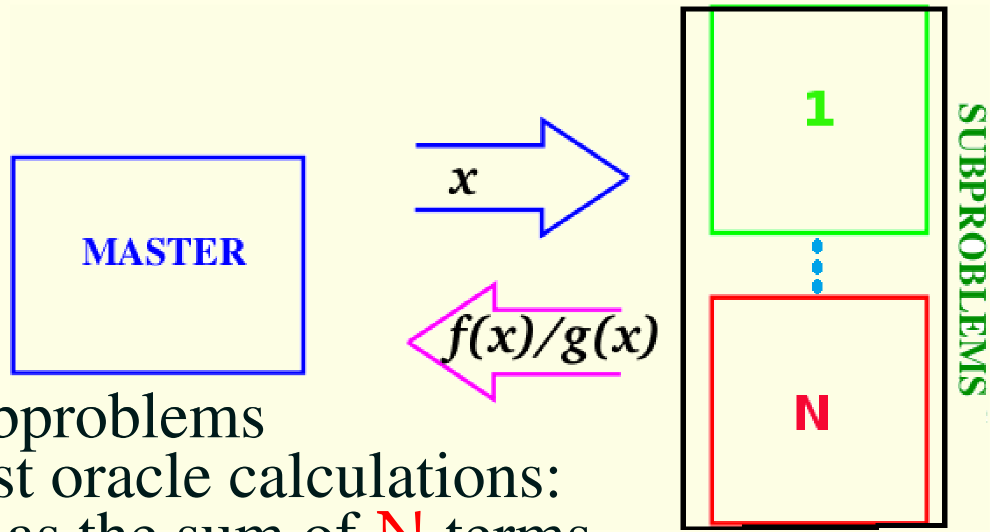
Separate subproblems
allow for fast oracle calculations:
f/g defined as the sum of N terms

Energy problems have intrinsic structure, made explicit by some **decomposition** method

by Lagrangian relaxation
by Benders decomposition

$\}$ **$\neq$ oracles**

Principle: if a problem is difficult to solve directly,
solve instead a sequence of easier subproblems.



Separate subproblems
allow for fast oracle calculations:
f/g defined as the sum of N terms

Let's start with Lagrange: energy management

- $p^j = (p^j_1, \ldots, p^j_T)$ j-th power plant generation
- $p^j \in \mathcal{P}^j$ operational constraints
- $\mathcal{C}^j(p^j)$ generation/operation cost
- $\sum_j p^j = d = (d_1, \ldots, d_T)$ demand satisfaction

GOAL:

$$
\texttt{(primal)} \quad \left\{ \begin{array}{ll} \min & \displaystyle\sum_j \mathcal{C}^j(p^j) \\ & p \in \mathcal{P} = \displaystyle\prod_j \mathcal{P}^j \\ & \boxed{\displaystyle\sum_j g^j(p^j) = d} \end{array} \right. \quad \boxed{\leftarrow x}
$$

$$
\texttt{(primal)} \quad \begin{cases} \min & \sum_j \mathcal{C}^j(p^j) \\ & p \in \mathcal{P} \\ & \boxed{\sum_j g^j(p^j) = d} \quad \boxed{\leftarrow x} \end{cases}
$$

exhibits separable structure after dualization

Key observation: $\min_p$ **and** $\sum_j$ **can be exchanged**

$$
\texttt{(dual)} \quad \max_x \left(-\langle x, d \rangle \; + \begin{cases} \min\limits_p & \sum\limits_{\mathbf{j}} \mathcal{C}^{\mathbf{j}}(p^j) + \left\langle x, \sum_j g^j(p^j) \right\rangle \\ p & \in \prod_{\mathbf{j}} \mathcal{P}^{\mathbf{j}} \end{cases} \right)
$$

$$
\texttt{(primal)} \quad \left\{ \begin{array}{ll} \min & \sum_j \mathcal{C}^j(p^j) \\ & p \in \mathcal{P} \\ & \boxed{\sum_j g^j(p^j) = d} \quad \boxed{\leftarrow x} \end{array} \right.
$$

exhibits separable structure after dualization

Key observation: $\min_p$ **and** $\sum_j$ **can be exchanged**

$$
\texttt{(dual)} \quad \max_x \left(-\langle x, d\rangle + \sum_j \left\{ \begin{array}{l} \min_{\mathbf{p^j}} \mathcal{C}^{\mathbf{j}}(p^j) + \left\langle x, g^j(p^j) \right\rangle \\ p^j \in \mathcal{P}^{\mathbf{j}} \end{array} \right. \right)
$$

$$
\texttt{(primal)} \quad \begin{cases} \min & \displaystyle\sum_j \mathcal{C}^j(p^j) \\ & p \in \mathcal{P} \\ & \boxed{\displaystyle\sum_j g^j(p^j) = d} \end{cases} \quad \boxed{\leftarrow x}
$$

exhibits separable structure after dualization

Key observation: $\min_p$ **and** $\sum_j$ **can be exchanged**

$$
\texttt{(dual)} \quad \max_x \left(-\langle x, d\rangle \; - \; \sum_j f^j(x) \right)
$$

$$
-f^j(x) := \begin{cases} \min & \mathcal{C}^j(p^j) - \left\langle x, g^j(p^j) \right\rangle \\ & p^j \in \mathcal{P}^j \end{cases}
$$

Energy management problems

Typically, evaluating

$$-f^j(x) := \begin{cases} \min & \mathcal{C}^j(p^j) - \left\langle x, g^j(p^j) \right\rangle \\ & p^j \in \mathcal{P}^j \end{cases}$$

corresponds to

local subproblems, related to one power plant, requiring sometimes heavy calculations

Energy management problems

Typically, evaluating

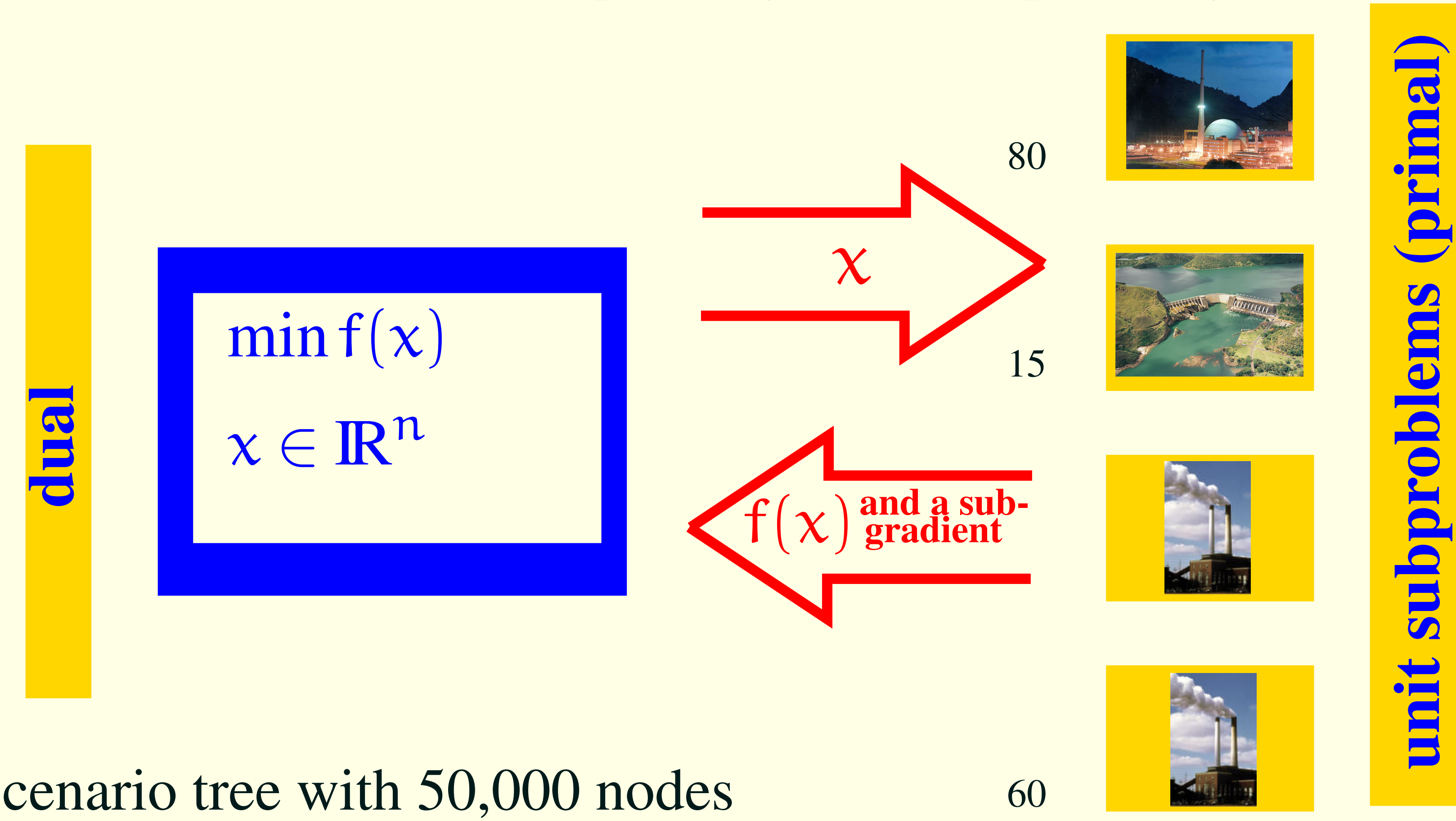
$$-f^j(x) := \begin{cases} \min & \mathcal{C}^j(p^j) - \left\langle x, g^j(p^j) \right\rangle \\ & p^j \in \mathcal{P}^j \end{cases}$$

corresponds to

local subproblems, related to one power plant, requiring sometimes heavy calculations

One subgradient for free: $g^j(p^j(x))$ once a solution $p^j(x)$ is available

Often, most of the CPU time is spent in the oracle calculations. For mid-term power generation planning:

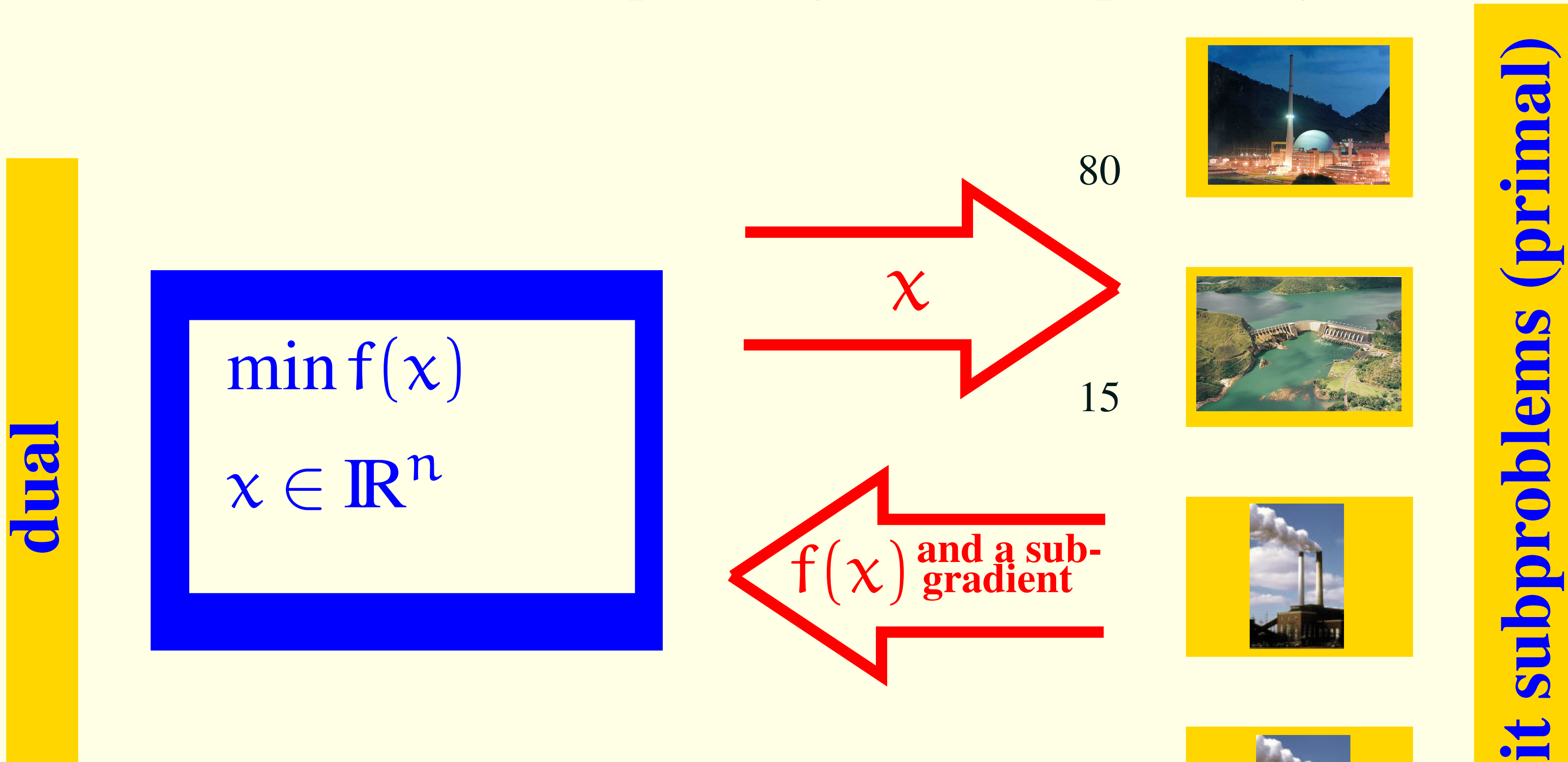


Scenario tree with 50,000 nodes

Nuclear subproblems are LPs with 100,000 variables

and 300,000 constraints, consuming **99%** total running time

Often, most of the CPU time is spent in the oracle calculations. For mid-term power generation planning:



Can we skip/solve approximately

nuclear subproblems,

consuming LESS running time without losing accuracy?

Benders decomposition example

Useful for decomposing expansion problems modelled as 2 stage programs:

$$p_t^j \in \mathcal{P}^j \quad \Longleftrightarrow \quad p_t^j \in \left[p_{\min}^j \quad , p_{\max}^j \quad \right]$$

Benders decomposition example

Useful for decomposing expansion problems modelled as 2 stage programs:

$$p_t^j \in \mathcal{P}^j \quad \iff \quad p_t^j \in \left[p_{\min}^j x_t^j, p_{\max}^j x_t^j\right] \text{ for } x_t^j \in [0,1]$$

Constraint active only if the j-th power plant is built at time t.

Benders decomposition example

Useful for decomposing expansion problems modelled as 2 stage programs:

$$p_t^j \in \mathcal{P}^j \quad \Longleftrightarrow \quad p_t^j \in \left[p_{\min}^j x_t^j, p_{\max}^j x_t^j\right] \text{ for } x_t^j \in [0,1]$$

Constraint active only if the j-th power plant is built at time t.

+ uncertainty, for example represented by a tree with N scenarios

$$\mathbf{SP_2} \quad \begin{aligned} &\min_{(x,p)} c^\top x + \mathcal{C}(p) \\ &x \in \mathcal{X}, p \in \mathcal{P} \\ &\mathbf{T}x + Wp = \mathbf{h} \end{aligned}$$

Benders decomposition example

Useful for decomposing expansion problems modelled as 2 stage programs:

$$p_t^j \in \mathcal{P}^j \quad \Longleftrightarrow \quad p_t^j \in \left[p_{\min}^j x_t^j, p_{\max}^j x_t^j\right] \text{ for } x_t^j \in [0,1]$$

Constraint active only if the j-th power plant is built at time t.

+ uncertainty, for example represented by a tree with N scenarios

$$\mathbf{SP_2} \quad \left.\begin{array}{l} \min\limits_{(x,p)} c^\top x + \boldsymbol{\mathcal{C}}(p) \\ x \in \boldsymbol{\mathcal{X}}, p \in \boldsymbol{\mathcal{P}} \\ \mathbf{T}x + Wp = \mathbf{h} \end{array}\right\} \equiv \left\{\begin{array}{l} \min\limits_{x} c^\top x + \mathbb{E}[Q(x;\xi)] \\ x \in \boldsymbol{\mathcal{X}} \\ \text{for } Q(x;\xi) = \left\{\begin{array}{l} \min\limits_{p \in \boldsymbol{\mathcal{P}}} \boldsymbol{\mathcal{C}}(p) \\ Wp = h - Tx \end{array}\right. \end{array}\right.$$

Benders decomposition example

Useful for decomposing expansion problems modelled as 2 stage programs:

$$p_t^j \in \mathcal{P}^j \quad \Longleftrightarrow \quad p_t^j \in \left[p_{min}^j x_t^j, p_{max}^j x_t^j\right] \text{ for } x_t^j \in \{0,1\}$$

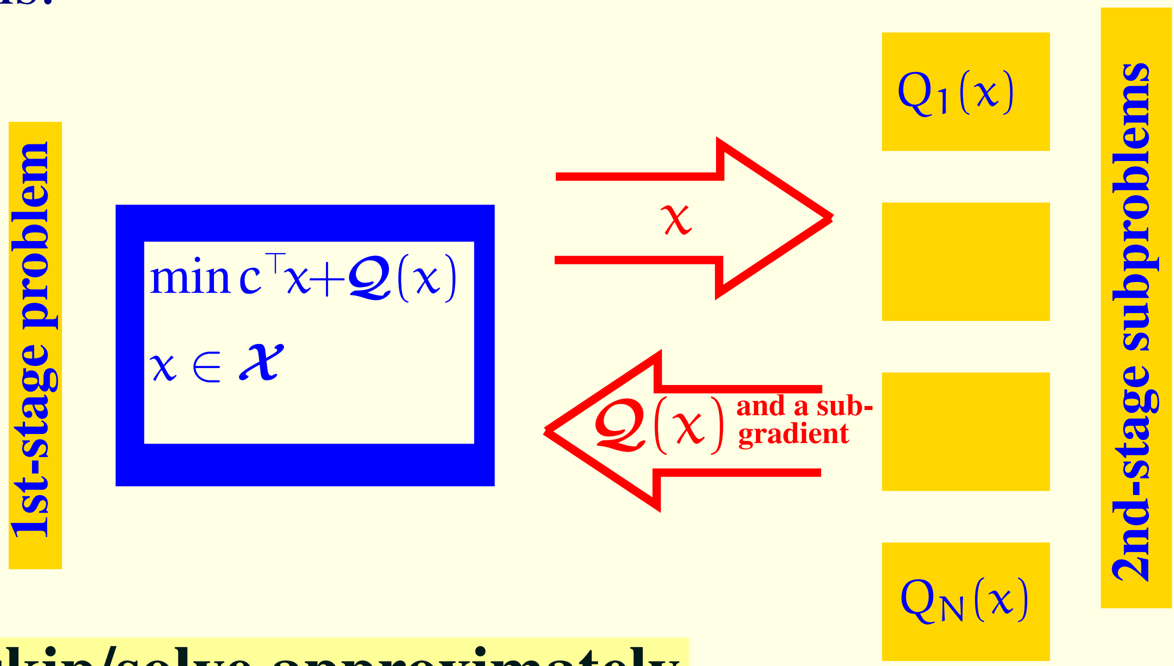
Constraint active only if the j-th power plant is built at time t.

+ uncertainty, for example represented by a tree with N scenarios

$$\textbf{SP}_2 \quad \left.\begin{array}{l} \min\limits_{(x,p)} c^\top x + \mathcal{C}(p) \\ x \in \mathcal{X}, p \in \mathcal{P} \\ \mathbf{T}x + Wp = \mathbf{h} \end{array}\right\} \equiv \left\{\begin{array}{l} \min\limits_{x} c^\top x + \mathbb{E}[Q(x;\xi)] \\ x \in \mathcal{X} \\ \text{for } Q(x;\xi) = \left\{\begin{array}{l} \min\limits_{p \in \mathcal{P}} \mathcal{C}(p) \\ Wp = h - Tx \end{array}\right. \end{array}\right.$$

Once again, one subgradient for free for $f(x) = c^\top x + \mathcal{Q}(x)$
where $\mathcal{Q}(x) = \mathbb{E}[Q(x;\xi)]$ **is computed as the sum of** N **terms**

Again, most of the CPU time is spent in the oracle calculations.

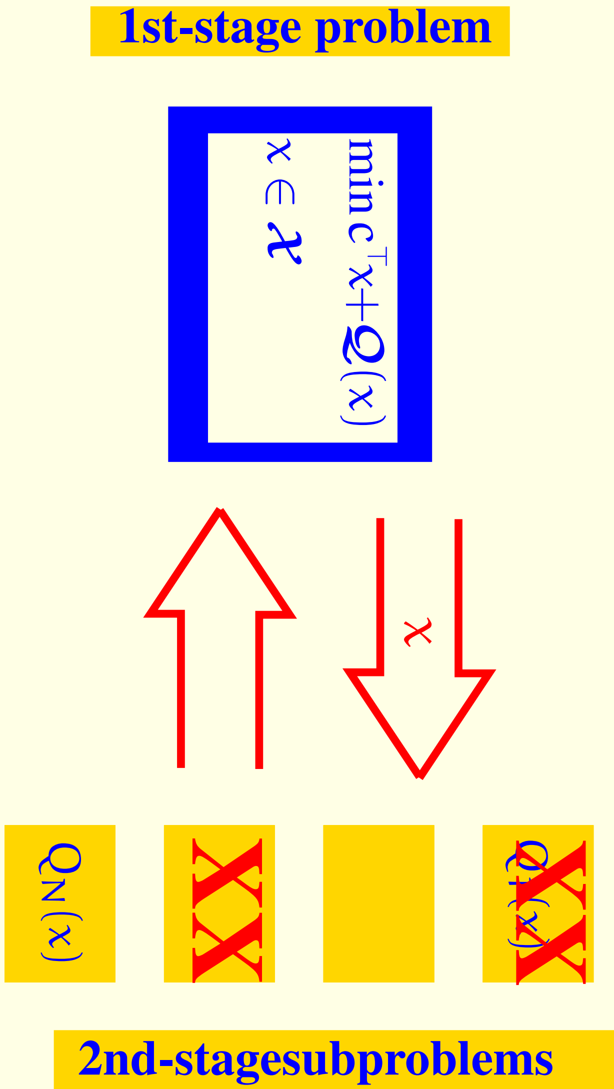


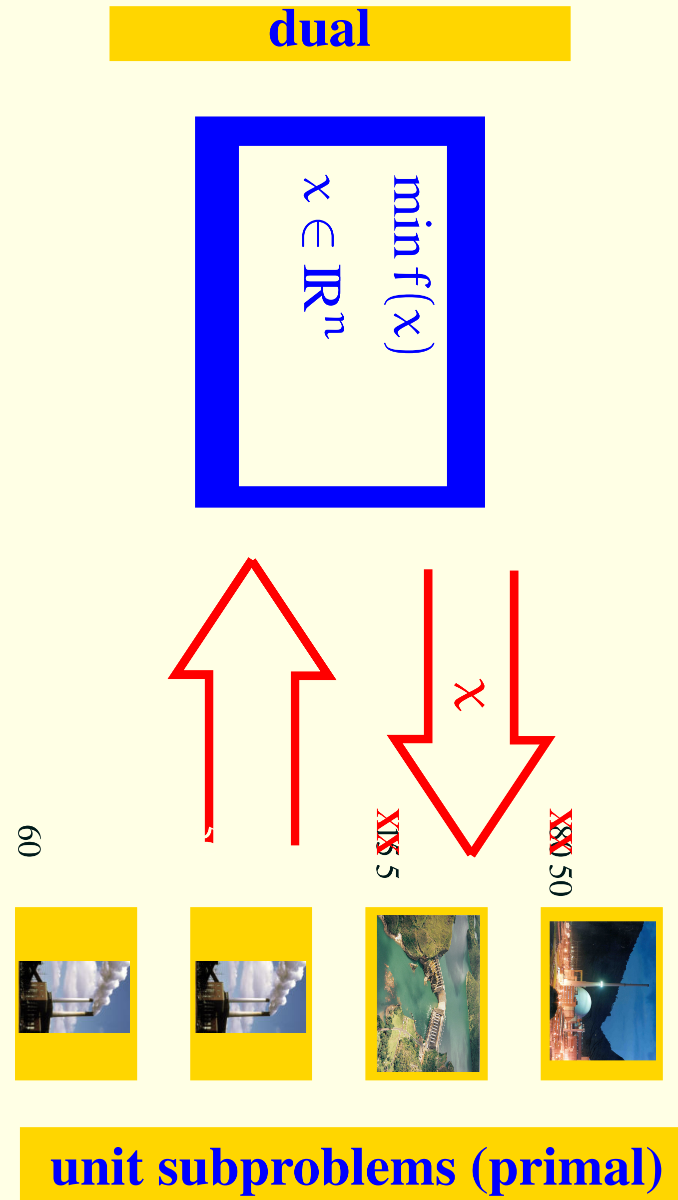
Can we skip/solve approximately

some $\xi_i = (q_i, T_i, h_i)$ instances,

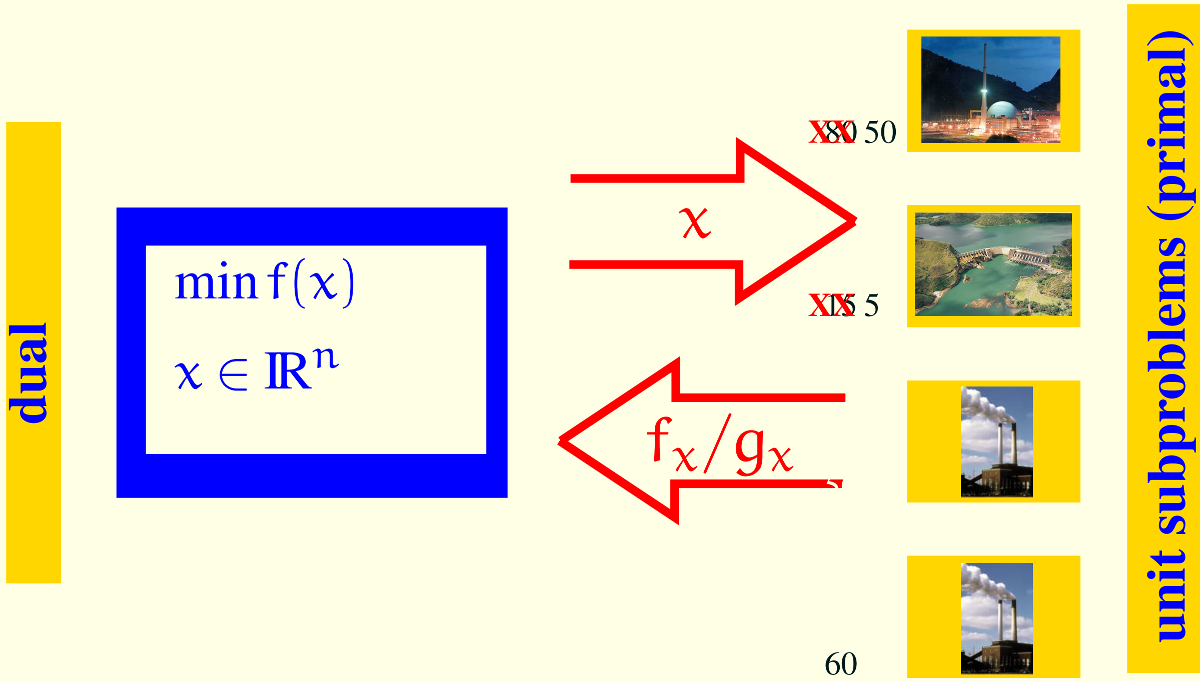
consuming LESS running time without losing accuracy?

Can we adapt the oracle response to the solver needs?

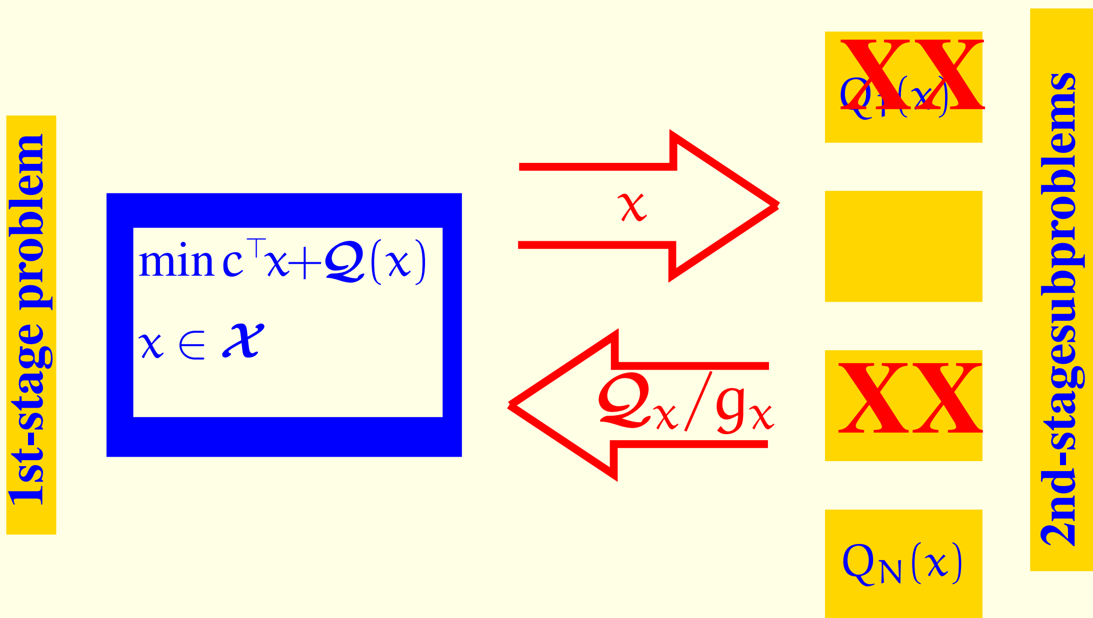




Can we adapt the oracle response to the solver needs?



now the oracle returns INEXACT values



Can we adapt the oracle response to the solver needs? **YES!**

with a NSO method capable of handling

oracles with on-demand accuracy

Can we adapt the oracle response to the solver needs? **YES!**

with a NSO method capable of handling

oracles with on-demand accuracy

DIVIDING TO CONQUER:

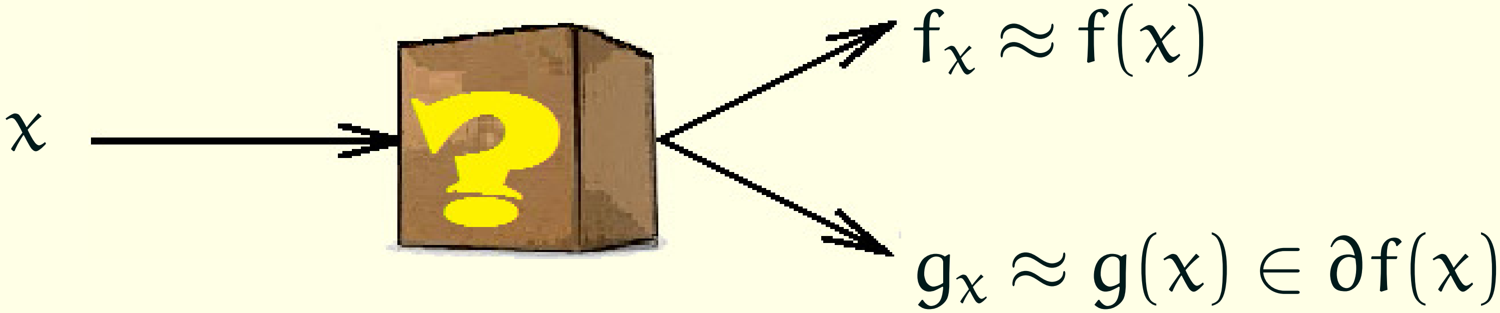
Lagrange, Benders,

Bundle,

and friends

Can we adapt the oracle response to the solver needs? **YES!**

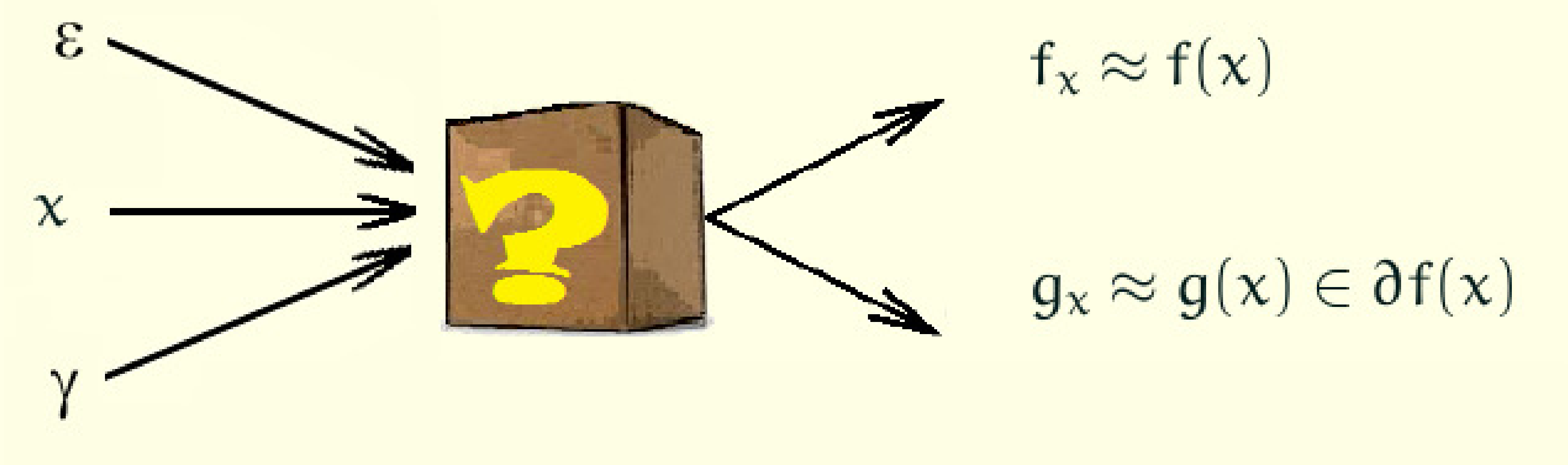
with a NSO method capable of handling **oracles with on-demand accuracy** created over noisy black-boxes



when we have the ability of computing f_x/g_x with more or less accuracy

Oracle with on-demand accuracy

This is a noisy black box that gets additional input:



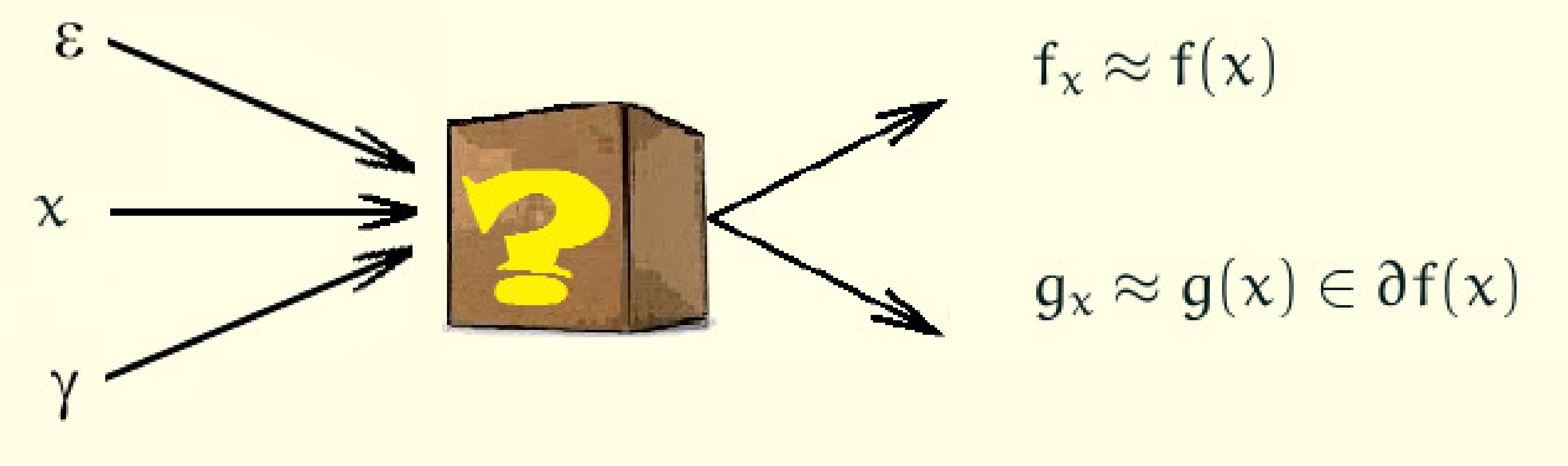
an **error bound ε and a descent target γ** such that

$$\left.\begin{aligned} f_x &= f(x) - \eta(x) \\ g_x &\in \partial_{\eta(x)} f(x) \end{aligned}\right\} \quad \text{for all } x, \text{ with } \eta(x) \geq 0$$

$$\eta(x) \leq \varepsilon \qquad \text{if } x \text{ gave enough descent: } f_x \leq \gamma$$

Oracle with on-demand accuracy

This is a noisy black box that gets additional input:



an **error bound ε and a descent target γ** such that

$$\left.\begin{array}{l} f_x = f(x) - \eta(x) \\ g_x \in \partial_{\eta(x)} f(x) \end{array}\right\} \quad \text{for all } x \text{, with } \eta(x) \geq 0 \ \textbf{unknown}$$

$$\eta(x) \leq \varepsilon \qquad \text{if } x \text{ gave enough descent: } f_x \leq \gamma$$

Oracle with on-demand accuracy: versatility

$$\left.\begin{array}{l} f_x = f(x) - \eta(x) \\ g_x \in \partial_{\eta(x)} f(x) \end{array}\right\} \quad \text{for all } x \text{, with } \eta(x) \geq 0$$

$$\eta(x) \leq \varepsilon \qquad \text{if } x \text{ gave enough descent: } f_x \leq \gamma$$

We control both ε and γ, which can vary with x:

$\varepsilon_x = 0$ and $\gamma_x = +\infty$: an exact oracle

$\varepsilon_x \to 0$ and $\gamma_x = +\infty$: an asymptotically exact oracle [ZakPhilpRyan01, Fabian00,EmielSagastiz10]

$\varepsilon_x = 0$ with finite γ_x: the partly inexact oracle [Kiw09]

$\varepsilon_x > 0$ unknown, but bounded, with $\gamma_x = +\infty$: the inexact oracle [Hint01,Sol03,Kiw06,OlivSagastizScheim11]

Oracle with on-demand accuracy: versatility

$$\left.\begin{array}{l} f_x = f(x) - \eta(x) \\ g_x \in \partial_{\eta(x)} f(x) \end{array}\right\} \quad \text{for all } x \text{, with } \eta(x) \geq 0$$

$$\eta(x) \leq \varepsilon \qquad \text{if } x \text{ gave enough descent: } f_x \leq \gamma$$

We control both ε and γ, which can vary with x:

$\varepsilon_x = 0$ and $\gamma_x = +\infty$: an exact oracle

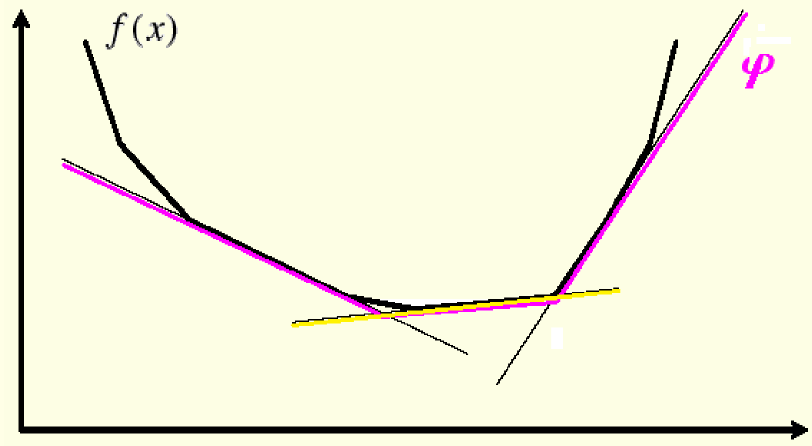
$\varepsilon_x \to 0$ and $\gamma_x = +\infty$: an asymptotically exact oracle [ZakPhilpRyan01, Fabian00,EmielSagastiz10]

$\varepsilon_x = 0$ with finite γ_x: the partly inexact oracle [Kiw09]

$\varepsilon_x > 0$ unknown, but bounded, with $\gamma_x = +\infty$: the inexact oracle [Hint01,Sol03,Kiw06,OlivSagastizScheim11]

NEW $\varepsilon_x \to 0$ and $\gamma_x < +\infty$: a partly asymptotically exact oracle [OlivSagastiz12]

What does this mean for the NSO problem?

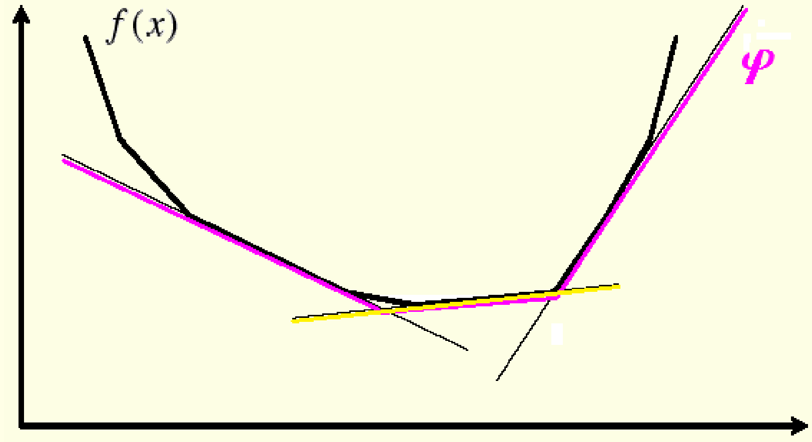


Oracle information defines **pieces**

$f(x) + g(x)^\top(\cdot - x)$

that put together create a **model φ** of f, used to define iterates.

What does this mean for the NSO problem?



Oracle information defines **pieces**

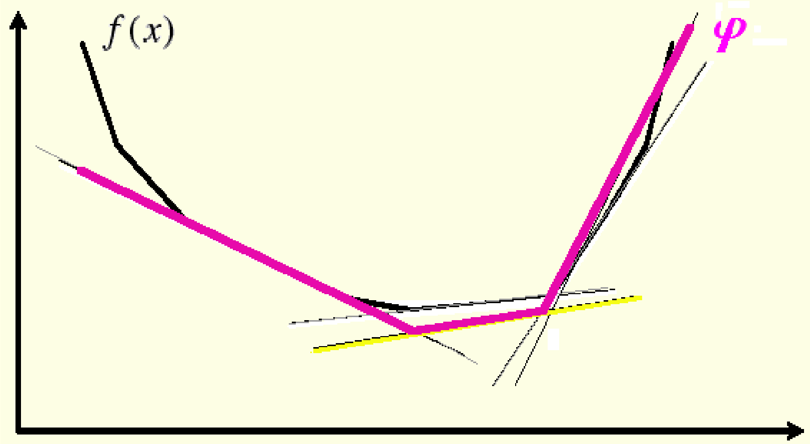
$$f(x) + g(x)^\top(\cdot - x)$$

that put together create a **model φ** of f, used to define iterates.

now linearizations may be inexact:

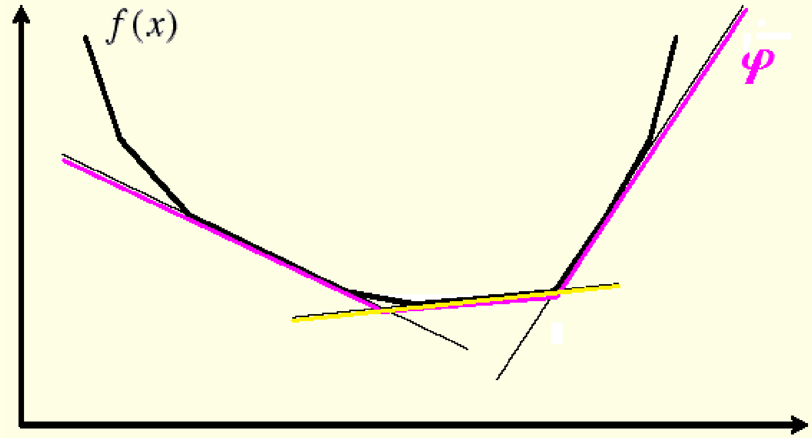
x^j  $f^j = f_{x^j}$, $g^j = g_{x^j}$

$$\Longrightarrow \varphi^i(x) = \max_{j \le i}\left\{f^j + g^{j\top}(x - x^j)\right\}$$



and the model may be "wrong"

What does this mean for the NSO problem?



Oracle information defines **pieces**

$f(x)+g(x)^\top(\cdot - x)$

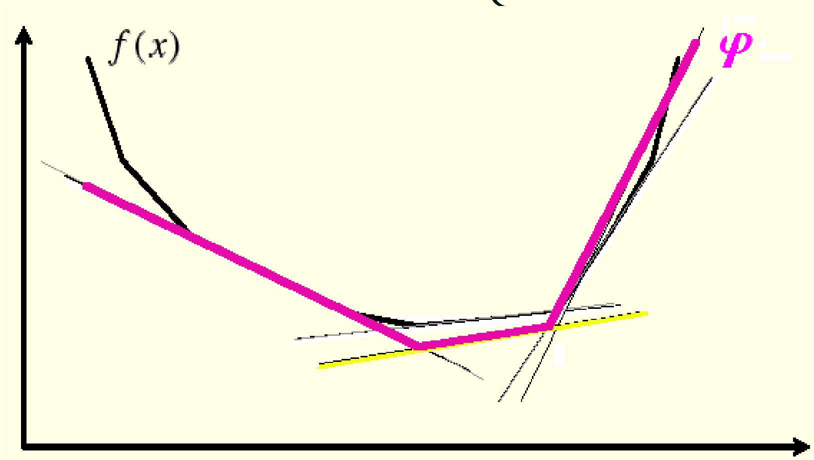
that put together create a **model φ** of f, used to define iterates.

now linearizations may be inexact:

x^j  $f^j = f_{x^j}$

$g^j = g_{x^j}$

$$\Longrightarrow \varphi^i(x) = \max_{j\le i}\left\{f^j + {g^j}^\top(x - x^j)\right\}$$

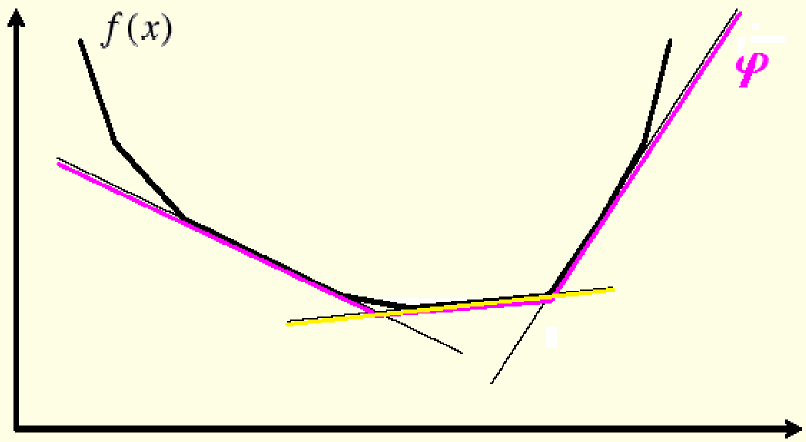


If too wrong:

noise needs to be attenuated

and the model may be “wrong”

What does this mean for the NSO problem?



Oracle information defines **pieces**

$$f(x) + g(x)^\top(\cdot - x)$$

that put together create a **model φ** of f, used to define iterates.

now linearizations may be inexact:

ε^j

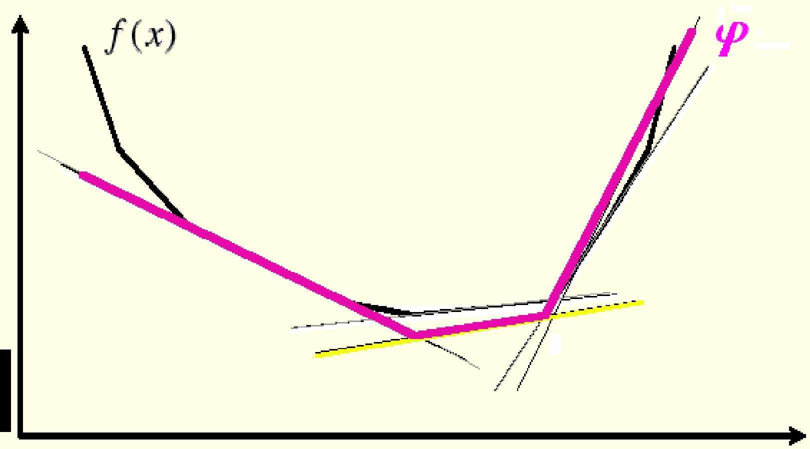
x^j

γ^j

$f^j = f_{x^j}$

$g^j = g_{x^j}$

$$\Longrightarrow \varphi^i(x) = \max_{j \le i}\left\{f^j + g^{j\top}(x - x^j)\right\}$$



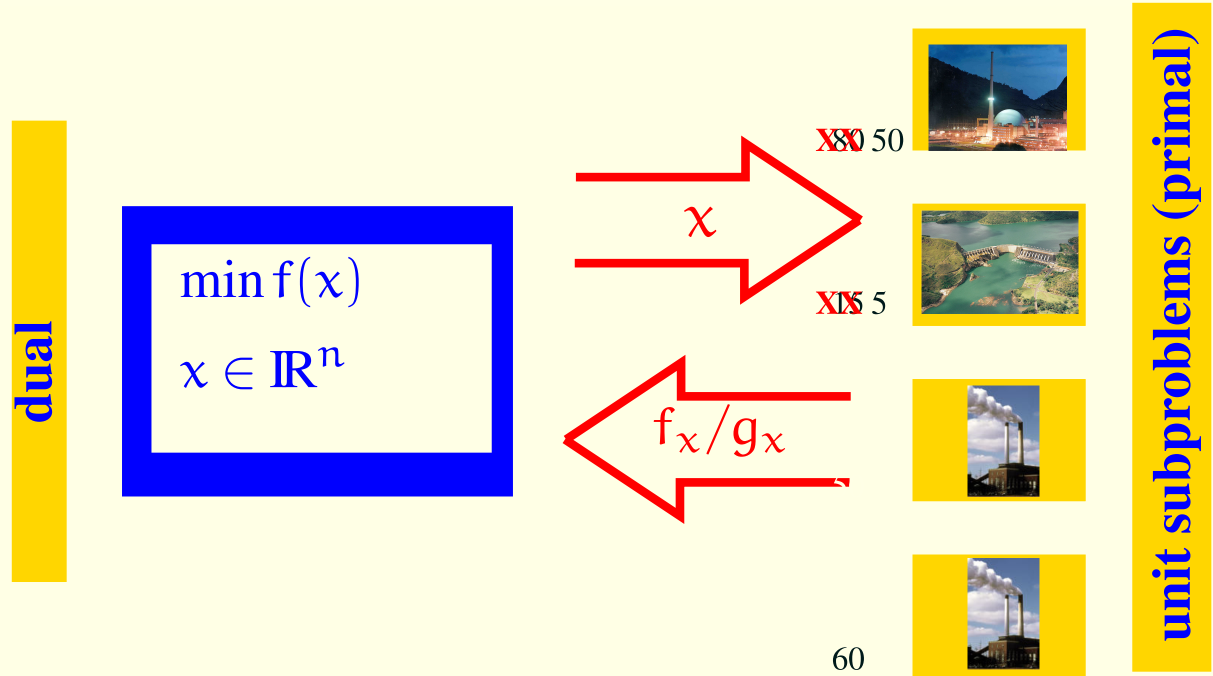
If too wrong:

noise needs to be attenuated

by exploiting the on-demand accuracy

feature of the oracle (joint work with W. Oliveira)

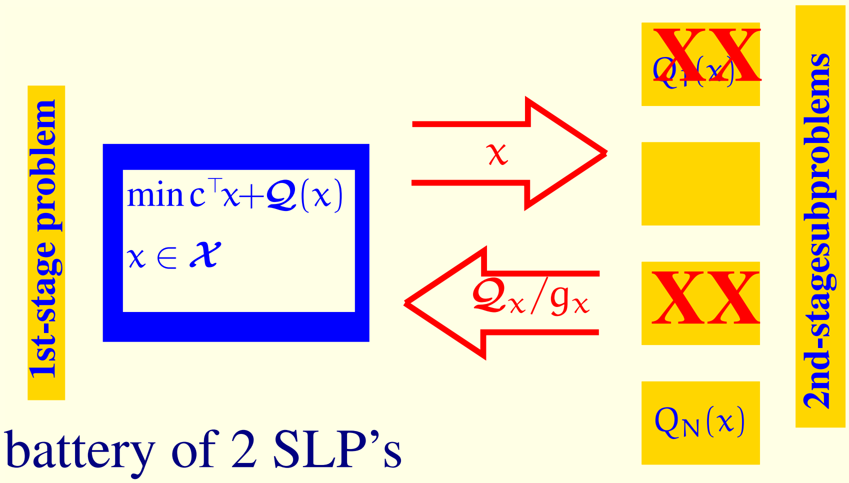
A glimpse of numerical results: Lagrange



for mid-term power planning problems (French mix)

25% less CPU time, same accuracy in the dual variable

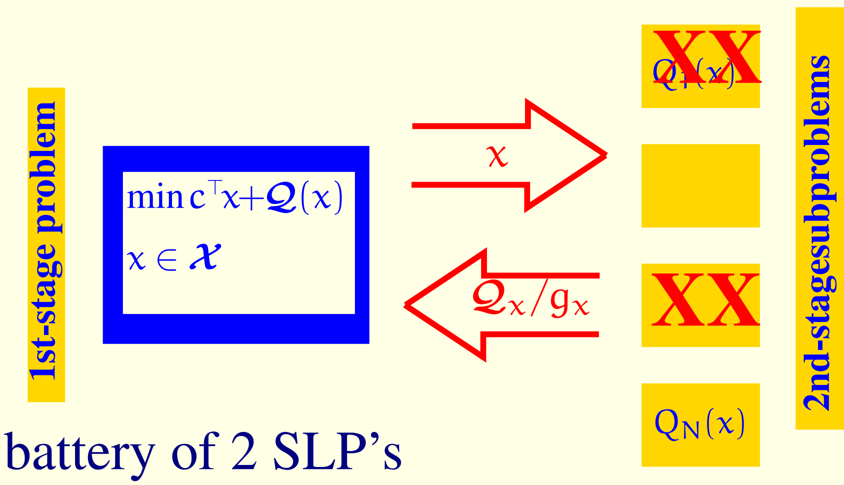
A glimpse of numerical results: Benders



For a large battery of 2 SLP's
versatility pays off:

Solver	% CPU time reduction
exact- cutting-Planes (L-shaped)	0
exact-ℓ [LNN95,Kiw95]	30
ℓ-Asymp. exact [Fab00]	53
ℓ-Partly asymp. exact	**72**

A glimpse of numerical results: Benders



asymptotically exact
oracle calculations
at serious steps only
reduce substantially
the computational effort

For a large battery of 2 SLP's
versatility pays off:

Solver	% CPU time reduction
exact- cutting-Planes (L-shaped)	0
exact-ℓ [LNN95,Kiw95]	30
ℓ-Asymp. exact [Fab00]	53
ℓ-Partly asymp. exact	**72**

DIVIDING TO CONQUER:

Lagrange, Benders,

Bundle,

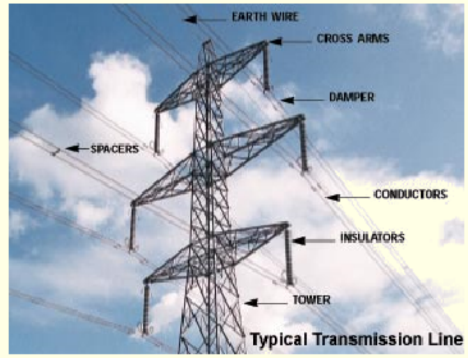
and friends:

the market

(a few representative examples of the power of decomposition)

Context

Generation



Transmission

Distribution

Until the 90's: a regulated monopoly

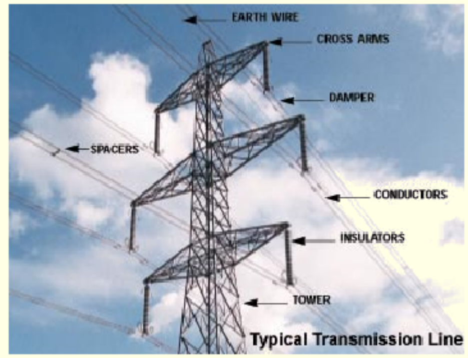
Rationale: 1 big firm brings economies of scale

Criticism: lack of incentive to innovate and to keep total capital/operations cost at its minimum

Note: Large transmission network connecting multiple G owners makes competition possible

Context

Generation



Transmission

Distribution

Restructuration: G competitive +TD regulated monopoly

Rationale: fierce competition of G provides higher incentive to minimize costs and, arguably, innovation

Criticism: it is not clear how fierce competition is···

This makes important to understand competitive interaction between several G firms seeking to maximize profit, taking into account unique aspects of electricity: not storable, yet supply needs to meet demand, energy needs to be transmitted from G plants to consumers, etc

A game-theoretical model

– $p^j = (p^j_1, \ldots, p^j_T)$ j-th power plant generation

– $p^j \in \mathcal{P}^j$ operational constraints

– $\mathcal{C}^j(p^j)$ generation/operation cost

GOAL:

– *G's take unilateral decisions, behave competitively, and want to recover fixed costs in the long term (including capital remuneration).*

– *A representative of the consumers, the ISO, focuses on the benefits of consumption, seeking a price that matches supply and demand, "keeping prices low".*

A game-theoretical model

– $p^j = (p^j_1, \ldots, p^j_T)$ j-th power plant generation

– $p^j \in \mathcal{P}^j$ operational constraints

– $\mathcal{C}^j(p^j)$ generation/operation cost

– **revenue maximization**: $\mathcal{R}^j(\pi, p^j) = \pi^\top p^j - \mathcal{C}^j(p^j)$
for equilibrium price $\pi = \pi(p)$.

GOAL:

– **G's** *take unilateral decisions, behave competitively, and want to recover fixed costs in the long term (including capital remuneration).*

– *A representative of the consumers, the ISO, focuses on the benefits of consumption, seeking a price that matches supply and demand, "keeping prices low".*

A game-theoretical model

– $p^j = (p^j_1, \ldots, p^j_T)$ j-th power plant generation

– $p^j \in \mathcal{P}^j$ operational constraints

– $\mathcal{C}^j(p^j)$ generation/operation cost

– revenue maximization: $\mathcal{R}^j(\pi, p^j) = \pi^\top p^j - \mathcal{C}^j(p^j)$
for equilibrium price $\pi = \pi(p)$.

– **market clearing/demand satisfaction** $h(p) = h(p^j, p^{-j}) \leq 0$

GOAL:

– *G's take unilateral decisions, behave competitively, and want to recover fixed costs in the long term (including capital remuneration).*

– *A representative of the consumers,* **the ISO***, focuses on the benefits of consumption, seeking a price that matches supply and demand, "keeping prices low".*

A Nash Game with a shared constraint as a VI

To have solutions of

$$\text{find } \bar{p} \in \mathcal{P} : h(\bar{p}) \leq 0 \text{ and } \max_{p^j \in \mathcal{P}^j} \mathcal{R}^j(\pi(\bar{p}), \bar{p}^j) \text{ for all } j,$$

look for variational equilibria, solving instead the VI:

$$\text{find } \bar{p} \in \mathcal{P} \cap \{h(p) \leq 0\} : \left\langle F(\bar{p}), (p - \bar{p}) \right\rangle \geq 0 \text{ for all feasible } p$$

A Nash Game with a shared constraint as a VI

To have solutions of

$$\text{find } \bar{p} \in \mathcal{P} : h(\bar{p}) \leq 0 \text{ and } \max_{p^j \in \mathcal{P}^j} \mathcal{R}^j(\pi(\bar{p}), \bar{p}^j) \text{ for all } j,$$

look for variational equilibria, solving instead the VI:

$$\text{find } \bar{p} \in \mathcal{P} \cap \{h(p) \leq 0\} : \left\langle F(\bar{p}), (p - \bar{p}) \right\rangle \geq 0 \text{ for all feasible } p$$

For the electricity market problem

$$\begin{aligned} F(p) &= (\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j))_{j=1}^N \\ \mathcal{P} &= \textstyle\prod_j \mathcal{P}^j \\ p \text{ feasible} &\implies h(p^j, p^{-j}) \leq 0 \end{aligned}$$

A Nash Game with a shared constraint as a VI

To have solutions of

$$\text{find } \bar{p} \in \mathcal{P} : h(\bar{p}) \leq 0 \text{ and } \max_{p^j \in \mathcal{P}^j} \mathcal{R}^j(\pi(\bar{p}), \bar{p}^j) \text{ for all } j,$$

look for variational equilibria, solving instead the VI:

$$\text{find } \bar{p} \in \mathcal{P} \cap \{h(p) \leq 0\} : \langle F(\bar{p}), (p - \bar{p}) \rangle \geq 0 \text{ for all feasible } p$$

For the electricity market problem

$$\begin{array}{rcl} F(p) & = & \left(\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j)\right)_{j=1}^{N} \\ \mathcal{P} & = & \prod_j \mathcal{P}^j \qquad \textbf{OK!} \\ p \text{ feasible} & \Longrightarrow & h(p^j, p^{-j}) \leq 0 \end{array}$$

A Nash Game with a shared constraint as a VI

To have solutions of

$$\text{find } \bar{p} \in \mathcal{P} : h(\bar{p}) \leq 0 \text{ and } \max_{p^j \in \mathcal{P}^j} \mathcal{R}^j(\pi(\bar{p}), \bar{p}^j) \text{ for all } j,$$

look for variational equilibria, solving instead the VI:

$$\text{find } \bar{p} \in \mathcal{P} \cap \{h(p) \leq 0\} : \left\langle F(\bar{p}), (p - \bar{p}) \right\rangle \geq 0 \text{ for all feasible } p$$

For the electricity market problem

$$\begin{array}{rcll}
F(p) & = & \left(\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j)\right)_{j=1}^{N} & \textbf{NOK} \\
\mathcal{P} & = & \prod_j \mathcal{P}^j & \textbf{OK!} \\
p \text{ feasible} & \Longrightarrow & h(p^j, p^{-j}) \leq 0 & \textbf{NOK}
\end{array}$$

VI Decomposition

Finding $\langle F(\bar{p}), (p - \bar{p}) \rangle \geq 0$ for $p \in \mathcal{P} \cap \{h(p) \leq 0\}$ for

$$
\begin{aligned}
F(p) \quad &= \quad (\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j))_{j=1}^N && \textbf{NO OK} \\
\mathcal{P} \quad &= \quad \textstyle\prod_j \mathcal{P}^j && \textbf{OK!} \\
p \text{ feasible} \quad &\Longrightarrow \quad h(p^j, p^{-j}) \leq 0 && \textbf{NO OK}
\end{aligned}
$$

VI Decomposition

Finding $\langle F(\bar{p}), (p - \bar{p}) \rangle \geq 0$ for $p \in \mathcal{P} \cap \{h(p) \leq 0\}$ for

$$
\begin{array}{rcll}
F(p) & = & (\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j))_{j=1}^N & \textbf{NO OK} \\
\mathcal{P} & = & \prod_j \mathcal{P}^j & \textbf{OK!} \\
p \text{ feasible} & \Longrightarrow & h(p^j, p^{-j}) \leq 0 & \textbf{NO OK}
\end{array}
$$

means finding a primal-dual solution $(\bar{p}\ \ , \bar{x}\ \ )$ s.t.

$$
\begin{array}{rl}
\left\langle F(\bar{p}\ \ ) + h'(\bar{p}\ \ )^\top \bar{x}\ \ , p - \bar{p}\ \ \right\rangle \geq 0 \text{ for all } p \in \mathcal{P} & \textbf{(SOL}_{\mathcal{P}}\textbf{)} \\
0 \leq -h(\bar{p}\ \ ) \perp \bar{x}\ \ \geq 0 & \textbf{(FEAS}_{\mathbf{h}}\textbf{)}
\end{array}
$$

VI Decomposition

Finding $\langle F(\bar{p}), (p-\bar{p})\rangle \geq 0$ for $p \in \mathcal{P} \cap \{h(p) \leq 0\}$ for

$$
\begin{array}{rcll}
F(p) & = & (\partial_{p^j}\mathcal{R}^j(\pi(p^j, p^{-j}), p^j))_{j=1}^N & \textbf{NO OK} \\
\mathcal{P} & = & \prod_j \mathcal{P}^j & \textbf{OK!} \\
p \text{ feasible} & \Longrightarrow & h(p^j, p^{-j}) \leq 0 & \textbf{NO OK}
\end{array}
$$

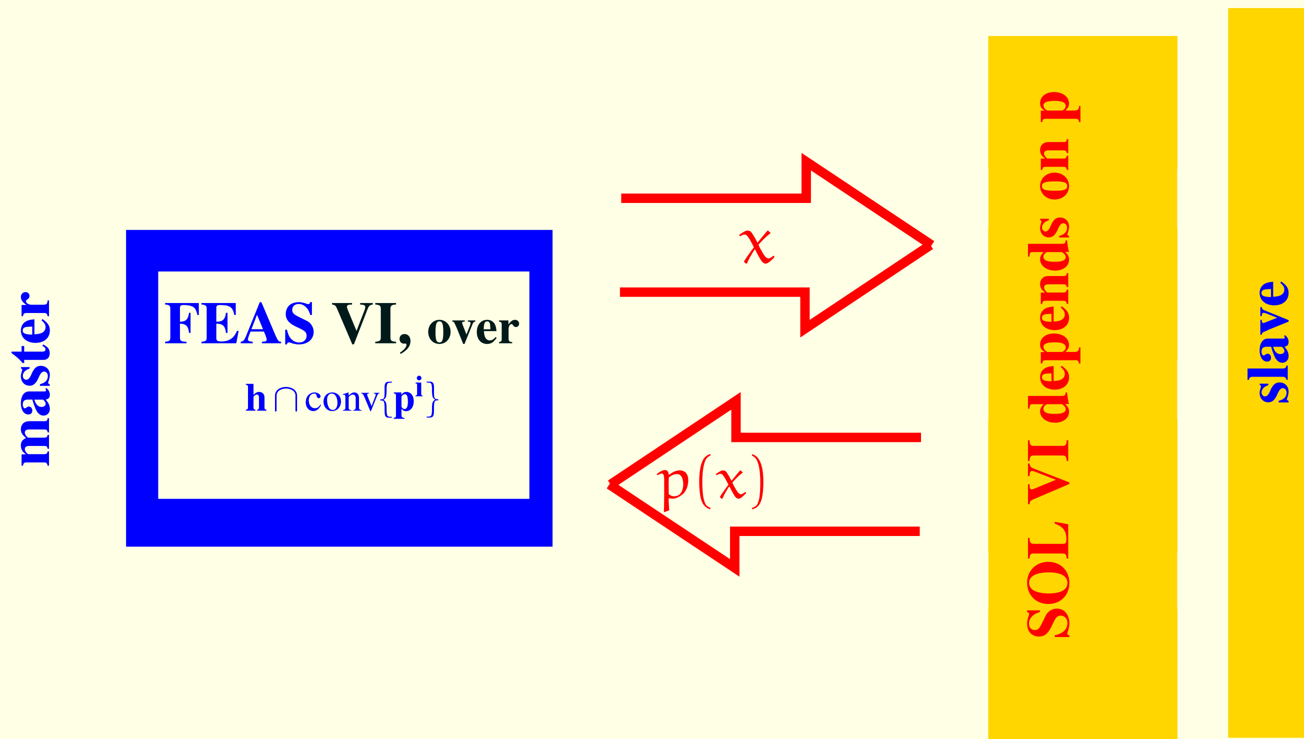
means finding a primal-dual sequence (p^k, x^k) s.t.

$$
\begin{array}{rl}
\left\langle F(p^k) + h'(p^k)^\top x^k, p - p^k \right\rangle \geq 0 \text{ for all } p \in \mathcal{P} & \mathbf{(SOL^k_{\mathcal{P}})} \\
0 \leq -h(p^k) \perp x^{k+1} \geq 0 & \mathbf{(FEAS^k_h)}
\end{array}
$$

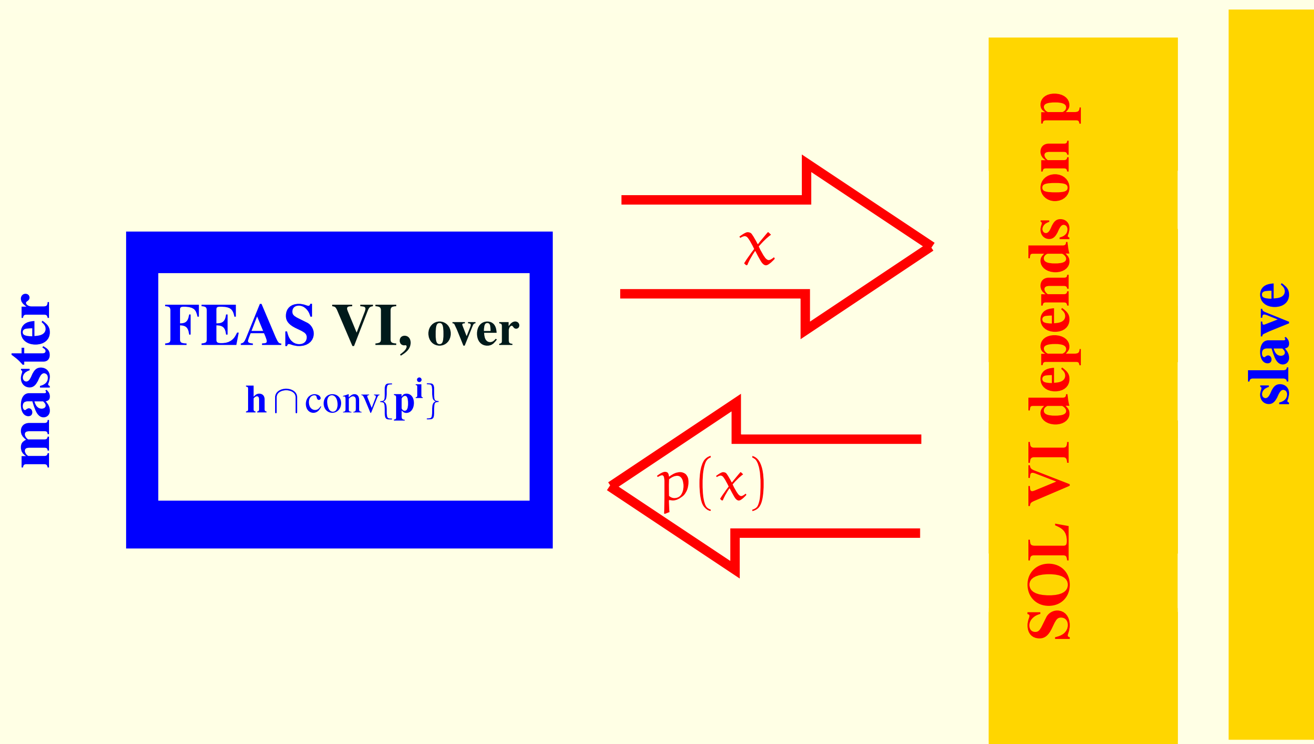
à la Dantzig-Wolfe[a]

[a] initiated by [ChFull03]

VI Decomposition



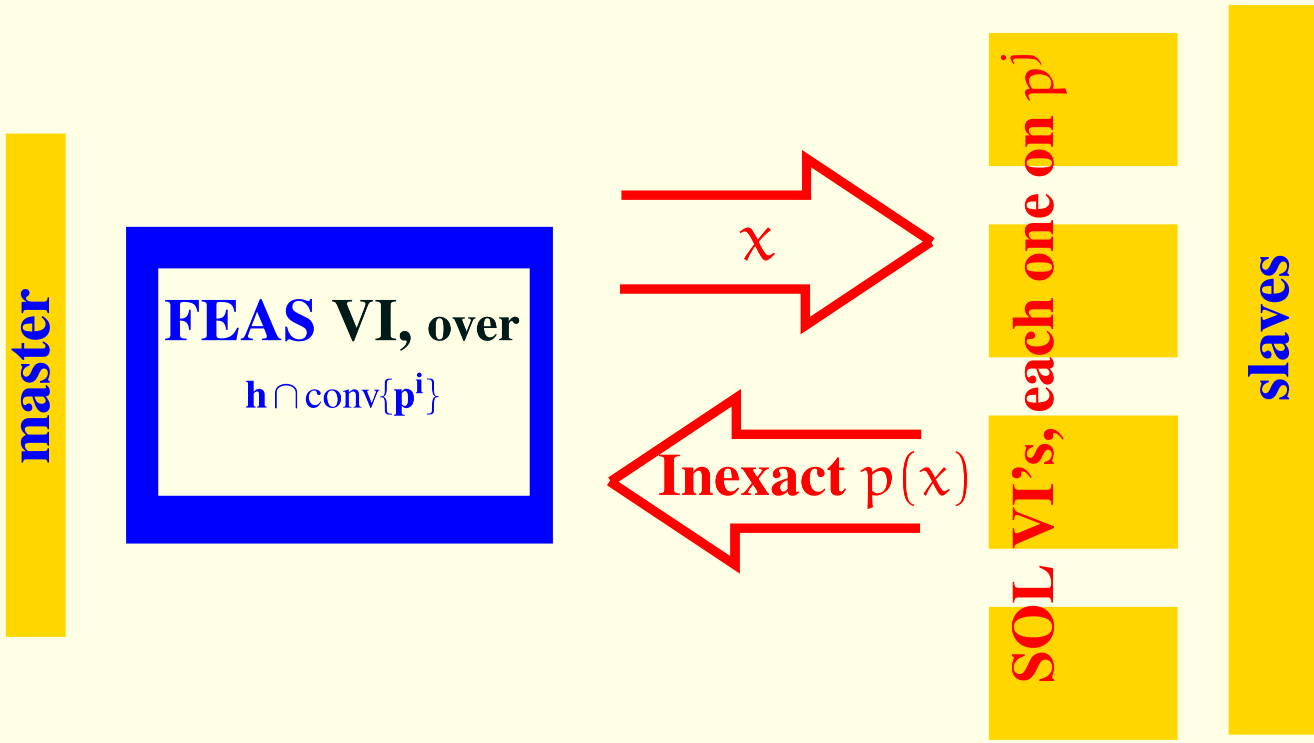
VI Decomposition has not "conquered" much!



SOL VI operator is not separable:

$$F(p) + h'(p)^\top x = (\partial_{p^j} \mathcal{R}^j(\pi(p^j, p^{-j}), p^j)) + h'(\pi(p^j, p^{-j}))^\top x$$

Separability induced, via inexact (Jacobi) oracle

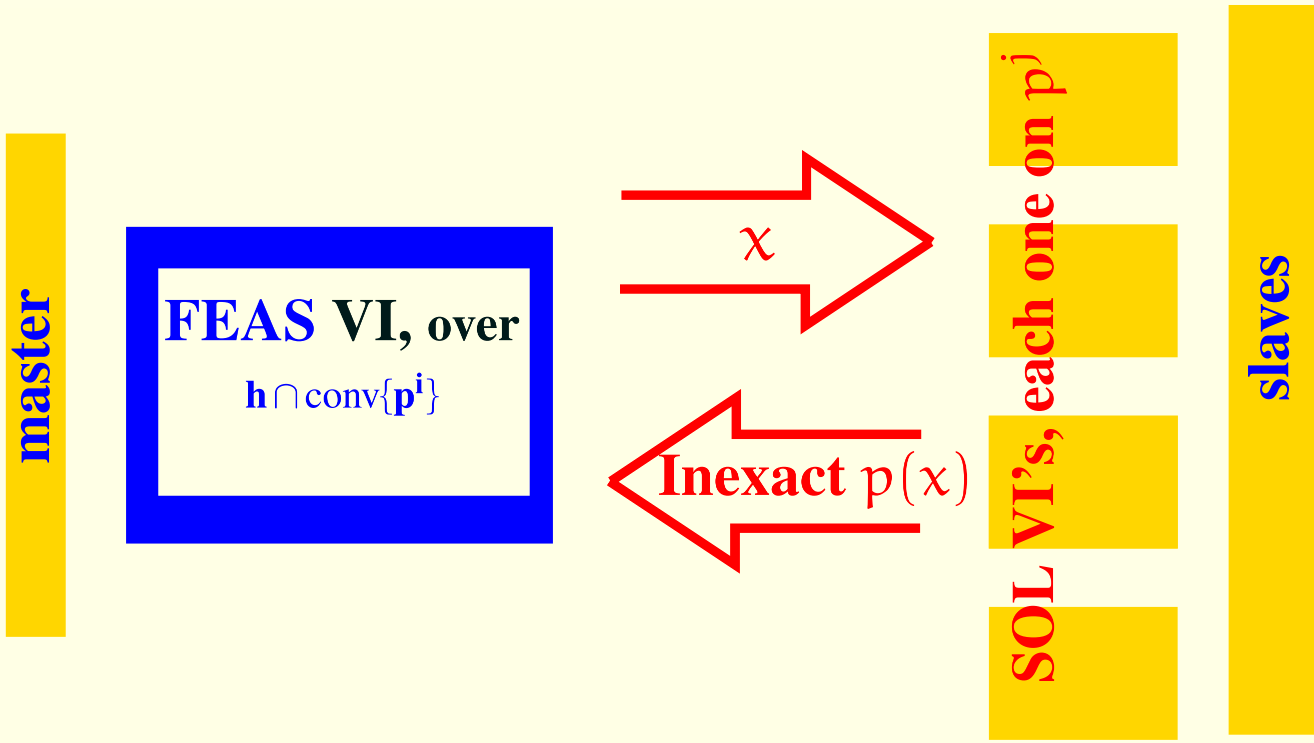


SOL VI operator is not separable, **approximated by**

$$F(p) + h'(p)^\top x \approx (\partial_{p^j} \mathcal{R}^j(\pi(p^j, \bar{p}^{-j}), p^j)) + h'(\pi(p^j, \bar{p}^{-j}))^\top x$$

for $\bar{p}$ the current iterate

Separability induced, via inexact (Jacobi) oracle



Good numerical results, extending the applicability of solvers like PATH

(joint work with J.P. Luna and M. Solodov)

Final remarks

– Energy problems are naturally challenging, some of them often need to be solved fast, but with high precision

– Decomposition methods are good in such a context, provided accurate solvers are employed for dealing with the "master" program

– Bundle methods able to handle on-demand accuracy oracles are promising in this respect. However:

  – Oracles with variable accuracy have an impact on `primal` variables, in a manner that is not yet well understood

  – Same for `dual` variables (progress is being done)

  – Research needs to be done regarding sound parameter updating rules (proximal/level) when the oracle is noisy.

  – Can we expect a bundle method to identify $\mathcal{VU}$ subspaces for asymptotically exact oracles?

– Can DW decomposition for VI's be stabilized, à la bundle?