From micro to macro structure: a journey in company of the Unit Commitment problem

Antonio Frangioni[1]

with T. Bacci[2], C. Gentile[2], R. Durbano Lobato[1], W. van Ackooij[3]

[1]Dipartimento di Informatica, Università di Pisa
[2]Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", C.N.R.
[3]EdF R&D "Osiris"

ICERM workshop "Linear and Non-Linear Mixed Integer Optimization"
Providence, February 28, 2023

Outline

The Electrical System

- Electrical system: the most complex machine mankind has developed
- Several sources of complexity:
  1. electricity is difficult to store $\Longrightarrow$ must be mostly produced exactly when needed
  2. electricity is difficult to route, goes where Kirchoff's laws say[1]
  3. growing renewables production is highly uncertain
  4. almost everything is (more or less highly) nonlinear
- All manner of (nasty) optimization problems, spanning from multi decades to sub-second
- Unit Commitment is one of the basic steps

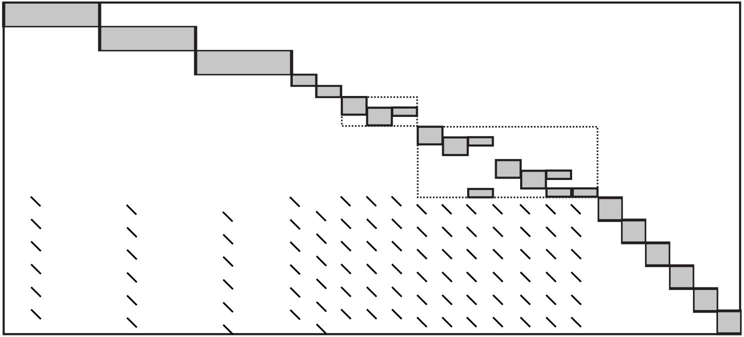
[1] Dan's talk yesterday

The Unit Commitment problem

- Schedule a set of generating units over a time horizon T (hours/15m in day/week) to satisfy the (forecasted) demand d_t at each $t \in T$
- Gazzillions €€€ / $$$, enormous amount of research[2]
- Different types of production units, different constraints:
  - Thermal (comprised nuclear): min/max production, min up/down time, ramp rates on production increase/decrease, start-up cost depending on previous downtime, others (modulation, ...)
  - Hydro (valleys): min/max production, min/max reservoir volume, time delay to get to the downstream reservoir, others (pumping, ...)
  - Non programmable (ROR hydro) intermittent units (solar/wind, ...)
  - Fancy things (small-scale storage, demand response, smart grids, ...)
- Plus the interconnection network (AC/DC, transmission/distribution) and reliability (primary/secondary reserve, $n - 1$ units, ...)

[2] van Ackooij, Danti Lopez, F., Lacalandra, Tahanan "Large-scale Unit Commitment Under Uncertainty [...]" *AOR* 2018

Algorithmic approaches



- Several types of almost independent blocks + linking constraints: perfect for decomposition methods[3], especially in the uncertain case[4]
- Many different structures, today's one: thermal units (but ∃ others, e.g. hydro units[5], Energy Communities[6], stochastic[2,4], . . .)

3 Borghetti, F., Lacalandra, Nucci "Lagrangian [. . .] for Hydrothermal Unit Commitment", *IEEE Trans. Power Sys.* 2003

4 Scuzziato, Finardi, F. "Comparing Spatial and Scenario Decomposition for Stochastic [. . .]" *IEEE Trans. Sust. En.* 2018

5 van Ackooij et. al. "Shortest path problem variants for the hydro unit commitment problem" *Elec. Notes Disc. Math.* 2018

6 Fioriti, F., Poli "Optimal Sizing of Energy Communities with Fair Revenue Sharing [. . .]" *Applied Energy* 2021

Basic aspects of thermal units

- Natural variables $p_t \in \mathbb{R}_+$: power level at time $t \in T$
- Standard constraints:
  - maximum ($\bar{p}_{max}$) and minimum ($\bar{p}_{min}$) power output at $t \in T$ (but start-up / shut-down limits $\bar{l}$ / $\bar{u}$ potentially $\neq$)
  - Ramp-up / down constraints (Δ_+ / Δ_- = ramp-up / down limit)
  - Min up / down-time constraints (τ_+ / τ_- = min up / down-time)
- Power cost: convex quadratic with fixed term

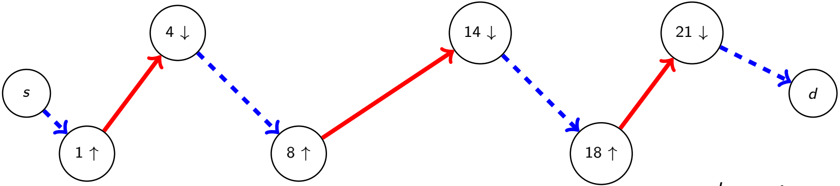
  $f_t(\, p_t \,) = a_t p_t^2 + b_t p_t + c_t$ (most often a, b, c independent from t)
- Possibly rather complex time-dependent start-up costs

  (dependent on intervening down time, maybe also on $t \in T$)

The basic DP algorithm

A(n improved) DP algorithm[7] based on the state-space graph G:

- nodes $(t, \uparrow)/(t, \downarrow)$: unit starts up/shuts down at time t
- arc $(\,(h, \uparrow)\,,\,(k, \downarrow)\,)$ with $k - h + 1 \geq \tau^+$: unit on from h to k (included)
- arc $(\,(h, \downarrow)\,,\,(k, \uparrow)\,)$ with $k - k - 2 \geq \tau^-$: unit off from $h + 1$ to $k - 1$
- An s-d path from represents a schedule for the unit



- "on" arcs $(\,(h, \uparrow)\,,\,(k, \downarrow)\,)$: optimal dispatching cost $z^*_{hk} + \sum_{t=h}^{k} c^i_t$
- "off" arcs $(\,(h, \downarrow)\,,\,(k, \uparrow)\,)$: start-up cost for $k - h - 2$ off time periods

[7] F., Gentile "Solving Nonlinear Single-Unit Commitment Problems with Ramping Constraints" *Op. Res.*, 2006

"on" arcs cost = Economic Dispatch

- Optimal dispatch cost z^*_{hk}: solving the Economic Dispatch problem (ED_{hk}) on $p_h, p_{h+1}, \ldots, p_k$

$$
\begin{aligned}
z^*_{hk} = \min \;\; & \textstyle\sum_{t=h}^{k} f^t(p_t) && && (1)\\
& p_{min} \leq p_h \leq \bar{l} && && (2)\\
& p_{min} \leq p_t \leq p_{max} && h+1 \leq t \leq k-1 && (3)\\
& p_{min} \leq p_k \leq \bar{u} && && (4)\\
& p_{t+1} - p_t \leq \Delta_+ && t = h, \ldots, k-1 && (5)\\
& p_t - p_{t+1} \leq \Delta_- && t = h, \ldots, k-1 && (6)
\end{aligned}
$$

- Complexity:
  - acyclic graph $O(n)$ nodes, $O(n^2)$ arcs $\Longrightarrow$ $O(n^2)$ for optimal path
  - $O(n^3)$ for computing costs via specialized inner DP[7] for (ED_{hk})

  $\Longrightarrow$ $O(n^3)$ overall

- Basic building block for efficient Lagrangian approaches[8]

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

Outline

Basic MIP formulation

- Natural variables $u^i_t \in \{0,1\}$: on / off state of unit $i \in P$ at $t \in T$
- Standard formulation with time-dependent start-up costs

$$\min \sum_{i \in P} \left[s^i(u^i) + \sum_{t \in T} \left(a^i_t (p^i_t)^2 + b^i_t p^i_t + c^i_t u^i_t \right) \right] \quad (7)$$

$$\bar{p}^i_{min} u^i_t \leq p^i_t \leq \bar{p}^i_{max} u^i_t \qquad t \in T \quad (8)$$

$$p^i_t \leq p^i_{t-1} + u^i_{t-1} \Delta^i_+ + (1 - u^i_{t-1}) \bar{l}^i \qquad t \in T \quad (9)$$

$$p^i_{t-1} \leq p^i_t + u^i_t \Delta^i_- + (1 - u^i_t) \bar{u}^i \qquad t \in T \quad (10)$$

$$u^i_t \leq 1 - u^i_{r-1} + u^i_r \qquad t \in T,\ r \in [t - \tau^i_+, t-1] \quad (11)$$

$$u^i_t \geq 1 - u^i_{r-1} - u^i_r \qquad t \in T,\ r \in [t - \tau^i_-, t-1] \quad (12)$$

requires extra constraints + continuous variables[9] for $s^i(u^i)$

- Global constraints: demand satisfaction + others (reserve, pollution, ...)

$$\sum_{i \in P} p^i_t = \bar{d}_t \qquad t \in T \quad (13)$$

- A nasty MIQP, unsolvable as-is for > 20 units (real-world versions worse)

[9] Nowak, Römisch "Stochastic Lagrangian Relaxation Applied to Power Scheduling [...]" *Annals O.R.* 2000

Improved MIP formulations (1)

- Convex hull of the min-up / down constraints (11) / (12) known[10]: exponential number of constraints, but separable in poly time
- Indeed, extended formulation[11]: start-up / shut-down v_t^i / w_t^i variables

$$u_t^i - u_{t-1}^i = v_t^i - w_t^i \qquad t \in T \tag{14}$$

- Can be extended to start-up / shut-down limits[12] ($\tau_+ \geq 2 \neq \tau_+ = 1$)

$$\begin{aligned} p_1 &\leq \bar{p}_{max} u_t && - (\bar{p}_{max} - \bar{u}) w_{t+1} \\ p_t &\leq \bar{p}_{max} u_t - (\bar{p}_{max} - \bar{l}) v_t && - (\bar{p}_{max} - \bar{u}) w_{t+1} \qquad t \in [2, |T| - 1] \\ p_T &\leq \bar{p}_{max} u_t - (\bar{p}_{max} - \bar{l}) v_t \end{aligned}$$

[10] Lee, Leung, Margot, "Min-up/Min-down polytopes", *Disc. Opt.*, 2004
[11] Rajan, Takriti, "Minimum Up/Down polytopes of the unit commitment problem with start-up costs", IBM RC23628, 2005
[12] Gentile, Morales-Espana, Ramos "A Tight MILP [...] Start-up and Shut-down Constraints", *EURO J. Comput. Opt.*, 2017

Improved MIP formulations (2)

- Ramp-up and Ramp-down polytopes studied separately[13]
- Ramp-up, convex hull for two-period case

$$
\begin{aligned}
& p_{min} u_t \leq p_t \leq p_{max} u_t \\
& 0 \leq v_{t+1} \leq u_{t+1} \\
& u_{t+1} - u_t \leq v_{t+1} \leq 1 - u_t \\
& p_{min} u_{t+1} \leq p_{t+1} \leq p_{max} u_{t+1} - (p_{max} - \bar{l}) v_{t+1} \\
& p_{t+1} - p_t \leq (p_{min} + \Delta_+) u_{t+1} + (\bar{l} - p_{min} - \Delta_+) v_{t+1} - p_{min} u_t
\end{aligned}
$$

- Some valid / facet defining inequalities for the general case
- Strengthened ramp-up / down constraints under some conditions[14]

[13] Damci-Kurty et al. "A Polyhedral Study of Ramping in Unit Commitment", *Math. Prog.* 2016

[14] Ostrowski, Anjos, Vannelli "Tight [...] formulations for the unit commitment problem" *IEEE TPWRS* 2012

Improved MIP formulations (3)

- Convex quadratic objective function with semi-continuous variables: Perspective Reformulations[15,16] $\sum_{t \in T} a_t^i (p_t^i)^2 / u_t^i + b_t^i p_t^i + c_t^i u_t^i$
- Several ways to deal with the "more nonlinearity"[17,18]
- Start-up cost is a concave in previous shut-down period length τ: $cs(\tau) = V(1 - e^{-\lambda\tau}) + F$ (only required for integer τ)
- Convex hull description of the start-up cost fragment: extended formulation with temperature variables[19]

---

15 F., Gentile "Perspective cuts for a class of convex 0-1 mixed integer programs" *Math. Prog.* 2006

16 F., Gentile, Lacalandra "Tighter approximated MILP formulations for Unit Commitment Problems" *IEEE TPWRS* 2009

17 F., Gentile "A Computational Comparison of [...]: SOCP vs. Cutting Planes" *ORL* 2009

18 F., Furini, Gentile "Approximated Perspective Relaxations: a Project&Lift Approach" *COAP* 2016

19 Silbernagl, Huber, Brandenberg "[...] MIP Unit Commitment by Modeling Power Plant Temperatures" *IEEE TPWRS* 2016

Improved MIP formulations (3)

- Convex quadratic objective function with semi-continuous variables: Perspective Reformulations[15,16] $\sum_{t \in T} a^i_t (p^i_t)^2 / u^i_t + b^i_t p^i_t + c^i_t u^i_t$
- Several ways to deal with the "more nonlinearity"[17,18]
- Start-up cost is a concave in previous shut-down period length τ: $cs(\tau) = V(1 - e^{-\lambda\tau}) + F$ (only required for integer τ)
- Convex hull description of the start-up cost fragment: extended formulation with temperature variables[19]
- All these works deal with partial fragments of the (thermal) (single-)Unit Commitment problem

[15] F., Gentile "Perspective cuts for a class of convex 0-1 mixed integer programs" *Math. Prog.* 2006
[16] F., Gentile, Lacalandra "Tighter approximated MILP formulations for Unit Commitment Problems" *IEEE TPWRS* 2009
[17] F., Gentile "A Computational Comparison of [...]: SOCP vs. Cutting Planes" *ORL* 2009
[18] F., Furini, Gentile "Approximated Perspective Relaxations: a Project&Lift Approach" *COAP* 2016
[19] Silbernagl, Huber, Brandenberg "[...] MIP Unit Commitment by Modeling Power Plant Temperatures" *IEEE TPWRS* 2016

DP algorithm → a new MIP formulation

- arc variables y_{on}^{hk} on arc $(\,(h,\uparrow)\,,\,(k,\downarrow)\,)$, y_{off}^{hk} on arc $(\,(h,\downarrow)\,,\,(k,\uparrow)\,)$
- network matrix E for G, rhs vector b for s-d path

$$Ey = b \tag{15}$$

- power variables p_t^{hk} for $t = h, \ldots, k$ for each "on" arc $(\,(h,\uparrow)\,,\,(k,\downarrow)\,)$

$$\left.\begin{array}{ll} \bar{p}_{min} y_{on}^{hk} \leq p_h^{hk} \leq \bar{l} y_{on}^{hk} & \\ \bar{p}_{min} y_{on}^{hk} \leq p_t^{hk} \leq \bar{p}_{max} y_{on}^{hk} & t = h+1, \ldots, k-1 \\ \bar{p}_{min} y_{on}^{hk} \leq p_k^{hk} \leq \bar{u} y_{on}^{hk} & \\ p_{t+1}^{hk} - p_t^{hk} \leq y_{on}^{hk} \Delta_+ & t = h, \ldots, k-1 \\ p_t^{hk} - p_{t+1}^{hk} \leq y_{on}^{hk} \Delta_- & t = h, \ldots, k-1 \end{array}\right\} \; \forall\,(h,k) \tag{16}$$

- (15)–(16) describes the convex hull if objective linear[20]
- Slightly ≠ version (independently obtained) use DP to separate cuts[21]

[20] F., Gentile "New MIP Formulations for the Single-Unit Commitment Problems with Ramping Constraints" IASI RR 2015

[21] Knueven, Ostrowski, Wang "Generating Cuts from the Ramping Polytope for the Unit Commitment [...]" *IJOC* 2018

About the new formulation

- $O(n^2)$ binary + $O(n^3)$ continuous variables, $O(n^3)$ constraints
- Computational usefulness dubious (but perfect for Structured DW[22])
- Convex hull proof uses polyhedral result, but no real reason for linearity
- In fact, "easy" MI-SOCP generalisation[23], useful because PR is SOC-able
  $v \geq ap^2 / u \quad \equiv \quad uv \geq ap^2$ (if $u \geq 0$) $\equiv$ rotated SOCP constraint
- Perspective Reformulation describes the convex envelope, which is (or at least it should have been[24]) clearly necessary
- Nonlinear generalization of known polyhedral result:
  appropriate composition of convex hulls gives the convex hull

[22] F., Gendron "A stabilized structured Dantzig-Wolfe decomposition method" *Math. Prog.*, 2013
[23] Bacci, F., Gentile Tavlaridis-Gyparakis "New MI-SOCP Formulations for the Single-Unit Commitment [...]" *IASI RR* 2019
[24] Bacci, F., Gentile "A Counterexample to an Exact Extended Formulation for the Single-Unit Commitment [...]" *IASI RR* 2019

More practical formulations I: p_t

- Idea 1: kill the many p_t^{hk} entirely
- Obvious map between 3-bin variables and flow ones

$$x_{it} = \sum_{(h,k): t \in T(h,k)} y_i^{hk} \;\;,\;\; v_{it} = \sum_{k \geq t} y_i^{tk} \;\;,\;\; w_{it+1} = \sum_{h \leq t} y_i^{ht}$$

- Strengthen 3-bin formulation using the flow variables:

$$p_{it} - p_{it-1} \leq -l_i \sum_{h : h \leq t-1} y_i^{ht-1} + \Delta_i^+ \sum_{(h,k): t-1 \in T(h,k-1)} y_i^{hk} + \bar{l}_i \sum_{k : k \geq t} y_i^{tk}$$

$$p_{it-1} - p_{it} \leq -l_i \sum_{k : k \geq t} y_i^{tk} + \Delta_i^- \sum_{(h,k): t-1 \in T(h,k-1)} y_i^{hk} + \bar{u}_i \sum_{h : h \leq t-1} y_i^{ht-1}$$

$$l_i \sum_{(h,k): t \in T(h,k)} y_i^{hk} \leq p_{it} \leq u_i \sum_{(h,k): t \in T(h,k)} y_i^{hk}$$

$$p_{it} \leq \bar{l}_i \sum_{k : k \geq t} y_i^{tk} + \bar{u}_i \sum_{h : h \leq t} y_i^{ht} + \sum_{(h,k): h < t < k} \psi_{it}^{hk} y_i^{hk}$$

(some changes needed when $\tau_i^+ = 1$ and at the beginning of time)

More practical formulations II: Start-Up

- Idea 2: aggregate the many p_t^{hk} somehow
- p_{it}^h for i started-up at h (don't care when shut-down), $p_{it} = \sum_{h:h \leq t} p_{it}^h$
- Modified formulation

$$p_{it}^h - p_{it-1}^h \leq -l_i y_i^{ht-1} + \Delta_i^+ \sum_{k:k \geq t} y_i^{hk}$$

$$p_{it-1}^h - p_{it}^h \leq \bar{u}_i y_i^{ht-1} + \Delta_i^- \sum_{k:k \geq t} y_i^{hk}$$

$$p_{i1}^0 \leq (\Delta^+ + p_0) \sum_{k:1 \leq k} y_i^{0k}$$

$$-p_{i1}^0 \leq (\Delta^- - p_0) \sum_{k:1 \leq k} y_i^{0k}$$

$$l_i \sum_{k:k \geq t} y_i^{hk} \leq p_{it}^h \leq u_i \sum_{k:k \geq t} y_i^{hk}$$

$$p_{ih}^h \leq \bar{l}_i \sum_{k:k > h} y_i^{hk} + \min\{\bar{l}_i, \bar{u}_i\} y_i^{hh}$$

$$p_{it}^h \leq \bar{u}_i y_i^{ht} + \sum_{k:k > t} \psi_{it}^{hk} y_i^{hk}$$

- Unfortunately did not work too well like "twin" Shut-Down[25]

[25] Bacci, F., Gentile "Start-up/Shut-down MINLP Formulations for the Unit Commitment [...]" *CTW2020*, 2021

Outline

The result: preliminaries

- Nonlinear version of "Approach no. 4"[26] known since Edmonds[27]
- Uses duality, hence in the nonlinear case has to be Lagrangian (was conic duality in the SOCP case[23])
- Closed convex $C = \{ z \in \mathbb{R}^n \,:\, f(z) \leq 0 \}$, its mixed-integer restriction $S = \{ z \in C \,:\, z_k \in \mathbb{Z} \quad k \in K \subseteq \{ 1, \dots, n \} \}$
- Arbitrary objective function $c \in \mathbb{R}^n$, support function of c
$$\sigma_C(c) = \inf \{ cz \,:\, z \in C \}$$
- Arbitrary objective function $c \in \mathbb{R}^n$, dual function of C
$$\sigma_C(c) \geq D(c) = \sup_{\lambda \geq 0} \{ L(\lambda; c) = \inf \{ cz + \lambda f(z) \} \}$$

[26] Wolsey "Integer Programming" 1998

[27] Edmonds "Matroids and the greedy algorithm" *Math. Prog.*, 1971

The result: preliminaries (2)

- Basic convex analysis: the Lagrangian dual does not distinguish a set from (the closure of) its convex hull $\Longrightarrow$ if the condition

$$\forall\, c \in \mathbb{R}^n \; \sigma_S(\, c\,) = \inf \{\, cz \,:\, z \in S \,\} = D(\, c\,) \tag{17}$$

holds, then $C = \overline{conv}(\, S\,)$

- Dual convex hull proof: $\forall\, c$ exhibit λ^* s.t. $L(\, \lambda^* \,;\, c\,) = \sigma_S(\, c\,)$

- We use the constraints description of C (un-necessary?)

Assumption

For each (closed convex) set C represented by (closed convex) constraint functions $f = [\, f_i \,]_{i=1,\dots,m} : \mathbb{R}^n \to \mathbb{R}^m$, assumptions hold such that the KKT conditions are both necessary and sufficient for global optimality

- Standard constraints qualification for f convex, but need not be[28]

[28] Lasserre, "On representations of the feasible set in convex optimization", *Opt. Letters*, 2010

The result: composition operation

- Two sets $S^h \subset \mathbb{R}^{n_h} \times \mathbb{R}$ for $h = 1, 2$, 1-sum composition:

$$S^1 \oplus S^2 = \{\, (\, x^1 \,,\, x^2 \,,\, y \,) \in \mathbb{R}^{n_1+n_2+1} \;:\; (\, x^h \,,\, y \,) \in S^h \quad h = 1, 2 \,\}$$

$\equiv$ "S^1 and S^2 only share the single variable y"

- $\oplus$ preserves both convexity and closedness
- The result: under mild assumptions, the 1-sum composition of convex hulls is the convex hull of the 1-sum composition
- Formal statement: for $h = 1, 2$, let $S^h \subset \mathbb{R}^{n_h} \times \mathbb{R}$. If
  1. the convex hull of S^h is described by the closed (convex) sets

$$C^h = \{\, (\, x^h \,,\, y \,) \in \mathbb{R}^{n_h+1} \;:\; y \geq 0 \;,\; f^h(\, x^h \,,\, y \,) \leq 0 \,\} \tag{18}$$

  2. Assumption 1 holds for all involved sets
  3. $(\, x^h \,,\, y \,) \in S^h \implies y \in \{\, 0 \,,\, 1 \,\}$ for $h = 1, 2$
  4. $\exists$ points $(\, \bar{x}^h \,,\, 0 \,) \in S^h$ and $(\, \tilde{x}^h \,,\, 1 \,) \in S^h$ for $h = 1, 2$

then $C^1 \oplus C^2 = \overline{conv}(\, S^1 \oplus S^2 \,)$

The result: sketch of proof (1)

- Arbitrarily choose $(c^1 , c^2 , d) \in \mathbb{R}^{n_1+n_2+1}$

- Define $L = \inf \{ c^1x^1 + c^2x^2 + dy \,:\, (x^h , y) \in S^h \quad h = 1, 2 \}$
  and $\quad L \geq \Pi = \inf \{ c^1x^1 + c^2x^2 + dy \,:\, (x^h , y) \in C^h \quad h = 1, 2 \}$

- Define the Lagrangian Dual of the latter

$$\Delta = \sup_{\mu \geq 0 \,,\, \lambda^1 \geq 0 \,,\, \lambda^2 \geq 0} \{ L(\mu , \lambda^1 , \lambda^2) \}$$

where $L(\mu , \lambda^1 , \lambda^2) =$

$$\inf_{x^1 \,,\, x^2 \,,\, y \geq 0} \{ c^1x^1 + c^2x^2 + (d - \mu)y + \lambda^1 f^1(x^1 , y) + \lambda^2 f^2(x^2 , y) \}$$

- Prove that $L = \Delta$

The result: sketch of proof (2)

- We extend (with zeros) subgradients of $f_i^1(\, x^1\, ,\, y\,)$ to account for x^2, and symmetrically for $f_i^2(\, x^2\, ,\, y\,)$
- By Assumption 1 the optimal solution giving Π satisfies

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} c^1 \\ c^2 \\ d - \mu \end{bmatrix} + \sum_{i=1}^{m_1} \lambda_i^1 \partial f_i^1(\, x^1\, ,\, y\,) + \sum_{i=1}^{m_2} \lambda_i^2 \partial f_i^2(\, x^2\, ,\, y\,) \qquad (19a)$$

$$\mu y = 0 \qquad (19b)$$

$$\lambda^1 f^1(\, x^1\, ,\, y\,) = 0 \qquad (19c)$$

$$\lambda^2 f^2(\, x^2\, ,\, y\,) = 0 \qquad (19d)$$

- For $h = 1, 2$ and fixed $y \in \{\, 0\, ,\, 1\,\}$ define

$$L_y^h = \min \{\, c^h x^h + dy \;:\; (\, x^h\, ,\, y\,) \in C^h\,\}$$

The result: sketch of proof (3)

- For $h = 1, 2$ define the equivalent (since $C^h = \overline{conv}(\, S^h \,)$) problems

$$\sigma^h = \min \{\, c^h x^h + (d + L_0^h - L_1^h) y \;:\; (\, x^h \,,\, y \,) \in S^h \,\} \tag{20}$$

$$\bar{\sigma}^h = \min \{\, c^h x^h + (d + L_0^h - L_1^h) y \;:\; (\, x^h \,,\, y \,) \in C^h \,\} \tag{21}$$

- Crucial property: $\bar{\sigma}^h = \sigma^h = L_0^h \implies$ both $y = 0$ and $y = 1$ is optimal

- Have dual solutions that satisfy KKT

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} c^h \\ d + L_0^h - L_1^h - \mu^h \end{bmatrix} + \sum_{i=1}^{m_h} \lambda_i^h \partial f_i^h(\, x^h \,,\, y \,) \tag{22a}$$

$$\mu^h y = 0 \tag{22b}$$

$$\lambda^h f^h(\, x^h \,,\, y \,) = 0 \tag{22c}$$

for both $y = 0$ and $y = 1$

The result: sketch of proof (4)

- Now, "easy" case: $L = L_0 \leq L_1$, i.e., $y = 0$ is optimal
- Can construct solution of (19) using these of (22) for $y = 0$
- "Complicated" case: $L = L_1 < L_0$, i.e., $y = 1$ is optimal
- Further auxiliary problem

  $\sigma = \min \{ (L - L_0)y \, : \, (x^1 , y) \in S^1 \} = \min \{ (L - L_0)y \, : \, (x^1 , y) \in C^1 \}$

  where every $(x^1 , 1) \in C^1$ is optimal, with KKT

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \in \begin{bmatrix} 0 \\ L - L_0 \end{bmatrix} + \sum_{i=1}^{m_1} \tilde{\lambda}_i \partial f_i^1(, \tilde{x}^1 , 1) \tag{23a}$$

$$\tilde{\lambda}^h f^h(\tilde{x}^h , 1) = 0 \tag{23b}$$

- Can construct solution of (19) using these of (22) for $y = 1$ and (23)
- All cases finished, proof completed

Consequence: star-shaped MINLP

- Star-shaped MINLP: constructed by a set of 1-sum compositions
- If each piece has the convex hull property, so does the MINLP
- Our formulation is of this kind:
  - network flow has the integrality property
  - for generic convex f, the Perspective Reformulation

  $$z^{hk} \geq \sum_{t \in T(h,k)} y^{hk} f(\, p_t^{hk} \,/\, y^{hk} \,)$$

  describes the convex hull (all p_t^{hk} depend on the same y^{hk})
- Likely to have several other applications
- We suspect simpler and/or more general proofs possible
- We need them for Start-Up and Shut-Down polytopes ...

Outline

Sample computational results: root note bound

	3bin		DP		p_t		LR	
units	time	gap	time	gap	time	gap	time	gap
10	0.21	1.03	78.56	0.67	1.00	0.53	0.46	0.67
20	0.90	0.93	480.02	0.51	2.58	0.27	0.83	0.51
50	4.18	0.81	3836.78	0.08	9.92	0.09	1.19	0.08

- Perspective Cuts always included (bounds too much worse if not)
- Obvious trade-off between root bound and LP cost, DP impractical
- Cplex cuts effective for small n / weaker formulation, less otherwise
- $|T| = 24$, p_t scales worse than 3bin for larger T (but bounds $\approx$ same)
- LR very competitive, but Lagrangian-based B&B still in the works

Sample computational results: overall B&C

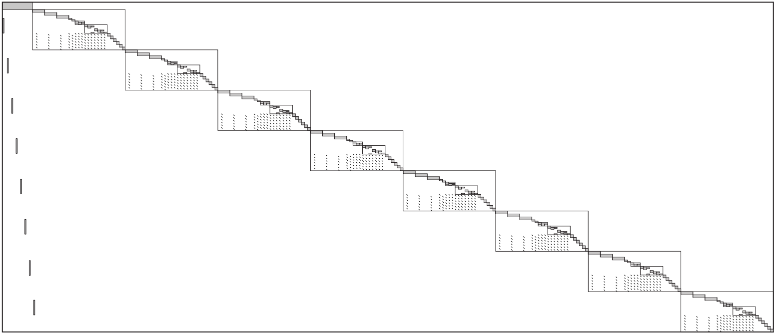
n	3-bin time	opt	nodes	gap	DP time	opt	nodes	gap	p_t time	opt	nodes	gap
10	28	5	275	0.01	832	5	599	0.01	5	5	41	0.01
20	7036	2	3561	0.08	7902	2	1961	0.05	1066	5	1234	0.01
50	10000	0	1619	0.12	10000	0	695	0.14	8095	1	2303	0.03
10	21	5	163	0.09	500	5	444	0.10	2	5	1	0.08
20	6002	2	1980	0.11	5490	4	1237	0.11	37	5	74	0.10
50	6052	2	1042	0.14	6927	3	504	0.11	160	5	148	0.08

- Above stop gap 1e−4, below stop gap 1e−3 (even less in practice)
- p_t formulation promising: maybe smaller exact formulation?
- $|T| = 24$, again p_t suffers more than 3bin for larger T
- Stabilised Structured DW may make DP / p_t^h (more) competitive (but 10 years in the making and still a lot of work to do)

Outline

Seasonal Storage Valuation

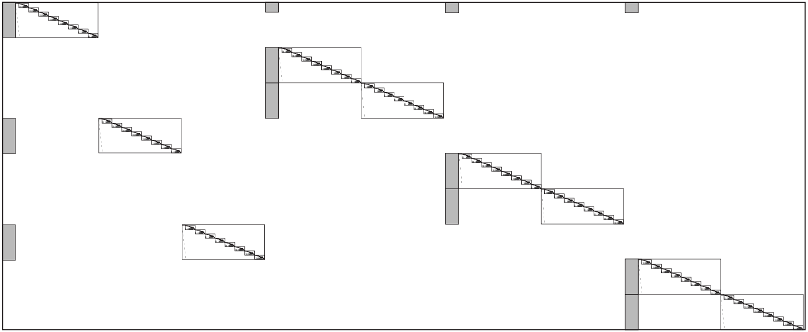
- Mid-term (1y) cost-optimal management of water levels in reservoirs considering uncertainties (inflows, temperatures, demands, ...)



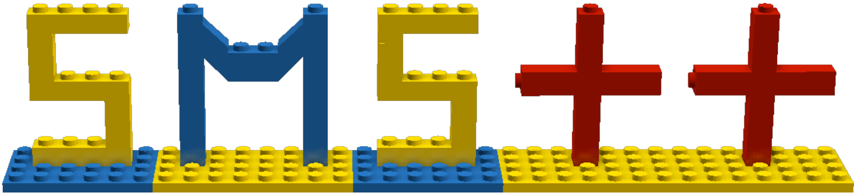
- Very large size, nested structure
- Perfect structure for Stochastic Dual Dynamic Programming
- SDDP requires (strong) convex relaxation (duals), any of the above

Investment Layer

- Long-term (30y) optimal (cost, pollution, CO_2 emissions, ...) planning of production/transmission investments considering multi-level uncertainties scenarios (technology, economy, politics, ...)



- Many scenarios, huge size, multiple nested structure $\Longrightarrow$ multiple nested Benders' or Lagrangian decomposition and/or SDDP
- Extremely challenging to implement, some help sorely needed



https://gitlab.com/smspp/smspp-project

"For algorithm developers, from algorithm developers"

- Open source (LGPL3)
- 1 "core" repo, 1 "umbrella" repo, 10+ problem and/or algorithmic-specific repos (public, more in development)
- All the above implemented (`UCBlock`, `SDDPBlock`, `BundleSolver`, . . .)
- Extensive Doxygen documentation https://smspp.gitlab.io
- But no real user manual as yet (except me)

What SMS++ is

- A core set of C++-17 classes implementing a modelling system that:
  - explicitly supports the notion of Block ≡ nested structure
  - separately provides "semantic" information from "syntactic" details (list of constraints/variables ≡ one specific formulation among many)
  - allows exploiting specialised Solver on Block with specific structure
  - manages any dynamic change in the Block beyond "just" generation of constraints/variables
  - supports reformulation/restriction/relaxation of Block
  - has built-in parallel processing capabilities
  - should be able to deal with almost anything (bilevel, PDE, ...)
- An hopefully growing set of specialized Block and Solver
- In perspective an ecosystem fostering collaboration and code sharing: a community-building effort as much as a (suite of) software product(s)

What SMS++ is not

- An algebraic modelling language: `Block` are C++ code (although it provides some modelling-language-like functionalities)
- For the faint of heart: primarily written for algorithmic experts (although users may benefit from having many pre-defined `Block`)
- Stable: only version 0.5.1, lots of further development ahead, significant changes in (parts) of interfaces actually expected (although current `Block` / `Solver` very thoroughly tested)
- Interfaced with many existing solvers: `Cplex`, `SCIP`, `MCFClass`, `StOpt` (although the list should hopefully grow)
- Ripe with native structure-exploiting solvers: `LagrangianDualSolver` and `SDDPSolver` for now (although the list should hopefully grow)

Outline

Seasonal Storage Valuation – some results I

- SDDPSolver requires convex problem: any of the above
- Single node (Switzerland)
- 60 stages (1+ year), 37 scenarios, 168 time instants (weekly UC)
- Units: 3 intermittent, 5 thermals, 1 hydro
- Out-of-sample simulation: all 37 scenarios to integer optimality

	Cont. relax.	Lag. relax.
Cost: Avg. / Std.	1.3165e+11 / 2.194e+10	1.2644e+11 / 2.167e+10
Time:	25m	7h30m

- Much longer, but:
  - simulation cost ≈ 30m per scenario, largely dominant
  - save 4% just changing a few lines in the configuration
  - LR time can be improved (ParallelBundleSolver not used)

Seasonal Storage Valuation – some results II

- A different single node (France)
- 60 stages (1+ year), 37 scenarios, 168 time instants (weekly UC)
- 83 thermals, 3 intermittent, 2 batteries, 1 hydro
- Out-of-sample simulation: all 37 scenarios to integer optimality

	Cont. relax.	Lag. relax.
Cost: Avg. / Std.	3.951e+11 / 1.608e+11	3.459e+11 / 8.903e+10
Time:	5h43m	7h54m

- Time not so bad (and 3h20m on average simulation per scenario) using `ParallelBundleSolver` with 5 threads per scenario
- That's 14% just changing a few lines in the configuration
- Starts happening regularly enough (and lower variance) to be believable

Investment Layer – some results I

- Simplified version: solve SDDP only once, run optimization with fixed value-of-water function + simulation (`SDDPGreedySolver`)
- EdF EU scenario: 11 nodes (France, Germany, Italy, Switzerland, Eastern Europe, Benelux, Iberia, Britain, Balkans, Baltics, Scandinavia), 20 lines
- Units: 1183 battery, 7 hydro, 518 thermal, 40 intermittent
- 78 weeks hourly (168h), 37 scenarios (demand, inflow, RES generation)
- Investments: 3 thermal units + 2 transmission lines.
- Average cost: original (operational) 6.510e+12
  optimized (investment + operational) 5.643e+12
- This is ≈ 1 Trillion Euro, 15%
- Running time: ??? hours for value-of-water functions (EdF provided) + 10 hours (4 scenarios in parallel + `ParallelBundleSolver` with 6 threads) for the investment problem

Investment Layer – some results II

- Simplified version (fixed value-of-water with continuous relaxation)
- Same 11 nodes, 19 lines
- Less units: 7 hydros, 44 thermals, 24 batteries, and 42 intermittent
- More investments: 82 units + 19 transmission lines.
- 78 weeks hourly (168h), 37 scenarios (demand, inflow, RES generation)
- Average cost: original (operational) 3.312e+12
  optimized (investment + operational) 1.397e+12
- This is ≈ 2 Trillion Euro, 137%
- Running time: 48 hours for value-of-water functions (2 nodes = 96 cores) + 5h 20m to solve the investment problem (1 nodes = 48 core)

Investment Layer – some results IIII

- Same simplified version as above
- EdF EU scenario: 14 nodes (France, Germany, Italy, Switzerland, Eastern Europe, Benelux, Iberia, Britain, Balkans, Baltics, Denmark, Finland, Sweden, Norway), 28 lines
- Units: 62 thermals, 54 intermittent, 8 hydros, 39 batteries
- 78 weeks hourly (168h), 37 scenarios (demand, inflow, RES generation)
- Investments: 99 units of all kinds + all transmission lines
- Average cost: original (operational) 3.465e+12
  optimized (investment + operational) 4.708e+11
  one order of magnitude saving (suspect most value of lost load)
  636% better investing on just 4 lines and 10 hydrogen power plants
- Running time: 7 hours on 48 cores, 375GB of RAM

Investment Layer – the Big Kahuna results

- The true version: value-of-water recomputed anew for each investment
- As usual, SDDP with Continuous or Lagrangian
- One node (48 core, 375Gb) not enough, must MPI-distribute over many
- Roll of drums . . .

Investment Layer – the Big Kahuna results

- The true version: value-of-water recomputed anew for each investment
- As usual, SDDP with Continuous or Lagrangian
- One node (48 core, 375Gb) not enough, must MPI-distribute over many
- Roll of drums . . .

Nope, sorry, still running

- Hoped it would be ready, but many problems (heavy checkpointing . . .)
- Our CINECA grant just expired and hasn't been renewed yet
- Not even the real Big Kahuna, we should have 5-years scenarios
- But we are getting there, thanks to `SMS++`

Outline

Conclusions — Energy Problems & MINLP

- Energy Problems = an endless source of inspiration
- "Challenging problems require good methodologies, challenging problems motivate methodological advances": very true for me
- 1^{st} complete (correct[24] and proven) convex hull formulation for (single)-UC with ramping and nonlinear costs
- Technical lemma fully expected but still possibly useful (extensions?)
- Possibly several other more star-shaped MINLPs (or similia)
- "Large" formulations possibly useful, trade-offs to be navigated (did I mention Stabilised Structured DW already?)
- From micro structure (Perspective Reformulation) to mid (1UC) to standard (UC) to large-scale (SSV) to huge-scale (IL): a hell of a ride!

Conclusions — SMS++

- SMS++ is there, actively developed
- Allows exploiting multiple nested heterogeneous structure, ≈ the only system designed for huge-scale (stochastic but not only) problems
- Could become really useful after having attracted mindshare, self-reinforcing loop (very hard to start)

[29] F., Perez Sanchez "Transforming Mathematical Models Using Declarative Reformulation Rules" *LNCS*, 2011

Conclusions — SMS++

- SMS++ is there, actively developed
- Allows exploiting multiple nested heterogeneous structure, ≈ the only system designed for huge-scale (stochastic but not only) problems
- Could become really useful after having attracted mindshare, self-reinforcing loop (very hard to start)
- Hefty, very likely rather unrealistic, sough-after impacts:
  - improve collaboration and code reuse, reduce huge code waste
  - significantly increase the addressable market of decomposition
  - a much-needed step towards higher uptake of parallel methods
  - the missing marketplace for specialised solution methods
  - a step towards a reformulation-aware modelling system[29]

[29] F., Perez Sanchez "Transforming Mathematical Models Using Declarative Reformulation Rules" *LNCS*, 2011

Conclusions — SMS++

- SMS++ is there, actively developed
- Allows exploiting multiple nested heterogeneous structure, ≈ the only system designed for huge-scale (stochastic but not only) problems
- Could become really useful after having attracted mindshare, self-reinforcing loop (very hard to start)
- Hefty, very likely rather unrealistic, sough-after impacts:
  - improve collaboration and code reuse, reduce huge code waste
  - significantly increase the addressable market of decomposition
  - a much-needed step towards higher uptake of parallel methods
  - the missing marketplace for specialised solution methods
  - a step towards a reformulation-aware modelling system[29]
- As much a community-building effort as an actual software project
- Lots of fun to be had, all contributions welcome

[29] F., Perez Sanchez "Transforming Mathematical Models Using Declarative Reformulation Rules" *LNCS*, 2011