A decomposition framework for solving dynamic MINLP problems under uncertainty

Laureano F. Escudero
Universidad Rey Juan Carlos, Mostoles (Madrid), Spain
laureano.escudero@urjc.es
Joint work w/ Araceli Garin, Maria Merino, Gloria Perez
Universidad del Pais Vasco, Spain

MINLP workshop
Carnegie Mellon University, Pittsburgh, June 2-5, 2014
Dynamic math optim under uncertainty
A MINLP Branch-and-Fix Coordination: Exact parallel algo
Asynchronous parallel algos:
  Fix-and-Relax Coordination
  Stochastic Dynamic Programming
Some computational experience for MILP.
  Large scale instances
Suggestion for efficiency increasing
MINLP under uncertainty. Can be done?
MINLP under uncertainty. Can be done?

Perhaps a strong statement: Stochastic programming machinery is ready for MILP models. See below some computational results.

It is possible for MINLP models. Why and what:
First, deterministic MINLP models up to the following # vars can be solved:
- Convex MINLP: 500 (nonconvex: 100)
- Convex NLP: $5 \times 10^4$ (nonconvex: 100)
- Convex SOCP: $10^5$ (nonconvex: 150)
- Convex MIQP: 1000 (nonconvex: 300)
- Convex QP: $5 \times 10^5$ (nonconvex: 300)

Source: MT 2 Mixed Integer Nonlinear Optimization by Layffer-Linderoth-Luedtke, SIAM OPT’14 (if I took it correctly).
Second, some types of MINLP problems exhibit this type of model:

\[
\begin{align*}
\text{máx } \eta \\
\text{s.t. } & f^\xi_0(x, y) \geq \eta \\
& f^\xi_i(x, y) \leq 0 \quad \forall i \in \mathcal{I} \\
& x \in \{0, 1\}^{n_01}, \quad y^g \in \mathbb{R}^{+n_c},
\end{align*}
\]  

(1)

where \( \xi \) represents the uncertainty in the parameters.

and third, exist a broad application field (e.g., stochastic networks, EGTCEP, SCM, PP), provided that non-optimal solns are also accepted:

- Dynamic models along a horizon
- \( f(x, y) \) (smooth?) convex functions
- non-very high number of 0-1 vars and
- non-high number of scenarios to represent parameters’ uncertainty.
Multistage scenario tree

A **stage** of a given horizon is a set of consecutive time units where the realization of the uncertain parameters takes place.

A **scenario** is a realization of the uncertain parameters along the stages of a given horizon.

A **scenario group** for a given stage is the set of scenarios with the same realization of the uncertain parameters up to the stage.
\( \Omega = \Omega^1 = \{10, 11, \ldots, 17\}; \quad \Omega^2 = \{10, 11, 12\} \)
\( G = \{1, \ldots, 17\}; \quad G_2 = \{2, 3, 4\} \)
\( \mathcal{A}^{17} = \{1, 4, 9, 17\} \)

**Figura:** Multistage nonsymmetric scenario tree
Notation

$\mathcal{T}$, set of the $T$ stages along the horizon.

$\Omega$, set of scenarios.

$\mathcal{G}$, set of scenario groups, so that we have a directed graph where $\mathcal{G}$ is the set of nodes.

$\mathcal{G}^t$, set of scenario groups in stage $t$, for $t \in \mathcal{T}$ ($\mathcal{G}^t \subseteq \mathcal{G}$).

$\Omega^g$, set of scenarios in group $g$, for $g \in \mathcal{G}$ ($\Omega^g \subseteq \Omega$).

$t(g)$, stage to which scenario group $g$ belongs to, for $g \in \mathcal{G}$.

$w^\omega$, likelihood or weight assigned by the user to scenario $\omega \in \Omega$.

$w^g$, weight assigned by modeler to scenario group $g \in \mathcal{G}$. It is computed as $w^g = \sum_{\omega \in \Omega} w^\omega$.

$\mathcal{A}^g$, set of ancestor nodes (scenario groups) in the scenario tree to node (scenario) $g$ (including itself), for $g \in \mathcal{G}$. Note: Any scenario group $g$ from last stage is a singleton and, since $\omega \in \Omega^g$ for $g \in \mathcal{G}^T$, then let us consider $g \equiv \omega$. 

Laureano F. Escudero  Universidad Rey Juan Carlos, Mostoles (M)  D-SMINLP
Multistage MINLP DEM: Risk neutral
Compact representation

\[ z_{RN} = \max \eta \]

\[ \text{s.t.} \quad \sum_{g \in G} w^g f^g_0 (x^g', y^g' \ \forall g' \in A^g) \geq \eta \]

\[ f^g_i (x^g', y^g' \ \forall g' \in A^g) \leq 0 \quad \forall i \in I, \ g \in G \]

\[ x^g \in \{0, 1\}^{n_x(g)}, \ y^g \in \mathbb{R}^{+n_y(g)} \quad \forall g \in G. \]
Break stage $t^*$, $t^*$—decomposition
Number of clusters, $|C|$, $1 < |C| \leq |\Omega|$, $|C|$ MINLP submodels:

- If $t^* = 1$ then, $|C| = 2$ cluster submodels.
- If $t^* = 2$ then, $|C| = 4$ cluster submodels.
- If $t^* = 3$ then, $|C| = 7$ cluster submodels.
Scenario clustering as a framework for MINLP problem solving

Break stage $t^*$, $t^*$—decomposition
Number of clusters, $|C|$, $1 < |C| \leq |\Omega|$, $|C|$ MINLP submodels:

- If $t^* = 1$ then, $|C| = 2$ cluster submodels.
- If $t^* = 2$ then, $|C| = 4$ cluster submodels.
- If $t^* = 3$ then, $|C| = 7$ cluster submodels.
Scenario clustering as a framework for MINLP problem solving

Break stage $t^*$, $t^*$—decomposition 
Number of clusters, $|C|$, $1 < |C| \leq |\Omega|$, $|C|$ MINLP submodels:

- If $t^* = 1$ then, $|C| = 2$ cluster submodels.
- If $t^* = 2$ then, $|C| = 4$ cluster submodels.
- If $t^* = 3$ then, $|C| = 7$ cluster submodels.

![Graph of submodels](attachment:image.png)
Scenario clustering as a framework for MINLP problem solving

Break stage $t^*$, $t^*$—decomposition
Number of clusters, $|C|$, $1 < |C| \leq |\Omega|$, $|C|$ MINLP submodels:

- If $t^* = 1$ then, $|C| = 2$ cluster submodels.
- If $t^* = 2$ then, $|C| = 4$ cluster submodels.
- If $t^* = 3$ then, $|C| = 7$ cluster submodels.
Scenario clustering

**Definition**

A **break stage** $t^*$ is a stage such that the set of scenario clusters is $C = |G^{t^*+1}|$, where $t^* + 1 \in \mathcal{T}$. In this case, any cluster $c \in C$ is induced by a group $g \in G^{t^*+1}$ and contains all the scenarios belonging to that group.

**Definition**

The MINLP **scenario cluster** submodels are those that result from the relaxation of the NAC until break stage $t^*$.

Let set $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, where $\mathcal{T}_1 = \{1, \ldots, t^*\}$ and $\mathcal{T}_2 = \mathcal{T} \setminus \mathcal{T}_1$. 
Scenario clustering

**Definition**

A **break stage** $t^*$ is a stage such that the set of scenario clusters is $C = |G^{t^*+1}|$, where $t^* + 1 \in \mathcal{T}$. In this case, any cluster $c \in C$ is induced by a group $g \in G^{t^*+1}$ and contains all the scenarios belonging to that group.

**Definition**

The MINLP **scenario cluster** submodels are those that result from the relaxation of the NAC until break stage $t^*$.

Let set $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$, where $\mathcal{T}_1 = \{1, \ldots, t^*\}$ and $\mathcal{T}_2 = \mathcal{T} \setminus \mathcal{T}_1$. 
MINLP submmodel for cluster $c \in C$

$$z_c = \max \eta_c$$

s.t.  
$$\sum_{g \in G_c} w^g_c f^g_0 (x^g_c \, , \, y^g_c \, \forall g' \in A^g) \geq \eta_c$$

$$f^g_i (x^g_c \, , \, y^g_c \, \forall g' \in A^g) \leq 0 \quad \forall i \in I, \, g \in G_c$$

$$x^g_c \in \{0, 1\}^{n_x(g)}, \, y^g_c \in \mathbb{R}^{+n_y(g)} \quad \forall g \in G_c.$$  

where $w^g_c = \sum_{\omega \in \Omega g \cap \Omega_c} w^\omega$, such that

$$w^g_c = w^{g'}$ being $g' \in G^{t^*+1} \cap G_c$ for $g : t(g) \in T_1$, and

$$w^g_c = w^g$ for $g : t(g) \in T_2.$$

Note: Implicit NAC (compact repr.) for each cluster.
NAC for linking cluster submodels

\[ x^g_c - x^g_{c'} = 0, \ y^g_c - y^g_{c'} = 0 \quad \forall c, c' \in C^g, \ g \in G^t, \ t \in T_1. \quad (4) \]

Note: \( G_c \cap G_{c'} \) is non-empty for \( c, c' \in C^g, \ g \in G^t \) for \( t \in T_1 \)
Cluster splitting-compact repr. of DEM MINLP (2)

\[ z_{RN} = \max_\eta \]

s.t. \[ \sum_{c \in C} \sum_{g \in G_c} w^g_c f^g_0 (x^g_c, y^g_c \forall g' \in A^g) \geq \eta \]

\[ f^g_i (x^g_c, y^g_c \forall g' \in A^g) \leq 0 \quad \forall i \in I, \ g \in G_c, \ c \in C \]

\[ x^g_c - x^{g'}_{c'} = 0, \ y^g_c - y^{g'}_{c'} = 0 \quad \forall c, c' \in C^g, \ g \in G_t, \ t \in T_1 \]

\[ x^g_c \in \{0, 1\}^{n_x(g)}, \ y^g_c \in \mathbb{R}^{+n_y(g)} \quad \forall g \in G_c, \ c \in C. \]

(5)

Branch-and-Fix Coordination (BFC) methodology: Relax from the model the explicit NAC (splitting variable repr.) between clusters, but it algorithmically takes care of those NAC for the \( x \)-variables, see LFE-Garín-Merino-Pérez COR’10,12.

What about the explicit NAC for the \( y \)-variables?
MINLP under uncertainty. Still it can be possible?

- It depends on the modeler-driven value of break stage $t^*$. 
  - $t^* = 0$: Full compact model (2): only one scenario cluster MINLP model, too big, no NAC relaxation. difficult!
  - $t^* = T - 1$: Full splitting model (5): Singleton scenario cluster MINLP models, too many NAC (4) on $y$-variables up to break stage $t^*$ to relax in a first shot, risky!.
- Best value for $t^*$: Smallest one such the largest scenario cluster MINLP model (3) is up to the following $\#$ vars:
  - Convex MINLP: 500 (nonconvex: 100)
  - Convex NLP: $5 \times 10^4$ (nonconvex: 100)
  - Convex SOCP: $10^5$ (nonconvex: 150)
  - Convex MIQP: 1000 (nonconvex: 300)
  - Convex QP: $5 \times 10^5$ (nonconvex: 300)


- Reason: The MINLP models (3) $\forall c \in C$ can be solved in a coordinated parallel mode for (algorithmically) satisfying the NAC (4) on the $x$-variables that have been relaxed up to break stage $t^*$.
Scenario cluster based Branch-and-Fix (BF) tree is the Branch-and-Bound tree for a scenario cluster, such that the optimization of the submodel \((3)\) for any scenario cluster \(c \in C\) is performed in a coordinated way with the submodels for the other clusters.

The BFC algorithm implicitly satisfies the NAC \((4)\) on the \(x\)- and \(y\)-variables for the set of stages in set \(T_2\), respectively, by using the engine of choice for solving the MINLP scenario cluster models themselves \((3)\) at each iteration.

On the other hand, the NAC \((4)\) on the \(x\)-variables for the stages in \(T_1\) are relaxed from the original DEM \((2)\), but their satisfaction is performed algorithmically.

Additionally, the NAC \((4)\) on the \(y\)-variables for the stages in \(T_1\) are not considered until a TNF integer set is reached.
Scenario cluster submodels (3) $\forall c \in C$ can be solved in a coordinated serial mode and even having the deterministic MINLP engine of choice as a 'subroutine' for the approach. Better proposal:

Running in **coordinated parallel** mode the $|C|$ BnC phases for solving the scenario cluster MINLP submodels (3),

So, **imbedding the decomposition scheme into MINLP solver** for strong **interfacing** between submodels branching. Then, sharing:

- Fixing and bound tightening on the 0-1 $x$ vars and the continuous $y$ vars for the stages in set $\mathcal{T}_1$.
- Valid inequalities generation and appending where only vars for the stages in set $\mathcal{T}_1$ are involved.
The branching on the 0-1 $x$-variables related to the scenario groups $g \in G^t$ for all stages in set $\mathcal{T}_1$ (i.e., stages up to break stage $t^*$) should be coordinated while solving in parallel the $|C|$ MINLP submodels, such that the replicas (one per each submodel) of each of those $x$-variables should be branched in the same direction (either 0 or 1) to algorithmically satisfying the related NAC (4).

**Solution’ feasibility.** For each coordinated feasible solution to the scenario cluster submodels (3) $\forall c \in C$, a feasible solution for the original RN model (2) could be obtained by fixing the $x$-variables to their current 0-1 values, such that the NAC (4) on the $y$-variables for the stages in $\mathcal{T}_1$ are implicitly satisfied in the resulting NLP model:
Solution’ feasibility

\[ z_{RN}^{feas} = \max \eta \]

s.t.  
\[ \sum_{g \in G} w^g f_0^g (x^g', y^g' \forall g' \in A^g) \geq \eta \]
\[ f_i^g (x^g', y^g' \forall g' \in A^g) \leq 0 \quad \forall i \in I, g \in G \]
\[ x^g = \hat{x}^g \]
\[ y^g \in \mathbb{R}^{+n_y(g)} \]
\[ y^g \in \mathbb{R}^{+n_y(g)} \quad \forall g \in G. \]
Dynamic math optim under uncertainty
A MINLP Branch-and-Fix Coordination: Exact parallel algo
Asynchronous parallel algs:
  - Fix-and-Relax Coordination
  - Stochastic Dynamic Programming

Some computational experience for MILP.
Large scale instances
Suggestion for efficiency increasing
Problem #1. Randomly generated instances

- **BFC-SDC-TC strategy.** $|P^{T-1}| = 2$, $|P^T| = 4$, $|T| = 5$.
- HW/SW: WS Precision T7600, Linux (version Debian2.6.32-48) with 64 bits, processor Intel(R) Xeon(R) CPU E5-2630 @ 2.3 GHz, 12 Gb of RAM and 8 threads.
- C++ experimental code.
- $|\Omega| = 574$ (844) scenarios.
- P8 (P9): $m=22650$ (32184) cons, $n01=6580$ (9390) 0-1 vars, $nc=14480$ (20550) continuous vars.
- Elapsed time: 291 (1208) secs, $GG\% = 0.13$ ($<0.01$) optimality gap versus plain use CPLEX.
- CPLEX v.12.5 (default options) obtains optimal sol but in 6 hours cannot prove it.
Problem #2. Tactical supply chain planning under uncertainty

- **Risk averse SDC.**
- PC with a 2.5 GHz dual-core Intel Core i5 processor, 8 Gb of RAM and the operating system was OS X 10.9.
- **Metaheuristic S-SDP (Serial Stochastic Dynamic Programming).**
- P3 (P12): $T=7$ (10) periods, $E=3$ (3) stages, $\Omega=64$ (512) scenarios.
- P3 (P12): $m=7827$ (212544) cons, $n01=1408$ (36864) 0-1 vars, $nc=4653$ (124596) continuous vars.
- P3 (P12): $nprob=544$ (1258) MILP subproblems, elapsed time=610 (6540.56) secs, $GG\%=1.82$ (.) optimality gap versus plain use CPLEX v.12.5,
- CPLEX v12.5: P3 (P12): Elapsed time 17480 secs (> 8 hours).
Risk neutral.

XPRESS, CPLEX v12.2 failed to find a feas sol in several hours.

SUN WS, 2.6Ghz, 16Gb RAM, Linux.

BFC-MS.

$|\Omega|=1000$ scenarios, $m=98456$ cons, $n01=34680$ 0-1 vars, $nc=22221$ continuous vars.

Elapsed time = 182 secs.

Optimal soln.

Ref. LFE-Garín-Merino-Pérez, COR’12.
Problem #4. Multi-stage location-assignment problem under uncertainty

- Pure combinatorial model: SLOC.
- Risk neutral. Metaheuristic FRC.
- HW/SW: Core 2 Duo, 2.60Ghz, 3Gb RAM. C++ v6.0.
- CPLEX v12.3 Running out-of-memory at 38705 sec elapsed time (no soln).
- Pilot case: $|I| = 15$ facilities, $|J| = 75$ customers, $|T| = 4$.
- $|\Omega| = 98$, $m = 203384$ cons, $n = 175525$ 0-1 vars.
- FRC elapsed time=4040 sec. Optimality GAP = 2.81 %.
Problem #5. Copper extraction planning under uncertainty in future copper prices

CPLEX v.12.2 has problems for obtaining optimal soln. Decomposition approaches are a must.

- Many risk averse measures.
- Big HW/SW platform: 2 quad-core Xeon E5450.3 3Ghz 64-bit processors with 6Mb of cache each.
- $|\Omega|=45$ scenarios, $m=480490$ cons, $n01=167951$ 0-1 vars, $nc=823$ continuous vars.
- Elapsed time: from 398 to 29892 secs.
- Optimal soln.
Problem #6. Airline Revenue Management

- Risk neutral.
- PC, 2.33hz, 8.5Gb RAM, Linux.
- Continuous model.
- Plain CPLEX v9 failed to find a feas sol in several hours.
- Metaheuristic SDP.
  - $|\Omega|=6561$ scenarios, $m=2296300$ cons, $nc=2624400$ continuous vars.
  - Elapsed time = 71 secs.
  - Optimality GAP=1.22 %.
Problem #7. Production planning under uncertainty

- Serial, Inner, Outer, Outer-Inner **asynchronous parallel** metaheuristic SDP (Stochastic Dynamic Programming).
- **Risk neutral.** CPLEX v12.5 for solving independent scenario cluster MILP subproblems.
- **MPI:** Message Passing Interface.
- **Big computing cluster,** SGI/IZO-SGIker at UPV/EHU, used 16 xeon cores (8 or 12 threads each), 48 Gb each.
- $T=16$ periods, Randomly generated $\Omega=7766$.
- P86 (P85): $m=5.56 \ (57.8)$ **M** cons, $n01=1.41 \ (15.04)$ **M** 0-1 varis, $nc=3.49 \ (38.5)$ **M** continuous vars.
- P86 (P85): $nprob=28997 \ (5177)$ MILP subproblems, elapsed time=978 (26180) secs.
- **P-SDP:** incumbent sol value with $GG\% = 0.16 \ (.).$ Efficiency=61.64 (89.15) %.
- Plain CPLEX: running out of memory (35Gb) after 8274 secs, sol value with $OG\% = 0.78 \ (.).$ optimality gap.
- Dynamic math optim under uncertainty
- A MINLP Branch-and-Fix Coordination: Exact parallel algo
- Asynchronous parallel algos:
  - Fix-and-Relax Coordination
  - Stochastic Dynamic Programming
- Some computational experience for MILP.
  Large scale instances
- Suggestion for efficiency increasing
Suggestion for efficiency increasing

- Imbedding decomposition scheme into MINLP solver.
- Running in coordinated parallel mode the BnC phases of the scenario cluster MINLP submodels (3).
- Strong interfacing between submodels branching. So, sharing:
  - Fixings and bound tightening on the 0-1 x vars and the continuous y vars for the stages in set $\mathcal{T}_1$.
  - Valid inequalities generation and appending where only vars for the stages in set $\mathcal{T}_1$ are evolved.
Our references on decomposition for large scale Stochastic MILP problems


Our references on decomposition for large scale Stochastic MILP problems (c)


