Analysis of Stochastic Problem Decomposition Algorithms in Computational Grids

Jesús María Latorre

Santiago Cerisola

Andrés Ramos

Rafael Palacios

Contents

- Introduction
- Computational grids
- Distributed Benders decomposition
- Problem decomposition algorithms
- Theoretical calculations
- Results
- Conclusions

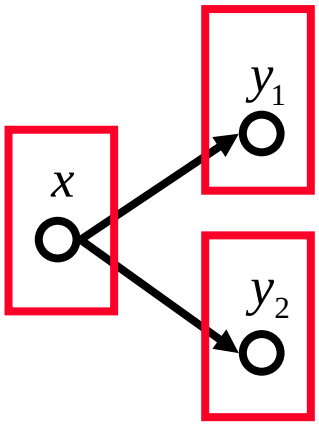
Introduction (I)

- Stochastic programming
  - Optimization problems dealing with uncertainty
  - Scenario tree representation
  - Usually large problems
- Problem decomposition:
  - Decreases individual problem size
  - Increases resolution time
  - Several approaches
    - Benders decomposition
    - Lagrangian relaxation

Introduction (II)

- Benders decomposition:

$$\begin{aligned} \min \quad & c^1 x + p_1 c_1^2 y_1 + p_2 c_2^2 y_2 \\ \text{s.t.} \quad & T_1 x + W_1 y_1 \geq h_1 \\ & T_2 x + W_2 y_2 \geq h_2 \\ & x \in X, y_1 \in Y_1, y_2 \in Y_2 \end{aligned}$$



Master problem

$$\begin{aligned} \min \quad & c^1 x + p_1 \theta_1 + p_2 \theta_2 \\ \text{s.t.} \quad & x \in X \end{aligned}$$

$$\theta_1 \geq \dot{\theta}_1^i - T_1 \pi_1^i \left(\dot{x}^i - x \right)$$

$$\theta_2 \geq \dot{\theta}_2^j - T_2 \pi_2^j \left(\dot{x}^j - x \right)$$

$\dot{x}$

θ_1^i, π_1^i

θ_2^j, π_2^j

Subproblems

$$\begin{aligned} \theta_1(\dot{x}) = \min \quad & c_1^2 y_1 \\ \text{s.t.} \quad & W_1 y_1 \geq h_1 - T_1 \dot{x} : \pi_1 \\ & y_1 \in Y_1 \end{aligned}$$

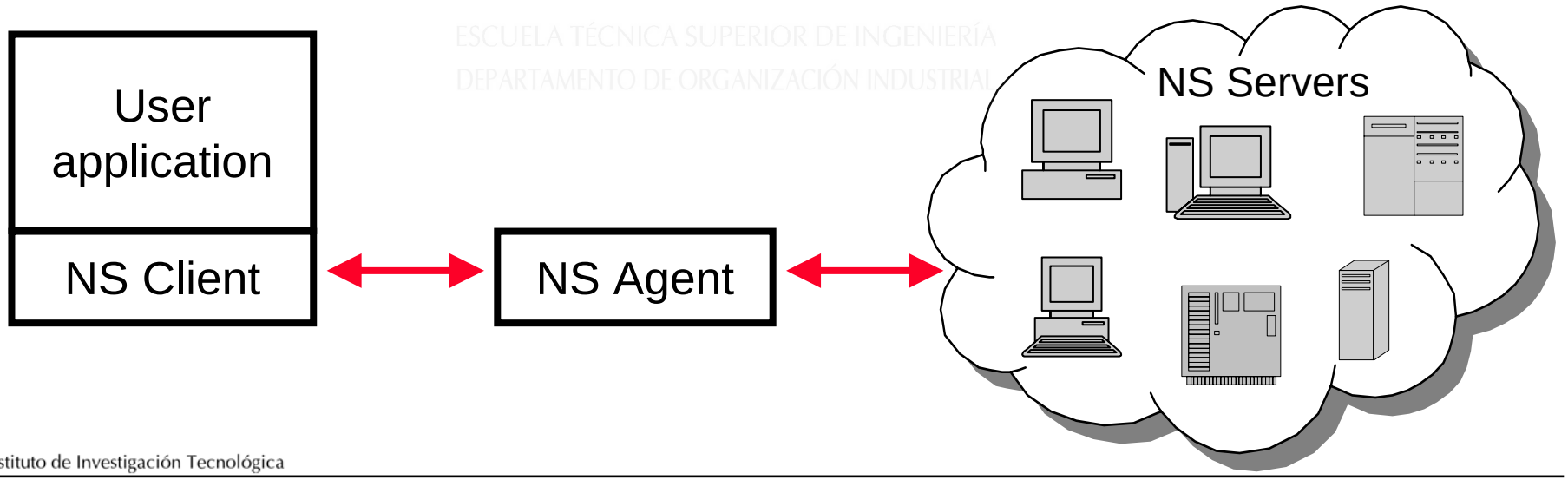
$$\begin{aligned} \theta_2(\dot{x}) = \min \quad & c_2^2 y_2 \\ \text{s.t.} \quad & W_2 y_2 \geq h_2 - T_2 \dot{x} : \pi_1 \\ & y_2 \in Y_2 \end{aligned}$$

Computational grids (I)

- Computational grid:
  - Distributed system
  - Allows sharing the resources:
    - Processing power
    - Data storage
- Characteristics of computational grids:
  - Heterogeneity
  - Unreliability
  - Dynamic availability
  - Geographically disperse
    - Loose communications

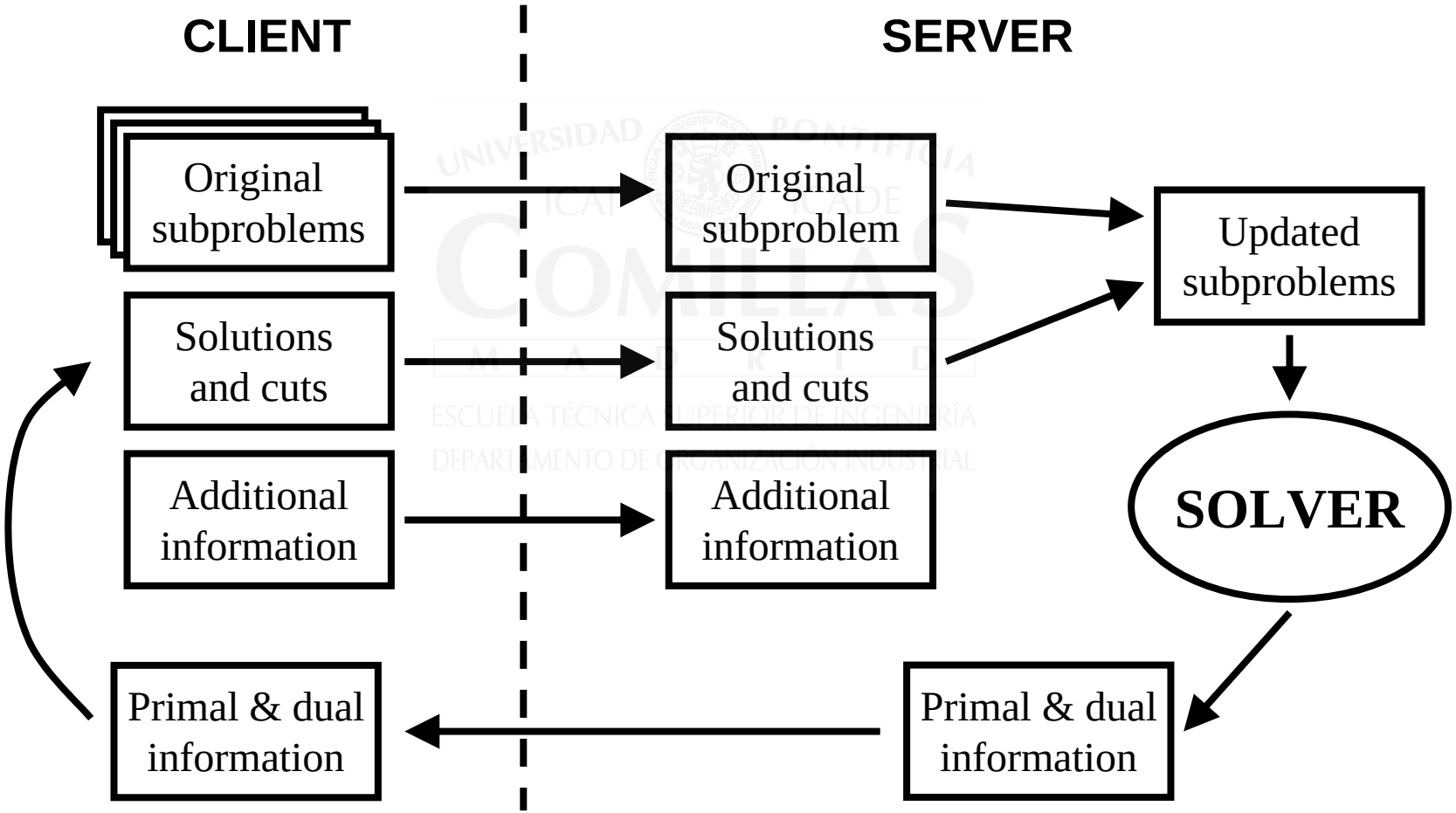
Computational grids (II)

- NetSolve
  - Developed at University of Tennessee
  - Components:
    - Client: interface of the user with the Grid
    - Agent: manages and assigns the resources
    - Server: performs the calculations



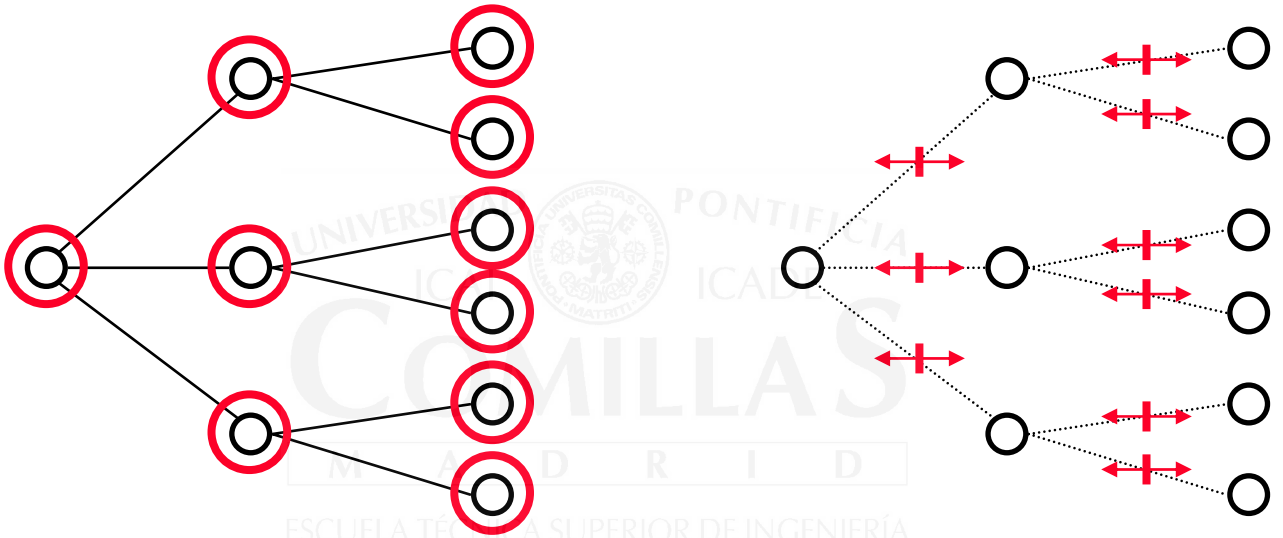
Computational grids (III)

- Information flow in grid environment:



Problem decomposition algorithms (I)

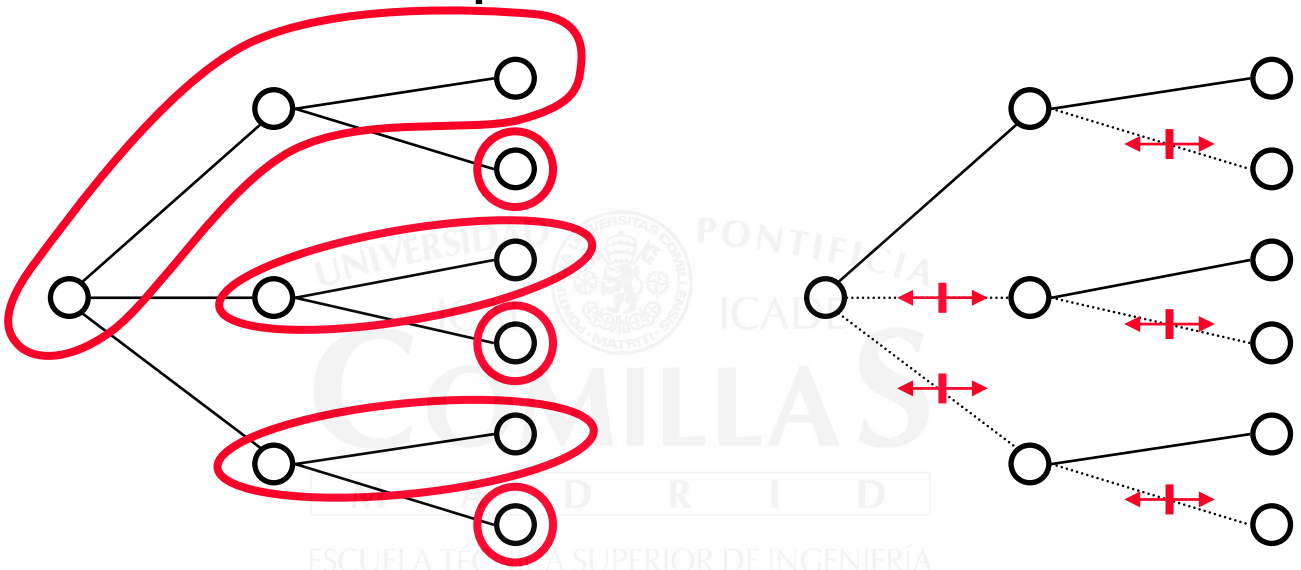
- Node decomposition



- Drawbacks:
  - Large number of problems
  - Infeasible solutions

Problem decomposition algorithms (II)

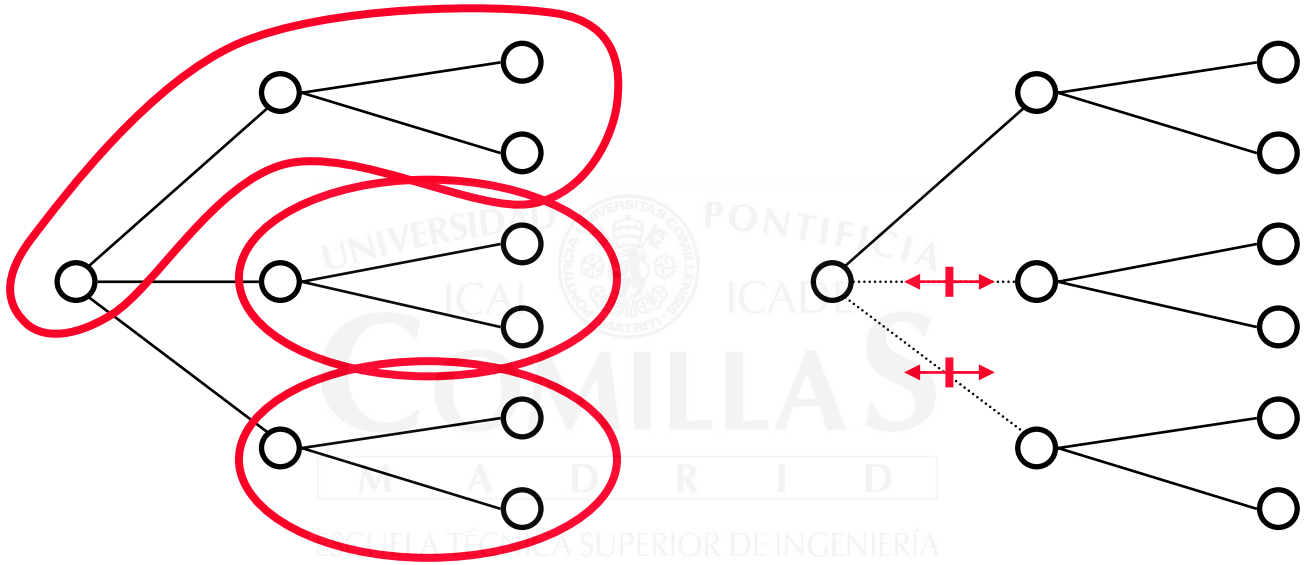
- Scenario decomposition



- SMPS Stoch file industry standard
- Obtains feasible solutions since the beginning

Problem decomposition algorithms (III)

- Sub-tree decomposition



- Advantages:
  - Adjusts problem size to computer resources
  - Allows load balance

Theoretical calculations (I)

- Assumptions
  1. There are available as many processors as needed.
  2. Every node's problem is solved in the same time.
  3. Balanced scenario trees, with the same branching all along the time scope.
  4. Complete recourse problems.
  5. Single root node.
- Considered values:
  - T number of stages.
  - N_t number of nodes at stage t.
  - R number of branches hanging from each node.

Theoretical calculations (II)

- Node decomposition
  - New processors needed at stage t : $N_t - N_{t-1}$
  - Use of these new processors:

$$(T - t - 1)/T$$

  - Average use of processors:

$$\left(\sum_{t=1}^{T} \frac{T - t + 1}{T} (N_t - N_{t-1}) \right) \Big/ \sum_{t=1}^{T} N_t$$

  - Time gain:

$$\left(\sum_{t=1}^{T} N_t \right) \Big/ T$$

  - Total number of processors needed: N_T

Theoretical calculations (III)

- Node decomposition (considering R)
  - Average use of processors: $\frac{1}{T}$
  - Time gain: $\frac{1-R^T}{1-R}\frac{1}{T}$
  - Total number of processors needed: R^{T-1}

Theoretical calculations (IV)

- Scenario decomposition
  - Problems with n nodes when solving step s:

$$N(s,n) = (R-1)\sum_{t=n+1}^{T-s+1} N(s-1,t)$$

  - Processor time used at step s:

$$\sum_{n=1}^{T-s+1} n\,N(s,n)$$

  - Serial resolution time:

$$\sum_{s=1}^{T}\sum_{n=1}^{T-s+1} n\,N(s,n)$$

  - Processor time allocated at step s:

$$(T-s+1)\max_{s'}\left[\sum_{n=1}^{T-s'+1} N(s',n)\right]$$

Theoretical calculations (V)

- Scenario decomposition
  - Processor usage:

$$\sum_{s=1}^{T}\left[\frac{\sum_{n=1}^{T-s+1} n\,N(s,n)}{(T-s+1)\max_{s'}\left[\sum_{n=1}^{T-s'+1} N(s',n)\right]}\right]$$

  - Time gain:

$$\left(\sum_{s=1}^{T}\sum_{n=1}^{T-s+1} n\,N(s,n)\right)\Big/\sum_{n=1}^{T} n$$

  - Total number of processors needed:

$$\max_{s}\left[\sum_{n=1}^{T-s+1} N(s,n)\right]$$

  - Non-recursive expression of

$$N(s,n)=(R-1)^{s-1}\binom{T-n-1}{s-2}$$

Theoretical calculations (VI)

- Sub-tree decomposition
  - Results similar to node decomposition, with some translation.
  - Instead of nodes, sub-trees are considered.
  - Sub-trees take the same time to be solved.
  - The number of sub-trees hanging from each sub-tree is R.
  - The depth of sub-trees is T.
  - At each stage t, there are N_t sub-trees.

Results (I)

- Study case: SSLP
  - Stochastic Server Location Problem
  - Part of SIPLIB (contributed by L. Ntaimo & S. Sen)
  - MIP solved relaxing integer variables
  - Range of sizes (SSLP_m_n_s)
    - m potential server locations: 5, 10, 15
    - n potential clients: 25, 50, 45
    - s scenarios: 5 to 500
- Experimental grid with 3 computers

Results (II)

- Execution times in the computational grid:

Model	Node	Scenario	Sub-tree
SSLP_5_25_50	70	62	3
SSLP_5_25_100	139	129	2
SSLP_10_50_50	252	169	5
SSLP_10_50_100	496	666	16
SSLP_10_50_500	3625	3031	224
SSLP_15_45_5		202	2
SSLP_15_45_10		97	1
SSLP_15_45_15	192	319	3

Results (III)

- Execution in a single computer:

Model	Node	Scenario	Sub-tree
SSLP_5_25_50			
SSLP_5_25_100			
SSLP_10_50_50			
SSLP_10_50_100			
SSLP_10_50_500			
SSLP_15_45_5			
SSLP_15_45_10			
SSLP_15_45_15			

Results (IV)

Conclusions