Multi-stage stochastic programming approach for a lot-sizing problem with remanufacturing

Franco Quezada[1]

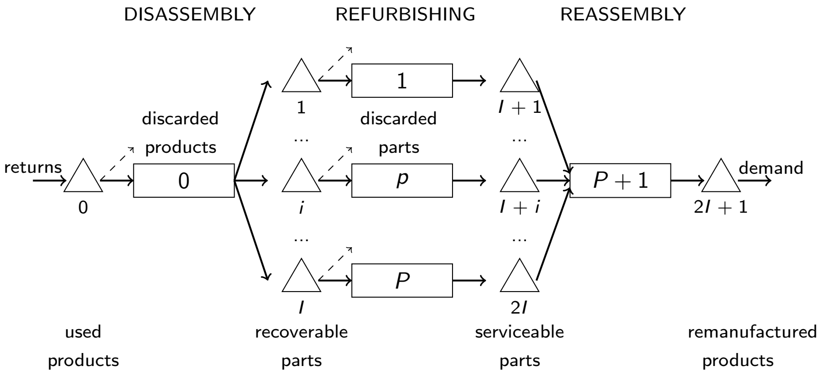
joint work with

Céline Gicquel[2] and Safia Kedad-Sidhoum[3]

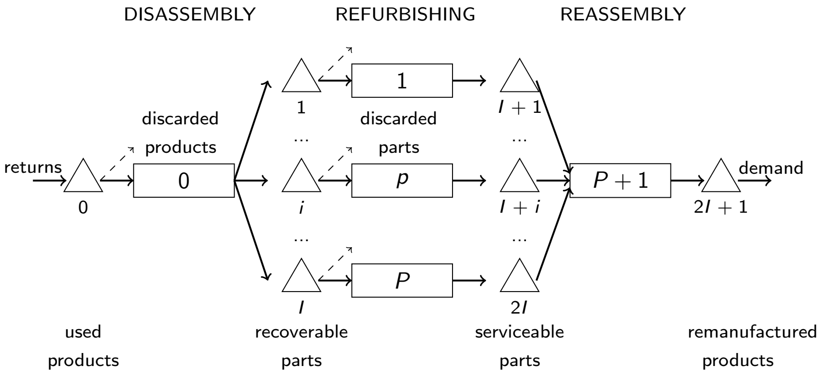
[1]University of Santiago of Chile, Industrial Engineering Department, Chile
[2]Université Paris Saclay, LISN, France
[3]Conservatoire national des arts et métiers, CEDRIC, France

December 13, 2024

A three-echelon remanufacturing system



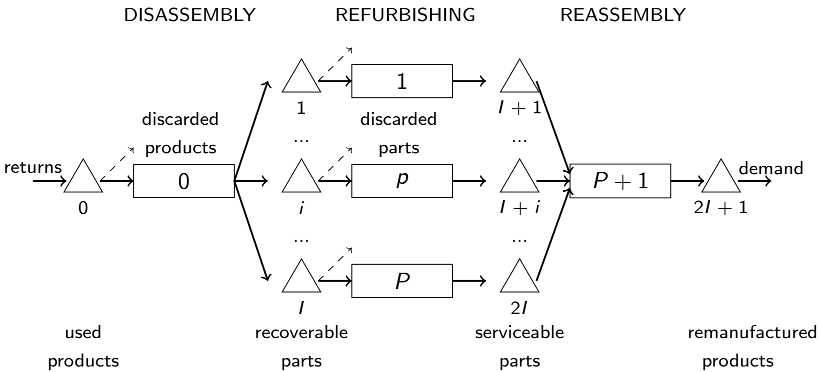
A three-echelon remanufacturing system



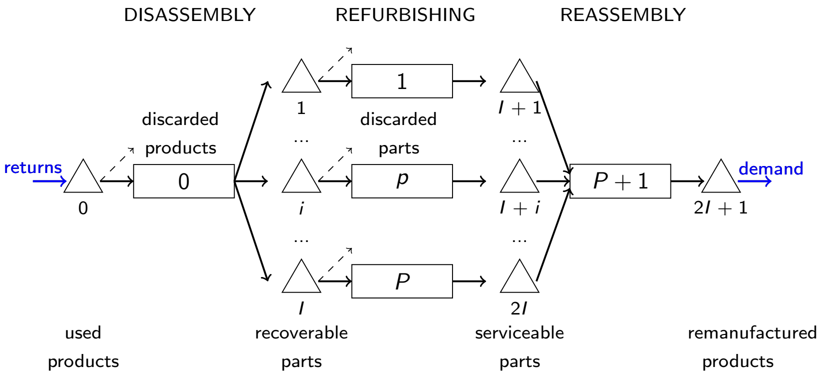
Industrial applications: mobile phones, electrical equipment...

Jayaraman [2006], Franke et al. [2006], Han et al. [2013], Ahn et al. [2011]

Assumptions

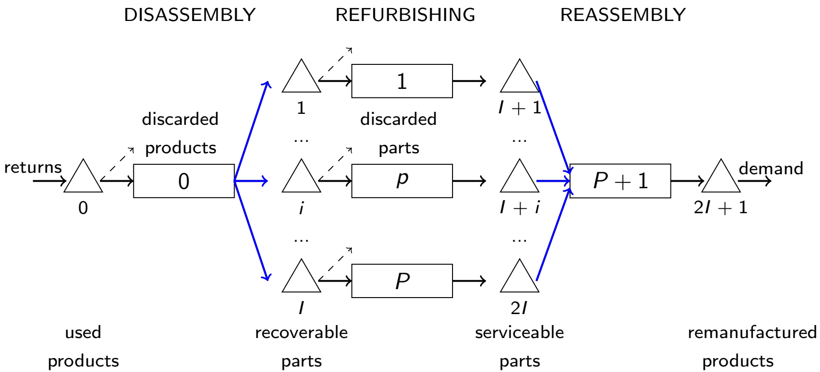


Assumptions



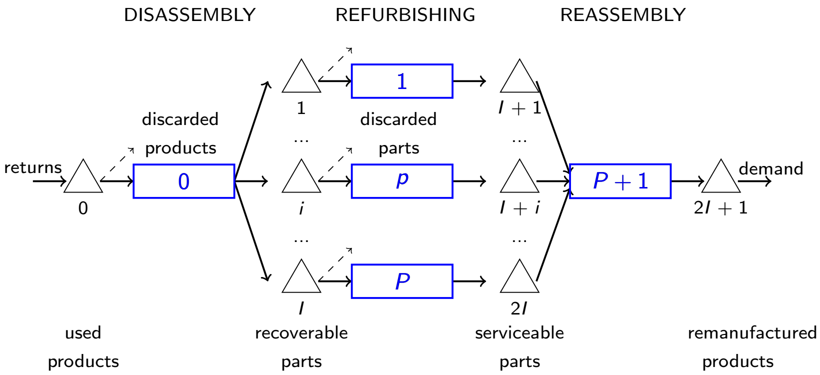
1. a single type of product

Assumptions



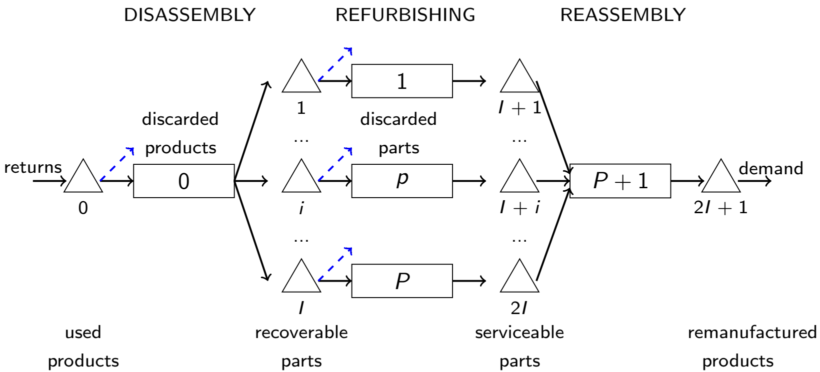
1. a single type of product
2. identical bill-of-materials

Assumptions



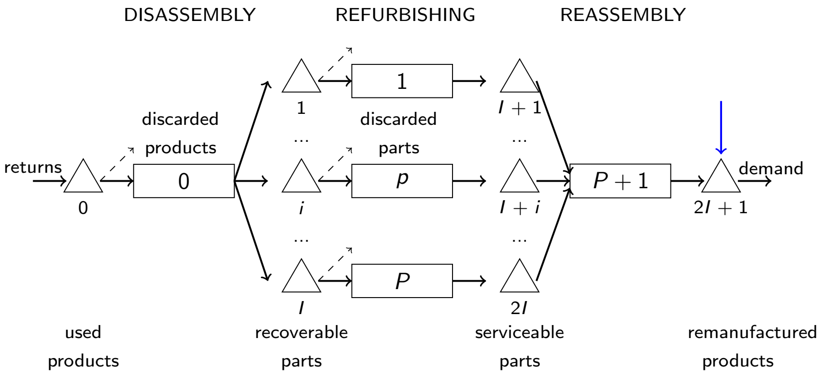
1. a single type of product
2. identical bill-of-materials
3. uncapacitated production processes

Assumptions



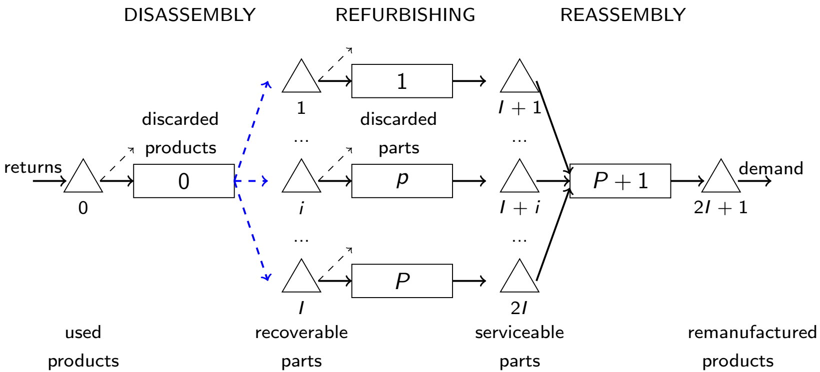
1. a single type of product
2. identical bill-of-materials
3. uncapacitated production processes
4. discard items

Assumptions



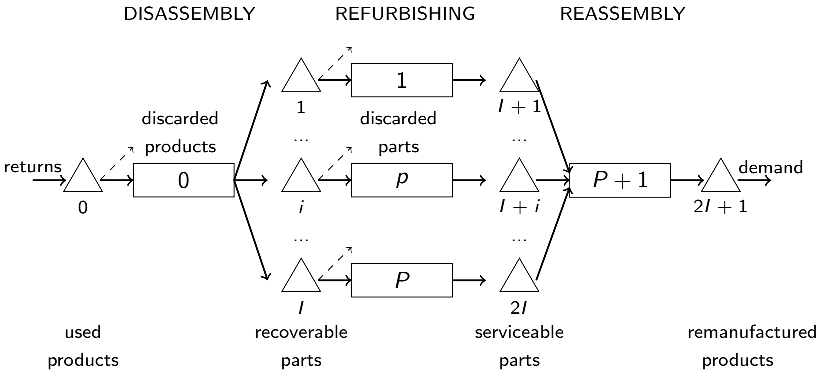
1. a single type of product
2. identical bill-of-materials
3. uncapacitated production processes
4. discard items
5. unmet demand is lost

Assumptions

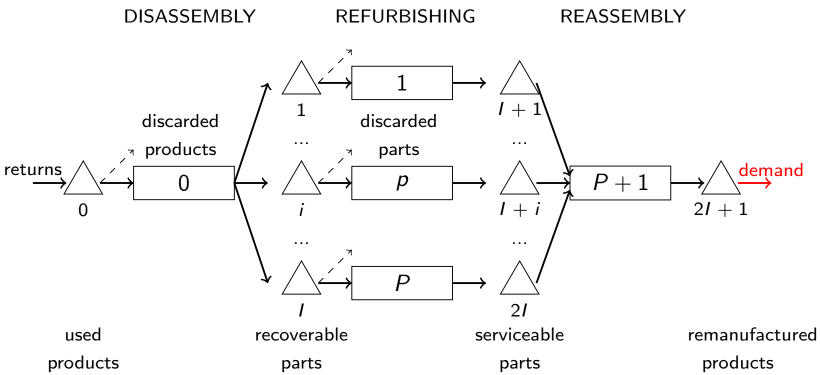


1. a single type of product
2. identical bill-of-materials
3. uncapacitated production processes
4. discard items
5. unmet demand is lost
6. yield of the disassembly process

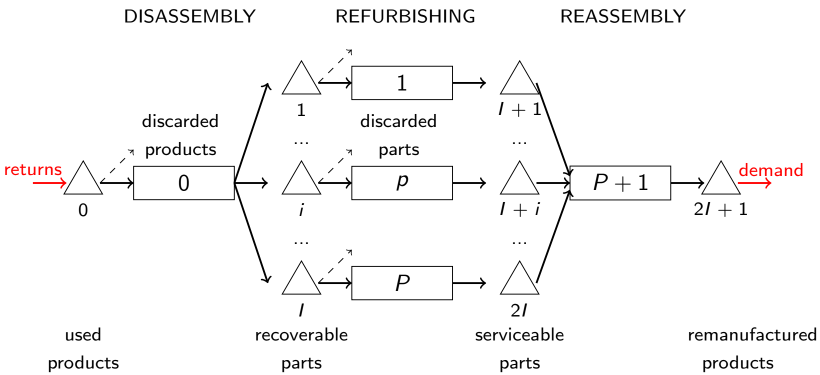
Sources of uncertainty



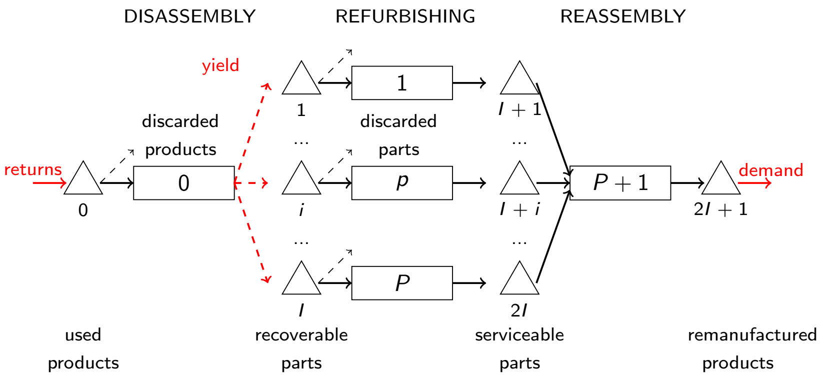
Sources of uncertainty



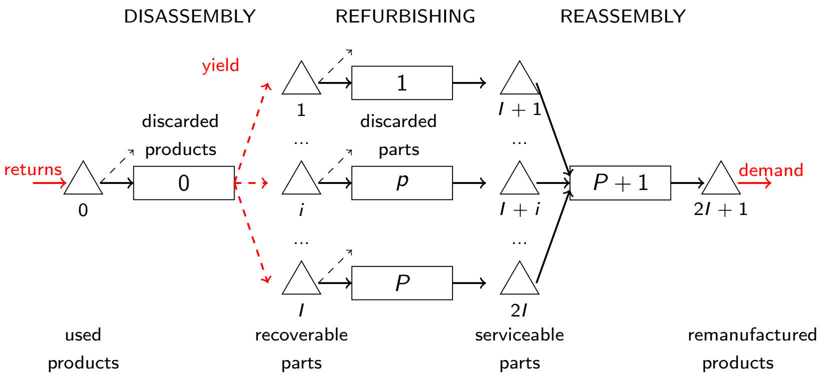
Sources of uncertainty



Sources of uncertainty



Sources of uncertainty



Ilgin and Gupta [2010], Lage Junior and Filho [2012], Lage Junior and Filho [2016]

Multi-stage stochastic programming

Assumption

- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

Multi-stage stochastic programming

Assumption

- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

Observation of random parameters for stage 1

Multi-stage stochastic programming

Assumption

- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

Observation of random parameters for stage 1 $\longrightarrow$ Production planning decisions for stage 1

Multi-stage stochastic programming

Assumption

- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

Observation of random parameters for stage 1 $\longrightarrow$ Production planning decisions for stage 1 $\longrightarrow$ Observation of random parameters for stage 2

Multi-stage stochastic programming

Assumption

- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

Observation of random parameters for stage 1 $\longrightarrow$ Production planning decisions for stage 1 $\longrightarrow$ Observation of random parameters for stage 2 $\longrightarrow$ Production planning decisions for stage 2

Multi-stage stochastic programming

Assumption

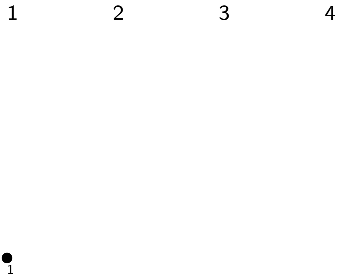
- Not all decisions have to be made before uncertainty realization.
- Some can be postponed to a future point in time.

Dynamic multi-stage decision process

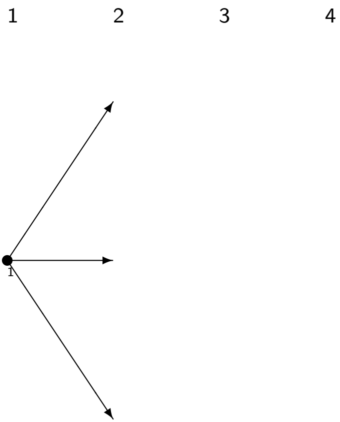
Observation of random parameters for stage 1 $\rightarrow$ Production planning decisions for stage 1 $\rightarrow$ Observation of random parameters for stage 2 $\rightarrow$ Production planning decisions for stage 2 $\rightarrow$ $\cdots$ $\rightarrow$ Observation of random parameters for stage S $\rightarrow$ Productions planning decisions for stage S

Scenario tree

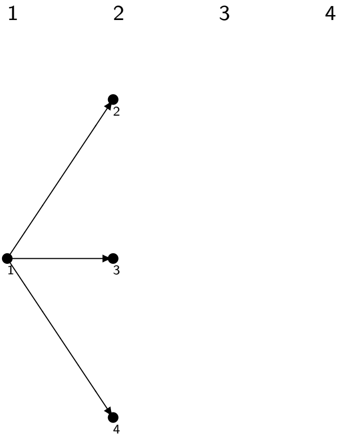
Scenario tree



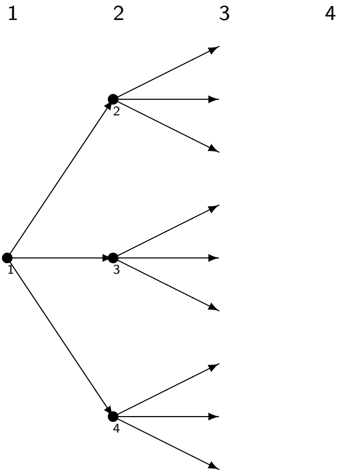
Scenario tree



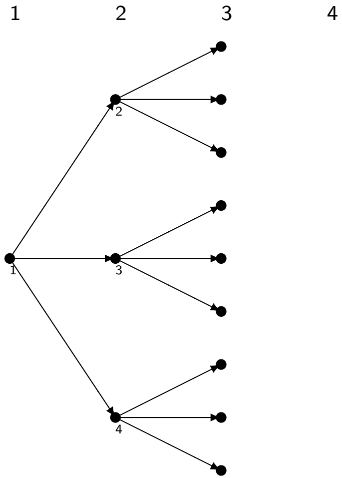
Scenario tree



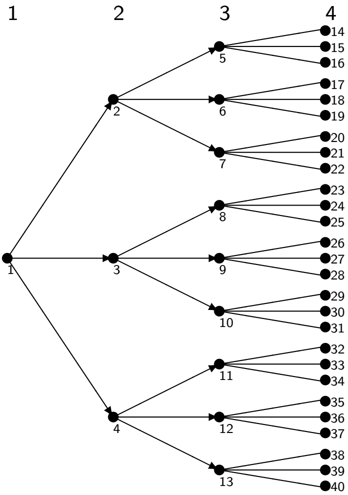
Scenario tree



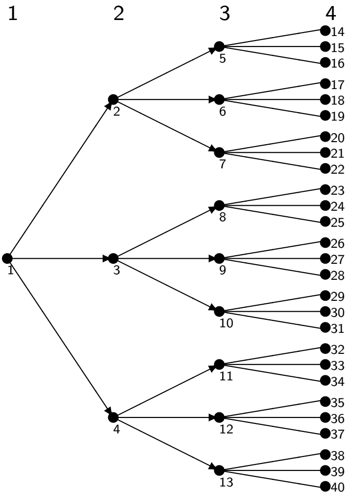
Scenario tree



Scenario tree



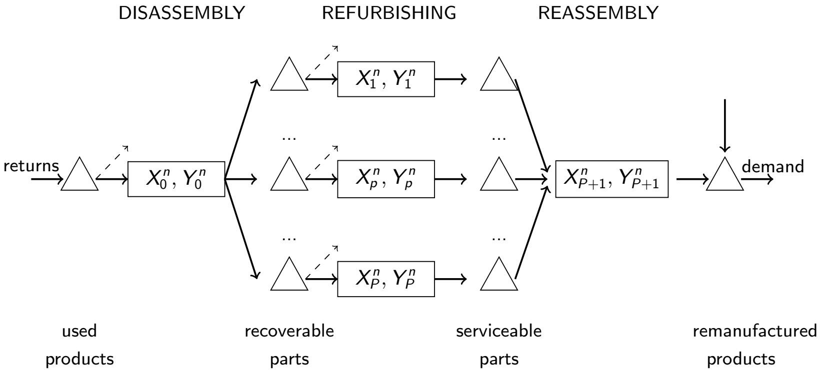
Scenario tree



Scenario tree is defined by a set of $\mathcal{V}$ nodes.

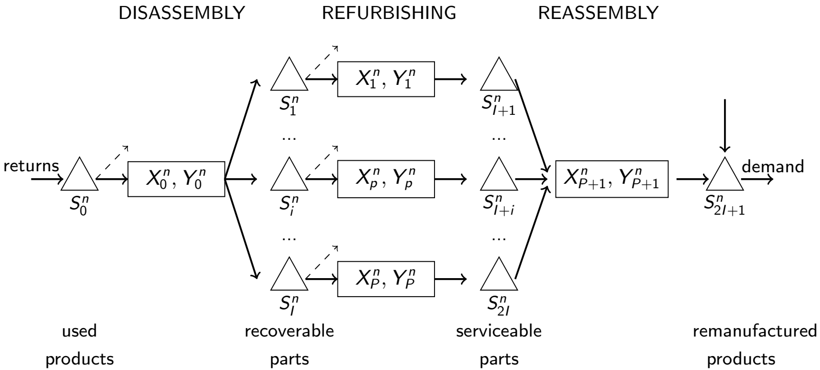
Decision variables

for every node $n \in \mathcal{V}$:



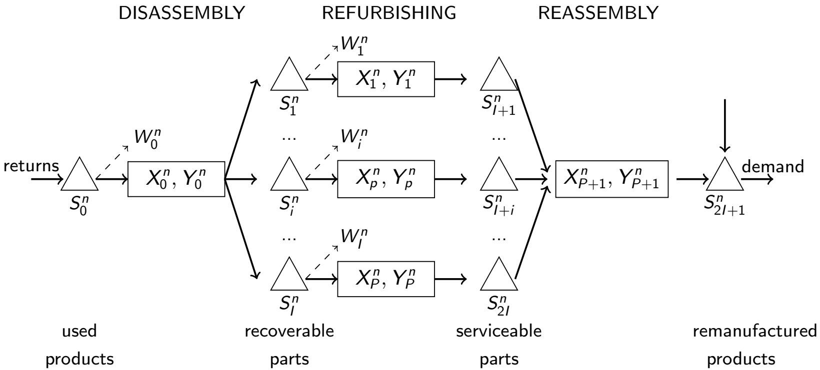
Decision variables

for every node $n \in \mathcal{V}$:



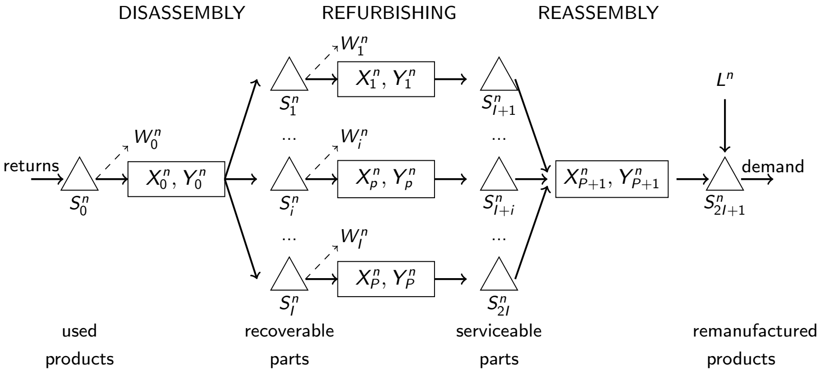
Decision variables

for every node $n \in \mathcal{V}$:



Decision variables

for every node $n \in \mathcal{V}$:



MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \leq M_p^n Y_p^n \qquad \forall p \in \mathcal{J}, \forall n \in \mathcal{V}$$

Setup-production

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \leq M_p^n Y_p^n \qquad \forall p \in \mathcal{J}, \forall n \in \mathcal{V}$$

Setup-production

$$S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n \qquad \forall n \in \mathcal{V}$$

Inventory balance

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \le M_p^n Y_p^n \qquad \forall p \in \mathcal{J}, \forall n \in \mathcal{V}$$

Setup-production

$$S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n \qquad \forall n \in \mathcal{V}$$

$$S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n \qquad \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V}$$

Inventory balance

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \le M_p^n Y_p^n \qquad \forall p\in\mathcal{J}, \forall n\in\mathcal{V}$$

Setup-production

$$S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n \qquad \forall n\in\mathcal{V}$$

$$S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n \qquad \forall i\in\mathcal{I}_r, \forall n\in\mathcal{V}$$

$$S_i^n = S_i^{n-1} + X_{i-l}^n - \alpha_i X_{l+1}^n \qquad \forall i\in\mathcal{I}_s, \forall n\in\mathcal{V}$$

Inventory balance

MILP formulation

$$\min \sum_{n \in \mathcal{V}} \Big(\sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \le M_p^n Y_p^n \qquad \forall p \in \mathcal{J}, \forall n \in \mathcal{V}$$

Setup-production

$$S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n \qquad \forall n \in \mathcal{V}$$

$$S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n \qquad \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V}$$

$$S_i^n = S_i^{n-1} + X_{i-I}^n - \alpha_i X_{I+1}^n \qquad \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V}$$

$$S_{2I+1}^n = S_{2I+1}^{n-1} + X_{I+1}^n - d^n + L^n \qquad \forall n \in \mathcal{V}$$

Inventory balance

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

setup/inventory/discarding/lost sales costs

subject to:

$$X_p^n \le M_p^n Y_p^n \qquad \forall p\in\mathcal{J}, \forall n\in\mathcal{V}$$

Setup-production

$$S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n \qquad \forall n\in\mathcal{V}$$

$$S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n \qquad \forall i\in\mathcal{I}_r, \forall n\in\mathcal{V}$$

$$S_i^n = S_i^{n-1} + X_{i-I}^n - \alpha_i X_{I+1}^n \qquad \forall i\in\mathcal{I}_s, \forall n\in\mathcal{V}$$

$$S_{2I+1}^n = S_{2I+1}^{n-1} + X_{I+1}^n - d^n + L^n \qquad \forall n\in\mathcal{V}$$

Inventory balance

$$S_i^0 = 0 \qquad \forall i\in\mathcal{I}$$

$$S_i^n, W_i^n, L^n \ge 0 \qquad \forall i\in\mathcal{I}, \forall n\in\mathcal{V}$$

$$X_p^n \ge 0,\ Y_p^n \in \{0,1\} \qquad \forall p\in\mathcal{J}, \forall n\in\mathcal{V}$$

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$
\begin{aligned}
& X_p^n \leq M_p^n Y_p^n && \forall p \in \mathcal{J}, \forall n \in \mathcal{V} \\
& S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n \in \mathcal{V} \\
& S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V} \\
& S_i^n = S_i^{n-1} + X_{i-l}^n - \alpha_i X_{l+1}^n && \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V} \\
& S_{2l+1}^n = S_{2l+1}^{n-1} + X_{l+1}^n - d^n + L^n && \forall n \in \mathcal{V} \\
& S_i^0 = 0 && \forall i \in \mathcal{I} \\
& S_i^n, W_i^n, L^n \geq 0 && \forall i \in \mathcal{I}, \forall n \in \mathcal{V} \\
& X_p^n \geq 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}, \forall n \in \mathcal{V}
\end{aligned}
$$

Difficulties:

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$\begin{aligned}
& X_p^n \le M_p^n Y_p^n && \forall p\in\mathcal{J}, \forall n\in\mathcal{V}\\
& S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n\in\mathcal{V}\\
& S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i\in\mathcal{I}_r, \forall n\in\mathcal{V}\\
& S_i^n = S_i^{n-1} + X_{i-l}^n - \alpha_i X_{l+1}^n && \forall i\in\mathcal{I}_s, \forall n\in\mathcal{V}\\
& S_{2l+1}^n = S_{2l+1}^{n-1} + X_{l+1}^n - d^n + L^n && \forall n\in\mathcal{V}\\
& S_i^0 = 0 && \forall i\in\mathcal{I}\\
& S_i^n, W_i^n, L^n \ge 0 && \forall i\in\mathcal{I}, \forall n\in\mathcal{V}\\
& X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p\in\mathcal{J}, \forall n\in\mathcal{V}
\end{aligned}$$

Difficulties:

Dependent demand
→ Dependence between echelons

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$\begin{aligned}
& X_p^n \le M_p^n Y_p^n && \forall p \in \mathcal{J}, \forall n \in \mathcal{V} \\
& S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n \in \mathcal{V} \\
& S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V} \\
& S_i^n = S_i^{n-1} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V} \\
& S_{2I+1}^n = S_{2I+1}^{n-1} + X_{I+1}^n - d^n + L^n && \forall n \in \mathcal{V} \\
& S_i^0 = 0 && \forall i \in \mathcal{I} \\
& S_i^n, W_i^n, L^n \ge 0 && \forall i \in \mathcal{I}, \forall n \in \mathcal{V} \\
& X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}, \forall n \in \mathcal{V}
\end{aligned}$$

Difficulties:

Setup constraints
$\rightarrow$ Poor linear relaxation

Dependent demand
$\rightarrow$ Dependence between echelons

MILP formulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$\begin{aligned}
& X_p^n \le M_p^n Y_p^n && \forall p \in \mathcal{J}, \forall n \in \mathcal{V}\\
& S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n \in \mathcal{V}\\
& S_i^n = S_i^{n-1} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V}\\
& S_i^n = S_i^{n-1} + X_{i-l}^n - \alpha_i X_{l+1}^n && \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V}\\
& S_{2l+1}^n = S_{2l+1}^{n-1} + X_{l+1}^n - d^n + L^n && \forall n \in \mathcal{V}\\
& S_i^0 = 0 && \forall i \in \mathcal{I}\\
& S_i^n, W_i^n, L^n \ge 0 && \forall i \in \mathcal{I}, \forall n \in \mathcal{V}\\
& X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}, \forall n \in \mathcal{V}
\end{aligned}$$

Difficulties:

Setup constraints
→ Poor linear relaxation

Dependent demand
→ Dependence between echelons

Solutions:

Echelon stock = Total inventory of an item in the system, as such or as a component of other items.

Pochet and Wolsey [2006]

New valid inequalities

Echelon stock reformulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$\begin{aligned}
& X_p^n \leq M_p^n Y_p^n && \forall p \in \mathcal{J}, \forall n \in \mathcal{V} \\
& S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n \in \mathcal{V} \\
& E_i^n = E_i^{n-1} + \pi_i^n \alpha_i X_0^n - d^n + L^n - W_i^n && \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V} \\
& E_i^n = E_i^{n-1} + X_{i-I}^n - d^n + L^n && \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V} \\
& E_{2I+1}^n = E_{2I+1}^{n-1} + X_{I+1}^n - d^n + L^n && \forall n \in \mathcal{V}
\end{aligned}$$

Echelon stock reformulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

subject to:

$$
\begin{array}{lr}
X_p^n \le M_p^n Y_p^n & \forall p \in \mathcal{J}, \forall n \in \mathcal{V} \\
S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n & \forall n \in \mathcal{V} \\
E_i^n = E_i^{n-1} + \pi_i^n \alpha_i X_0^n - d^n + L^n - W_i^n & \forall i \in \mathcal{I}_r, \forall n \in \mathcal{V} \\
E_i^n = E_i^{n-1} + X_{i-l}^n - d^n + L^n & \forall i \in \mathcal{I}_s, \forall n \in \mathcal{V} \\
E_{2l+1}^n = E_{2l+1}^{n-1} + X_{l+1}^n - d^n + L^n & \forall n \in \mathcal{V} \\
E_i^n - E_{i+l}^n \ge 0 & \forall i \in \mathcal{I} \\
E_{i+l}^n - E_{2l+1}^n \ge 0 & \forall i \in \mathcal{I} \\
E_i^n \ge 0 & \forall i \in \mathcal{I}, \forall n \in \mathcal{V} \\
E_i^0 = 0 & \forall i \in \mathcal{I} \\
W_i^n, L^n \ge 0 & \forall i \in \mathcal{I}, \forall n \in \mathcal{V} \\
X_p^n \ge 0,\ Y_p^n \in \{0,1\} & \forall p \in \mathcal{J}, \forall n \in \mathcal{V}
\end{array}
$$

Linking constraints

Echelon stock reformulation

$$\min \sum_{n\in\mathcal{V}} \Big(\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n \Big)$$

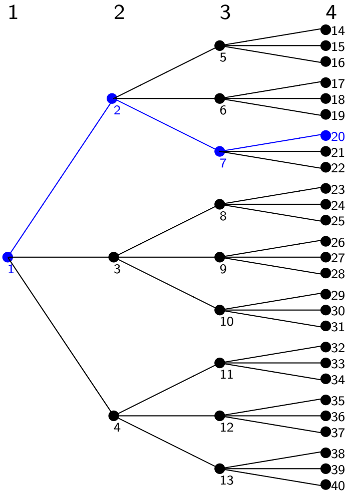
subject to:

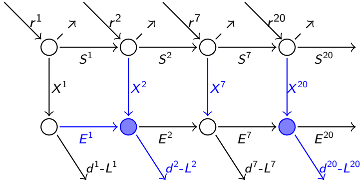
$$\begin{aligned}
&X_p^n \le M_p^n Y_p^n && \forall p\in\mathcal{J}, \forall n\in\mathcal{V}\\
&S_0^n = S_0^{n-1} + r^n - X_0^n - W_0^n && \forall n\in\mathcal{V}\\
&E_i^n = E_i^{n-1} + \pi_i^n \alpha_i X_0^n - d^n + L^n - W_i^n && \forall i\in\mathcal{I}_r, \forall n\in\mathcal{V}\\
&E_i^n = E_i^{n-1} + X_{i-l}^n - d^n + L^n && \forall i\in\mathcal{I}_s, \forall n\in\mathcal{V}\\
&E_{2l+1}^n = E_{2l+1}^{n-1} + X_{l+1}^n - d^n + L^n && \forall n\in\mathcal{V}\\
&E_i^n - E_{i+l}^n \ge 0 && \forall i\in\mathcal{I}\\
&E_{i+l}^n - E_{2l+1}^n \ge 0 && \forall i\in\mathcal{I}\\
&E_i^n \ge 0 && \forall i\in\mathcal{I}, \forall n\in\mathcal{V}\\
&E_i^0 = 0 && \forall i\in\mathcal{I}\\
&W_i^n, L^n \ge 0 && \forall i\in\mathcal{I}, \forall n\in\mathcal{V}\\
&X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p\in\mathcal{J}, \forall n\in\mathcal{V}
\end{aligned}$$

$P+1$ independent single-item single-echelon lot-sizing problems with lost sales

} Linking constraints

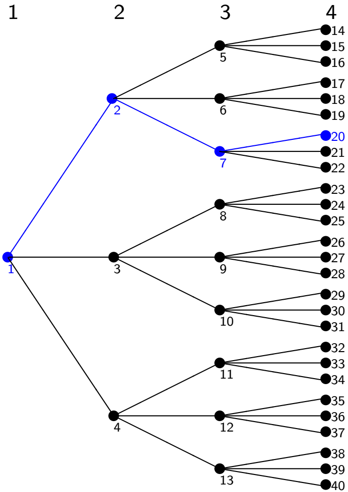
Path inequalities

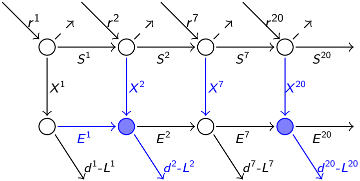




Given $k = 1$ and a subset $U = \{2, 20\}$.

Path inequalities

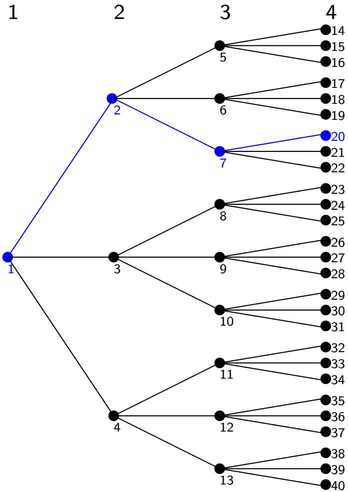


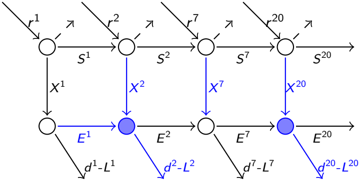


Given $k = 1$ and a subset $U = \{2, 20\}$.

$$E^1 + X^2 + X^7 + X^{20} \geq (d^2 - L^2) + (d^{20} - L^{20})$$

Path inequalities



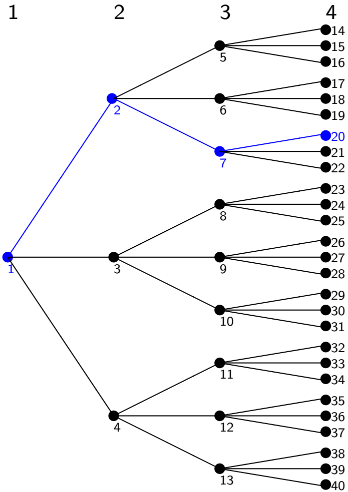


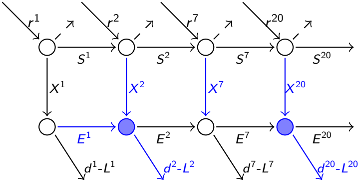
Given $k = 1$ and a subset $U = \{2, 20\}$.

$$E^1 + X^2 + X^7 + X^{20} \geq (d^2 - L^2) + (d^{20} - L^{20})$$

$$E^k + Y^2(d^2 + d^{20}) + Y^7(d^{20}) + Y^{20}(d^{20}) \geq \sum_{t \in U} (d^n - L^n)$$

Path inequalities





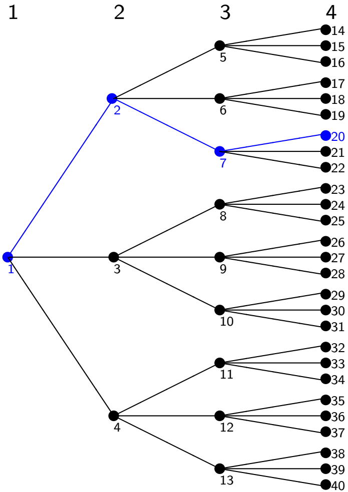
Given $k = 1$ and a subset $U = \{2, 20\}$.

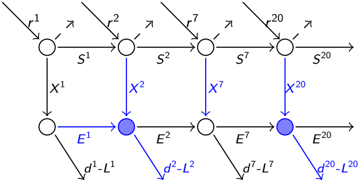
$$E^1 + X^2 + X^7 + X^{20} \geq (d^2 - L^2) + (d^{20} - L^{20})$$

$$E^k + Y^2(d^2 + d^{20}) + Y^7(d^{20}) + Y^{20}(d^{20}) \geq \sum_{t \in U} (d^n - L^n)$$

$$E^k + \sum_{k+1 \leq t \leq t^*} Y^n \sum_{v \in U: v \geq t} d^v \geq \sum_{t \in U} (d^n - L^n)$$

Path inequalities





Given $k = 1$ and a subset $U = \{2, 20\}$.

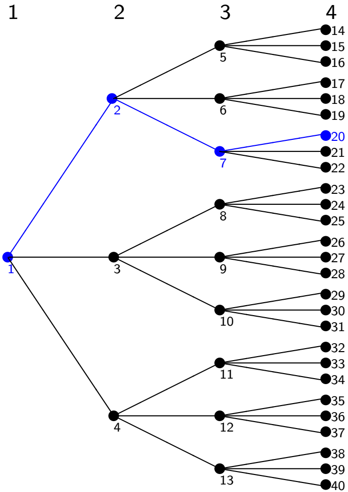
$$E^1 + X^2 + X^7 + X^{20} \geq (d^2 - L^2) + (d^{20} - L^{20})$$

$$E^k + Y^2(d^2 + d^{20}) + Y^7(d^{20}) + Y^{20}(d^{20}) \geq \sum_{t \in U} (d^n - L^n)$$

$$E^k + \sum_{k+1 \leq t \leq t^*} Y^n \sum_{v \in U : v \geq t} d^v \geq \sum_{t \in U} (d^n - L^n)$$

Loparic et al. [2001]

Path inequalities



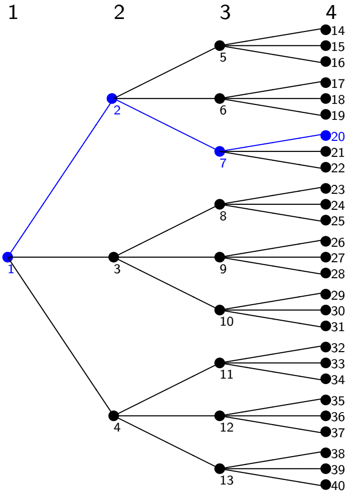
Let k be a non-leaf node and l be a leaf node. Let $U \in \mathcal{P}(k,l)$.

$$E^k_{p+l} \geq \sum_{n \in U} \left[d^n \left(1 - \sum_{\nu \in \mathcal{P}(k,n)} Y^{\nu}_p - L^n\right)\right]$$

where:

- $\mathcal{P}(n,m)$: path between node n and m in the scenario tree

Path inequalities



Let k be a non-leaf node and l be a leaf node. Let $U \in \mathcal{P}(k,l)$.

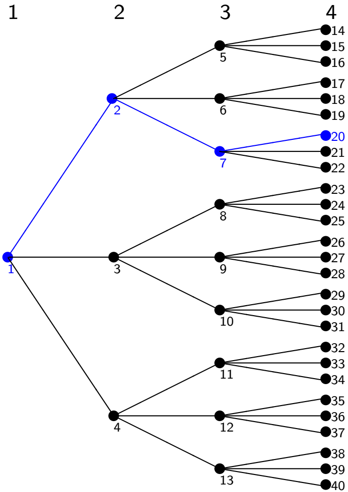
$$E^k_{p+l} \geq \sum_{n \in U} \left[d^n (1 - \sum_{\nu \in \mathcal{P}(k,n)} Y^\nu_p - L^n) \right]$$

where:

- $\mathcal{P}(n,m)$: path between node n and m in the scenario tree

Separation
Polynomial in $\mathcal{O}(|\mathcal{V}|^2)$

Path inequalities



Let k be a non-leaf node and l be a leaf node. Let $U \in \mathcal{P}(k,l)$.

$$E^k_{p+l} \geq \sum_{n \in U} \left[d^n \left(1 - \sum_{\nu \in \mathcal{P}(k,n)} Y^\nu_p - L^n\right)\right]$$

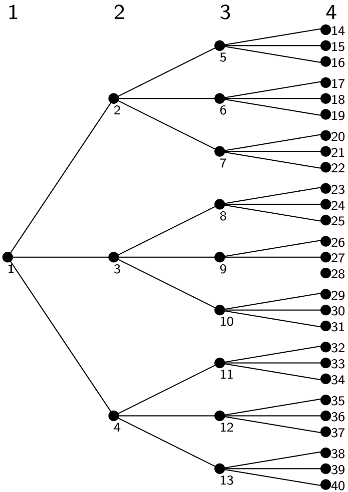
where:

- $\mathcal{P}(n,m)$: path between node n and m in the scenario tree

Separation
Polynomial in $\mathcal{O}(|\mathcal{V}|^2)$

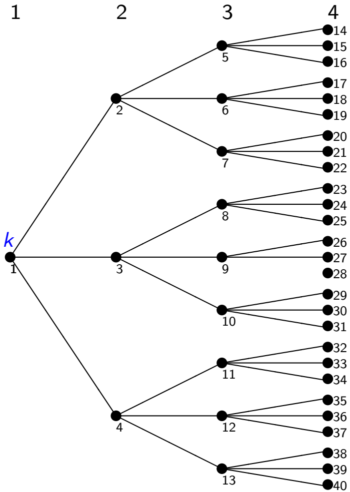
Key point in practice
Careful selection of the valid inequalities to be added to the formulation

Cutting-planes generation algorithm



Two main ingredients:

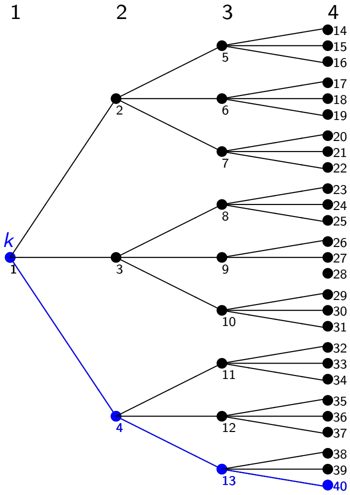
Cutting-planes generation algorithm



Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

Cutting-planes generation algorithm

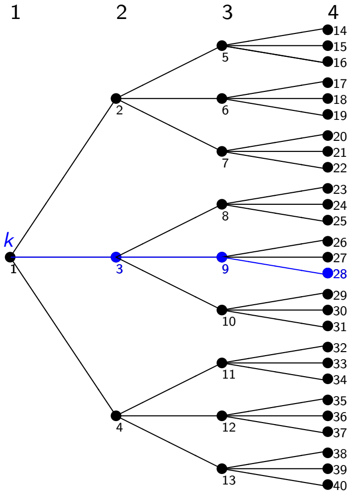


Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

$\rightarrow$ the most violated valid inequality.

Cutting-planes generation algorithm

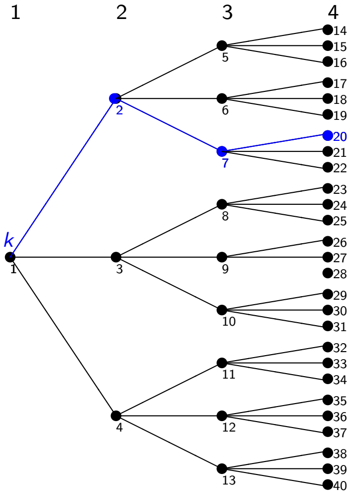


Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

→ the most violated valid inequality.

Cutting-planes generation algorithm

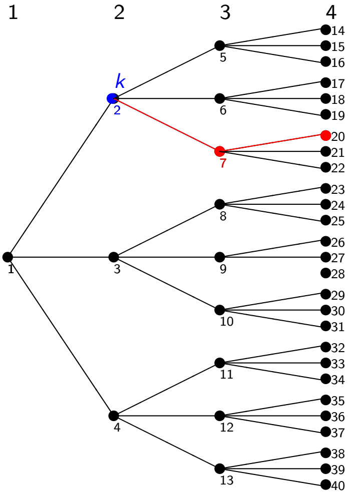


Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

$\rightarrow$ the most violated valid inequality.

Cutting-planes generation algorithm

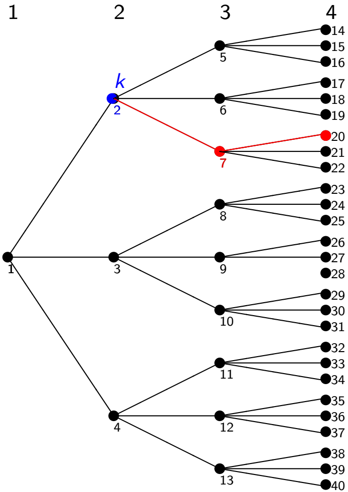


Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

- $\rightarrow$ the most violated valid inequality.
- $\rightarrow$ avoiding the addition of inequalities involving similar subsets of set-up variables.

Cutting-planes generation algorithm



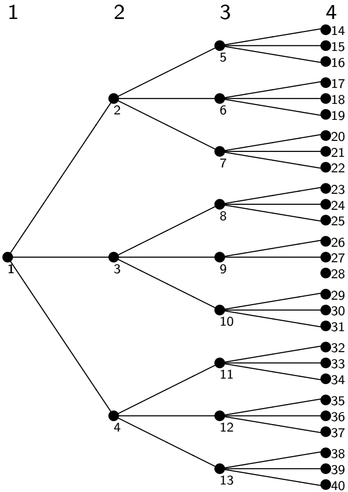
Two main ingredients:

For each $k \in \mathcal{V}$, we search for:

- $\rightarrow$ the most violated valid inequality.
- $\rightarrow$ avoiding the addition of inequalities involving similar subsets of set-up variables.

Advantages:

Cutting-planes generation algorithm



Two main ingredients:

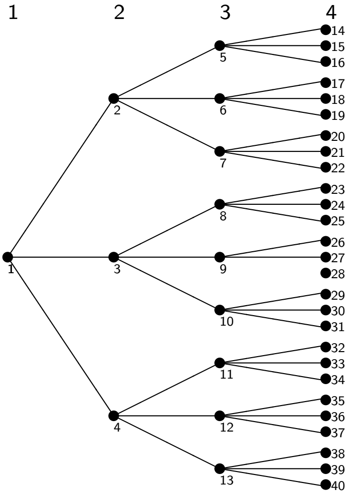
For each $k \in \mathcal{V}$, we search for:

- → the most violated valid inequality.
- → avoiding the addition of inequalities involving similar subsets of set-up variables.

Advantages:

- → all valid inequalities are still potentially considered for inclusion in the formulation.

Cutting-planes generation algorithm



Two main ingredients:

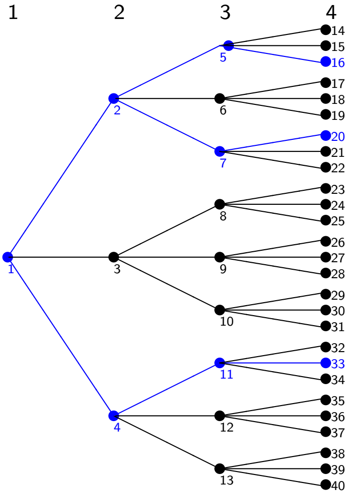
For each $k \in \mathcal{V}$, we search for:

- → the most violated valid inequality.
- → avoiding the addition of inequalities involving similar subsets of set-up variables.

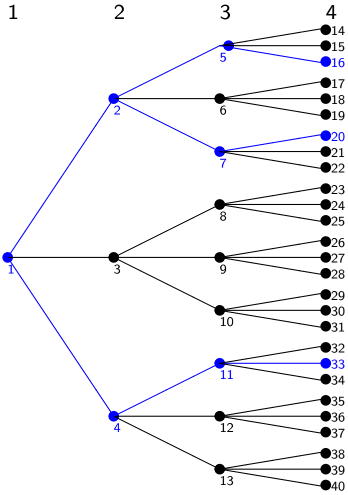
Advantages:

- → all valid inequalities are still potentially considered for inclusion in the formulation.
- → the separation problem is solved exactly.

From Path to Tree inequalities

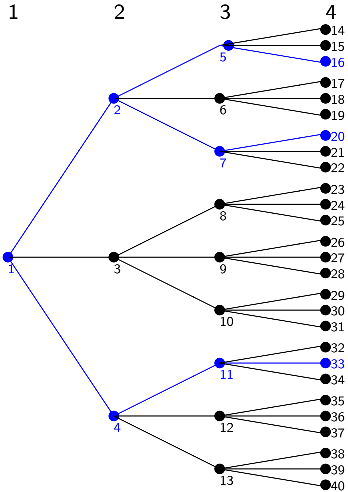


From Path to Tree inequalities



We follow the procedure proposed by Guan et al. [2009]

From Path to Tree inequalities

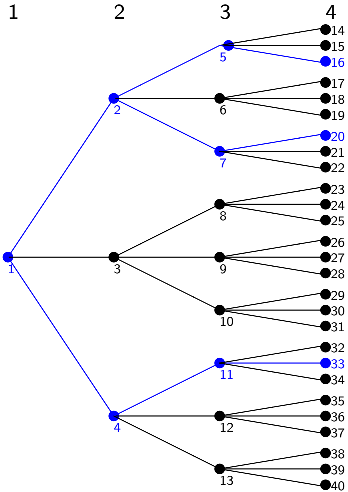


We follow the procedure proposed by Guan et al. [2009]

Let $k \in \mathcal{V}$ be a non-leaf node and $U \in \mathcal{V}(k)$.

$$EI^k + \sum_{\nu \in U} L^\nu + \sum_{\mu \in \mathcal{V}(k)\setminus\{k\}} \phi^\mu Y^\mu \geq \sum_{\nu \in U_{\sigma_{|\mathcal{L}(k)|}}} d^\nu$$

From Path to Tree inequalities



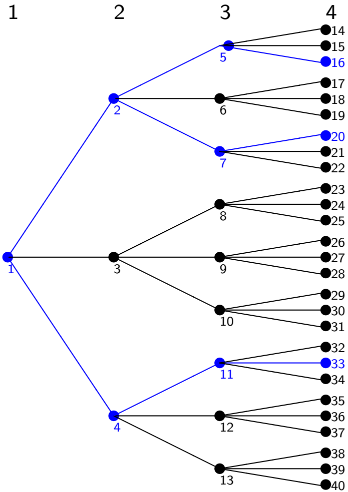
We follow the procedure proposed by Guan et al. [2009]

Let $k \in \mathcal{V}$ be a non-leaf node and $U \in \mathcal{V}(k)$.

$$EI^k + \sum_{\nu \in U} L^\nu + \sum_{\mu \in \mathcal{V}(k)\setminus\{k\}} \phi^\mu Y^\mu \geq \sum_{\nu \in U_{\sigma_{|\mathcal{L}(k)|}}} d^\nu$$

Separation
Exact separation: very long computation times

From Path to Tree inequalities



We follow the procedure proposed by Guan et al. [2009]

Let $k \in \mathcal{V}$ be a non-leaf node and $U \in \mathcal{V}(k)$.

$$EI^k + \sum_{\nu \in U} L^\nu + \sum_{\mu \in \mathcal{V}(k) \setminus \{k\}} \phi^\mu Y^\mu \geq \sum_{\nu \in U_{\sigma_{|\mathcal{L}(k)|}}} d^\nu$$

Separation
Exact separation: very long computation times

$\rightarrow$ Heuristic separation algorithm based on a neighborhood search

Instances Generation and Branch-and-cut algorithms

Random generation based on the numerical values used by [Ahn *et al*, 2011] and [Jayaraman, 2006]

Instances Generation and Branch-and-cut algorithms

Random generation based on the numerical values used by [Ahn *et al*, 2011] and [Jayaraman, 2006]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$

Instances Generation and Branch-and-cut algorithms

Random generation based on the numerical values used by [Ahn *et al*, 2011] and [Jayaraman, 2006]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 16 different structures of scenario tree resulting in a total of 1440 instances
- Time limit of 900 seconds.

Instances Generation and Branch-and-cut algorithms

Random generation based on the numerical values used by [Ahn *et al*, 2011] and [Jayaraman, 2006]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 16 different structures of scenario tree resulting in a total of 1440 instances
- Time limit of 900 seconds.

Branch-and-Cut algorithms:

- CPLEX: the generic branch-and-cut algorithm embedded in CPLEX 12.8.
- Path and Tree: a customized branch-and-cut algorithm using the (k, U) Path inequalities and the newly introduced (k, U) Tree inequalities to strengthen the echelon-stock formulation.

Numerical results

Instances		CPLEX default				Path and Tree			
I	Nodes	Gap_{LP}	Gap_{MIP}	Time	# Opt	Gap_{LP}	Gap_{MIP}	Time	# Opt
5	381								
	1022								
10	381								
	1022								
	1365								

- Total of 3600 randomly generated instances
- Average values for 720 randomly generated instances
- Resolution with CPLEX 12.6.1 on a PC running under Windows 10, Intel Core i7, 8 GB of RAM
- Time limit : 900s

- Results published in *Computers & Operations Research*. Quezada et al. [2020].

Numerical results

Instances		CPLEX default				Path and Tree			
I	Nodes	Gap_{LP}	Gap_{MIP}	Time	# Opt	Gap_{LP}	Gap_{MIP}	Time	# Opt
5	381	11.33	0.15	786.34	119				
	1022	11.04	0.45	900.29	0				
10	381	7.92	0.30	897.49	9				
	1022	10.17	3.11	901.38	0				
	1365	15.67	11.02	900.49	0				

- Total of 3600 randomly generated instances
- Average values for 720 randomly generated instances
- Resolution with CPLEX 12.6.1 on a PC running under Windows 10, Intel Core i7, 8 GB of RAM
- Time limit : 900s

- Results published in *Computers & Operations Research*. Quezada et al. [2020].

Numerical results

Instances		CPLEX default				Path and Tree			
I	Nodes	Gap_{LP}	Gap_{MIP}	Time	# Opt	Gap_{LP}	Gap_{MIP}	Time	# Opt
5	381	11.33	0.15	786.34	119	1.67	0.05	564.96	340
	1022	11.04	0.45	900.29	0	1.71	0.31	900.24	3
10	381	7.92	0.30	897.49	9	1.20	0.19	817.73	103
	1022	10.17	3.11	901.38	0	1.38	0.84	900.91	0
	1365	15.67	11.02	900.49	0	1.84	1.11	900.85	0

- Total of 3600 randomly generated instances
- Average values for 720 randomly generated instances
- Resolution with CPLEX 12.6.1 on a PC running under Windows 10, Intel Core i7, 8 GB of RAM
- Time limit : 900s

- Results published in *Computers & Operations Research*. Quezada et al. [2020].

SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, ..., T$$

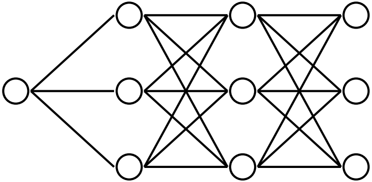
SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, \ldots, T$$

- Recombining scenario tree



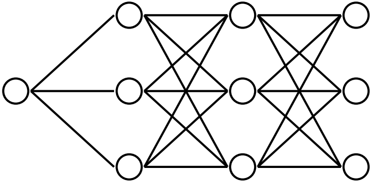
SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, \ldots, T$$

- Recombining scenario tree



- Polyhedral recourse function in LP setting.
$\rightarrow$ Benders' cuts.

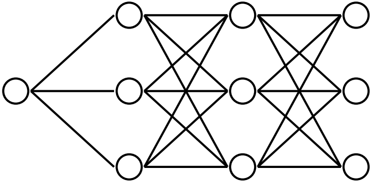
SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, \ldots, T$$

- Recombining scenario tree



- Polyhedral recourse function in LP setting.
$\rightarrow$ Benders' cuts.

However, we are trying to solve an MILP.

Zou et al. [2019] introduced the SDDiP algorithm.

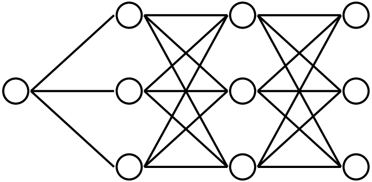
SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, \ldots, T$$

- Recombining scenario tree



- Polyhedral recourse function in LP setting.
$\rightarrow$ Benders' cuts.

However, we are trying to solve an MILP.

Zou et al. [2019] introduced the SDDiP algorithm.

- Key Assumption: Binary state variables

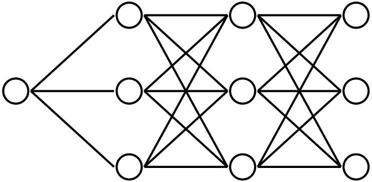
SDDiP algorithm

- First introduced by Pereira and Pinto [1991]

- Key assumption: stage-wise independent process

$$\xi_t \text{ is independent of } \xi_1, \ldots, \xi_{t-1} \quad \text{for } t = 2, ..., T$$

- Recombining scenario tree



- Polyhedral recourse function in LP setting.
→ Benders' cuts.

However, we are trying to solve an MILP.

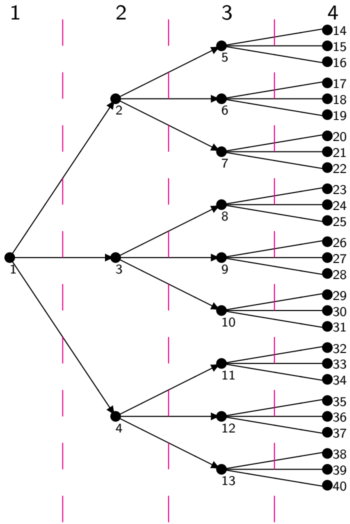
Zou et al. [2019] introduced the SDDiP algorithm.

- Key Assumption: Binary state variables

-**Our contribution**:

→ We propose a new extension of the SDDiP algorithm for solving multistage stochastic lot-sizing problems.

Dynamic programming formulation: a full decomposition



Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n + \sum_{m \in \mathcal{C}(n)} \rho^{nm} Q^m(S^n)$$

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n + \sum_{m \in \mathcal{C}(n)} \rho^{nm} Q^m(S^n)$$

subject to:

$$\begin{aligned}
& X_p^n \leq M_p^n Y_p^n && \forall p \in \mathcal{J} \\
& S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n && \\
& S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r \\
& S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s \\
& S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n && \\
& S_i^n \geq 0 && \forall i \in \mathcal{I} \\
& L^n \geq 0 && \\
& X_p^n \geq 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}
\end{aligned}$$

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n + \underbrace{\sum_{m\in\mathcal{C}(n)} \rho^{nm} Q^m(S^n)}_{\mathcal{Q}^n}$$

subject to:

$$\begin{aligned}
& X_p^n \le M_p^n Y_p^n && \forall p \in \mathcal{J} \\
& S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n && \\
& S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r \\
& S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s \\
& S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n && \\
& S_i^n \ge 0 && \forall i \in \mathcal{I} \\
& L^n \ge 0 && \\
& X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}
\end{aligned}$$

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n + \underbrace{\sum_{m \in \mathcal{C}(n)} \rho^{nm} Q^m(S^n)}_{\mathcal{Q}^n = \mathcal{Q}^t}$$

subject to:

$$\begin{aligned}
& X_p^n \leq M_p^n Y_p^n && \forall p \in \mathcal{J} \\
& S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n && \\
& S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r \\
& S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s \\
& S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n && \\
& S_i^n \geq 0 && \forall i \in \mathcal{I} \\
& L^n \geq 0 && \\
& X_p^n \geq 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}
\end{aligned}$$

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n + \underbrace{\sum_{m \in \mathcal{C}(n)} \rho^{nm} Q^m(S^n)}_{\mathcal{Q}^n = \mathcal{Q}^t}$$

subject to:

$$\begin{aligned}
& X_p^n \leq M_p^n Y_p^n && \forall p \in \mathcal{J} \\
& S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n && \\
& S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r \\
& S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s \\
& S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n && \\
& S_i^n \geq 0 && \forall i \in \mathcal{I} \\
& L^n \geq 0 && \\
& X_p^n \geq 0, Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}
\end{aligned}$$

Decomposes the original problem into a series of single-node sub-problems

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \underbrace{\sum_{p\in\mathcal{J}} f_p^n Y_p^n + \sum_{i\in\mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i\in\mathcal{I}_r\cup\{0\}} q_i^n W_i^n + g^n X_0^n}_{F^n} + \underbrace{\sum_{m\in\mathcal{C}(n)} \rho^{nm} Q^m(S^n)}_{\mathcal{Q}^n = \mathcal{Q}^t}$$

subject to:

$$\begin{aligned}
& X_p^n \le M_p^n Y_p^n && \forall p \in \mathcal{J}\\
& S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n &&\\
& S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n && \forall i \in \mathcal{I}_r\\
& S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n && \forall i \in \mathcal{I}_s\\
& S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n &&\\
& S_i^n \ge 0 && \forall i \in \mathcal{I}\\
& L^n \ge 0 &&\\
& X_p^n \ge 0,\ Y_p^n \in \{0,1\} && \forall p \in \mathcal{J}
\end{aligned}$$

Decomposes the original problem into a series of single-node sub-problems

Dynamic programming formulation: a full decomposition

For each node $n \in \mathcal{V}$:

$$Q^n(s^{a^n}) := \min \underbrace{\sum_{p \in \mathcal{J}} f_p^n Y_p^n + \sum_{i \in \mathcal{I}} h_i^n S_i^n + l^n L^n + \sum_{i \in \mathcal{I}_r \cup \{0\}} q_i^n W_i^n + g^n X_0^n}_{F^n} + \underbrace{\sum_{m \in \mathcal{C}(n)} \rho^{nm} Q^m(S^n)}_{\mathcal{Q}^n = \mathcal{Q}^t}$$

subject to:

$$\mathcal{X}^n \begin{cases} X_p^n \leq M_p^n Y_p^n & \forall p \in \mathcal{J} \\ S_0^n = S_0^{a^n} + r^n - X_0^n - W_0^n & \\ S_i^n = S_i^{a^n} + \pi_i^n \alpha_i X_0^n - X_i^n - W_i^n & \forall i \in \mathcal{I}_r \\ S_i^n = S_i^{a^n} + X_{i-I}^n - \alpha_i X_{I+1}^n & \forall i \in \mathcal{I}_s \\ S_{2I+1}^n = S_{2I+1}^{a^n} + X_{I+1}^n - d^n + L^n & \\ S_i^n \geq 0 & \forall i \in \mathcal{I} \\ L^n \geq 0 & \\ X_p^n \geq 0,\ Y_p^n \in \{0,1\} & \forall p \in \mathcal{J} \end{cases}$$

Decomposes the original problem into a series of single-node sub-problems

Reformulation to use SDDiP

For each node $n \in \mathcal{V}$:

$$Q^n(u^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(u^n)$$

subject to: $$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n$$

Reformulation to use SDDiP

For each node $n \in \mathcal{V}$:

$$Q^n(u^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(u^n)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n$$

$$S_i^n = \sum_{\lambda \in \mathcal{B}} 2^\lambda u_i^{n,\lambda} \qquad \forall i \in \mathcal{I}$$

$$S_i^{a^n} = \sum_{\lambda \in \mathcal{B}} 2^\lambda z_i^{n,\lambda} \qquad \forall i \in \mathcal{I}$$

Binary approximation

$$u_i^{n,\lambda} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B}$$

$$z_i^{n,\lambda} \in (0,1) \qquad \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B}$$

Reformulation to use SDDiP

For each node $n \in \mathcal{V}$:

$$Q^n(u^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(u^n)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n$$

$$S_i^n = \sum_{\lambda \in \mathcal{B}} 2^\lambda u_i^{n,\lambda} \qquad \forall i \in \mathcal{I}$$

Binary approximation

$$S_i^{a^n} = \sum_{\lambda \in \mathcal{B}} 2^\lambda z_i^{n,\lambda} \qquad \forall i \in \mathcal{I}$$

$$z_i^{n,\lambda} = u_i^{a^n,\lambda} \qquad \forall \lambda \in \mathcal{B}$$

Copy Constraint

$$u_i^{n,\lambda} \in \{0,1\} \qquad \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B}$$

$$z_i^{n,\lambda} \in (0,1) \qquad \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B}$$

Reformulation to use SDDiP

For each node $n \in \mathcal{V}$:

$$Q^n(u^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(u^n)$$

subject to:

$$
\begin{aligned}
& (X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n && \\
& S_i^n = \sum_{\lambda \in \mathcal{B}} 2^\lambda u_i^{n,\lambda} && \forall i \in \mathcal{I} \\
& S_i^{a^n} = \sum_{\lambda \in \mathcal{B}} 2^\lambda z_i^{n,\lambda} && \forall i \in \mathcal{I} \\
& z_i^{n,\lambda} = u_i^{a^n,\lambda} && \forall \lambda \in \mathcal{B} \\
& u_i^{n,\lambda} \in \{0,1\} && \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B} \\
& z_i^{n,\lambda} \in (0,1) && \forall i \in \mathcal{I}, \forall \lambda \in \mathcal{B}
\end{aligned}
$$

Binary approximation

Copy Constraint

Subproblems involve a larger number of binary variables

Approximate subproblems

For each node $n \in \mathcal{V}$:

$$Q^n(S^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(S^n)$$

subject to:

$$
\begin{aligned}
& (X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n && \\
& \sigma_i^n = S_i^{a^n} && \forall i \in \mathcal{I} \\
& \sigma_i^n \geq 0 && \forall i \in \mathcal{I} \\
& X^n, S^n, W^n, L^n \geq 0;\ Y_n \in \{0,1\} &&
\end{aligned}
$$

Approximate subproblems

For each node $n \in \mathcal{V}$:

$$Q^n(S^{a^n}) := \min F^n(X^n, Y^n, S^n, W^n, L^n) + \mathcal{Q}^t(S^n)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n$$
$$\sigma_i^n = S_i^{a^n} \qquad \forall i \in \mathcal{I}$$
$$\sigma_i^n \geq 0 \qquad \forall i \in \mathcal{I}$$
$$X^n, S^n, W^n, L^n \geq 0; Y_n \in \{0,1\}$$

Approximation of $\mathcal{Q}^t(\cdot)$ by Strengthened Benders' Cuts

Strengthened Benders' Cuts

For each node $n \in \mathcal{V}$:

$$\hat{R}^n(S^{a^n}) := \min_{y,x,s,\sigma} F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{i \in \mathcal{I}} \pi_i^n (S_i^{a^n} - \sigma_i^n) + \psi^t(S^n)$$

subject to:

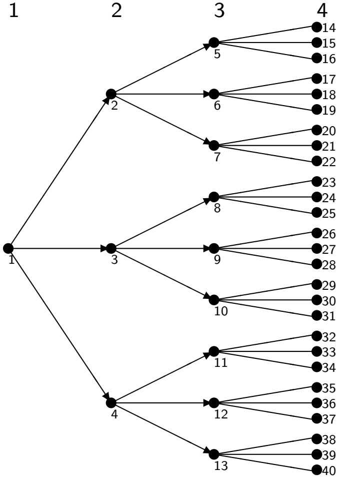
$$(X^n, Y^n, S^n, X^n, L^n) \in \mathcal{X}^n$$

$$\sigma_i^n \geq 0 \qquad \forall i \in \mathcal{I}$$

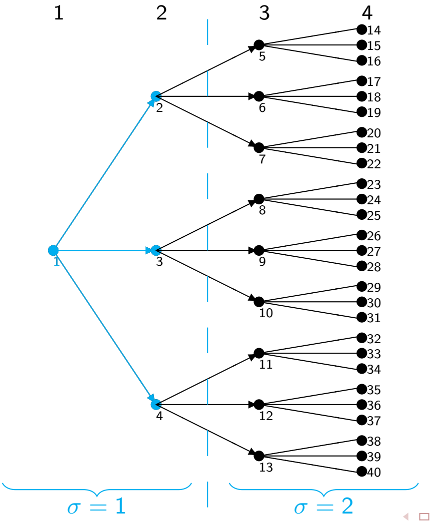
$$\psi_i^n(u^n) := \min\{\theta^n : \theta^n \geq \sum_{m \in \mathcal{C}(n)} \rho^{nm}(v_l^m + (\pi_l^m)^\intercal \sigma^n)\} \qquad \forall l = 1, ..., i-1$$

$$X^n, S^n, W^n, L^n \geq 0;\ Y_n \in \{0, 1\}$$

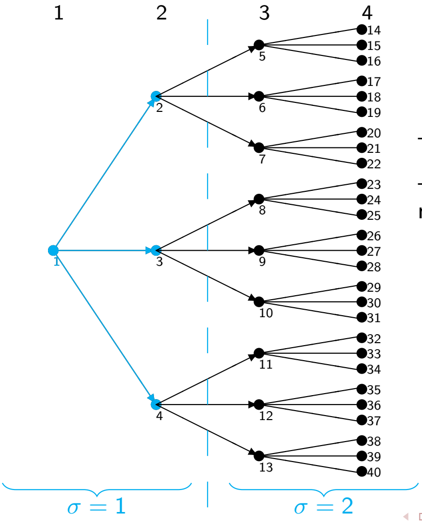
Dynamic programming formulation: a partial decomposition



Dynamic programming formulation: a partial decomposition



Dynamic programming formulation: a partial decomposition



-Set of root nodes $\mho$

-Subtree defined by a set of nodes $\mathcal{W}^\eta$

Dynamic programming formulation: a partial decomposition

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^{n} F^{n}(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) + \sum_{\ell \in \mathfrak{L}(\eta)} \sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^{m}(S^{\ell})$$

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^m(S^{\ell})$$

subject to:

$$\begin{aligned} &(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n && \forall n \in \mathcal{W}^{\eta} \\ &X^n, S^n, W^n, L^n \geq 0 \geq 0,\ Y^n_p \in \{0, 1\} && \forall p \in \mathcal{J}, n \in \mathcal{W}^{\eta} \end{aligned}$$

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^{n} F^{n}(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) + \sum_{\ell \in \mathfrak{L}(\eta)} \underbrace{\sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^{m}(S^{\ell})}_{\mathcal{Q}^{t}}$$

subject to:

$$(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) \in \mathcal{X}^{n} \qquad \forall n \in \mathcal{W}^{\eta}$$

$$X^{n}, S^{n}, W^{n}, L^{n} \geq 0 \geq 0, Y_{p}^{n} \in \{0,1\} \qquad \forall p \in \mathcal{J}, n \in \mathcal{W}^{\eta}$$

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^{n} F^{n}(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) + \sum_{\ell \in \mathfrak{L}(\eta)} \underbrace{\sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^{m}(S^{\ell})}_{\mathcal{Q}^{t}}$$

subject to:

$$\begin{aligned} &(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) \in \mathcal{X}^{n} && \forall n \in \mathcal{W}^{\eta} \\ &X^{n}, S^{n}, W^{n}, L^{n} \geq 0 \geq 0, Y_{p}^{n} \in \{0,1\} && \forall p \in \mathcal{J}, n \in \mathcal{W}^{\eta} \end{aligned}$$

Decomposes the original problem into a series of small stochastic sub-problems

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \underbrace{\sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^m(S^{\ell})}_{\mathcal{Q}^t}$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n \qquad \forall n \in \mathcal{W}^{\eta}$$

$$X^n, S^n, W^n, L^n \geq 0 \geq 0, Y^n_p \in \{0, 1\} \qquad \forall p \in \mathcal{J}, n \in \mathcal{W}^{\eta}$$

Decomposes the original problem into a series of small stochastic sub-problems

Advantages:

- Reduced number of expected cost-to-go functions to approximate.
- Forward solution of better quality.

Dynamic programming formulation: a partial decomposition

For each node $\eta \in \mho$:

$$Q^{\eta}(s^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^{n} F^{n}(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) + \sum_{\ell \in \mathfrak{L}(\eta)} \underbrace{\sum_{m \in \mathcal{C}(\ell)} \rho^{nm} Q^{m}(S^{\ell})}_{\mathcal{Q}^{t}}$$

subject to:

$$(X^{n}, Y^{n}, S^{n}, W^{n}, L^{n}) \in \mathcal{X}^{n} \qquad \forall n \in \mathcal{W}^{\eta}$$

$$X^{n}, S^{n}, W^{n}, L^{n} \geq 0 \geq 0,\ Y^{n}_{p} \in \{0, 1\} \qquad \forall p \in \mathcal{J}, n \in \mathcal{W}^{\eta}$$

Decomposes the original problem into a series of small stochastic sub-problems

Advantages:

- Reduced number of expected cost-to-go functions to approximate.
- Forward solution of better quality.

Disadvantages:

- Larger size of subproblems.

Additional Strengthened Benders' Cuts

Additional Strengthened Benders' Cuts

For each node $\eta \in \mho$:

$$\hat{R}^n(S^{a^n}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \psi^t(S^\ell)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n \qquad \forall n \in \mathcal{W}^\eta$$

$$\sigma_p^\eta = S_p^{a^\eta} \qquad \forall p \in \mathcal{I}$$

$$\sigma_p^\eta \geq 0 \qquad \forall p \in \mathcal{I}$$

$$\psi^\ell(S^\ell) := \min\{\theta^\ell : \theta^\ell \geq \sum_{m \in \mathcal{C}(n)} \rho^{nm}(v_l^m + (\pi_l^m)^\intercal \sigma^\ell)\} \qquad \forall l = 1, ..., i$$

$$X^n, S^n, W^n, L^n \geq 0; \; Y^n \in (0,1) \qquad \forall p \in \mathcal{I}, n \in \mathcal{W}^\eta$$

Additional Strengthened Benders' Cuts

For each node $\eta \in \mho$:

$$\hat{R}^n(S^{a^n}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \psi^t(S^\ell)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n \qquad \forall n \in \mathcal{W}^\eta$$

$$\sigma_p^\eta = S_p^{a^\eta} \qquad \forall p \in \mathcal{I}$$

$$\sigma_p^\eta \geq 0 \qquad \forall p \in \mathcal{I}$$

$$S_p^k \geq \sum_{n \in U} [d^n(1 - \sum_{\nu \in \mathcal{P}(k,n)} Y_p^\nu - L^n)] \qquad \forall U \in \mathcal{P}(k,l), k, l \in \mathcal{W}^\eta$$

$$\psi^\ell(S^\ell) := \min\{\theta^\ell : \theta^\ell \geq \sum_{m \in \mathcal{C}(n)} \rho^{nm}(v_l^m + (\pi_l^m)^\intercal \sigma^\ell)\} \qquad \forall l = 1, ..., i$$

$$X^n, S^n, W^n, L^n \geq 0; \; Y^n \in (0,1) \qquad \forall p \in \mathcal{I}, n \in \mathcal{W}^\eta$$

Additional Strengthened Benders' Cuts

For each node $\eta \in \mho$:

$$\hat{R}^n(S^{a^n}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{p \in \mathcal{I}} \pi_p^n (S_p^{a^n} - \sigma_p^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \psi^t(S^\ell)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n \qquad \forall n \in \mathcal{W}^\eta$$

$$\sigma_p^\eta \geq 0 \qquad \forall p \in \mathcal{I}$$

$$S_p^k \geq \sum_{n \in U} [d^n(1 - \sum_{\nu \in \mathcal{P}(k,n)} Y_p^\nu - L^n)] \qquad \forall U \in \mathcal{P}(k,l), k, l \in \mathcal{W}^\eta$$

$$\psi^\ell(S^\ell) := \min\{\theta^\ell : \theta^\ell \geq \sum_{m \in \mathcal{C}(n)} \rho^{nm}(v_l^m + (\pi_l^m)^\intercal \sigma^\ell)\} \qquad \forall l = 1, ..., i$$

$$X^n, S^n, W^n, L^n \geq 0; \; Y^n \in \{0, 1\} \qquad \forall p \in \mathcal{I}, n \in \mathcal{W}^\eta$$

Additional Strengthened Benders' Cuts

For each node $\eta \in \mho$:

$$\hat{R}^n(S^{a^n}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{p \in \mathcal{I}} \pi_p^n (S_p^{a^n} - \sigma_p^n) + \sum_{\ell \in \mathfrak{L}(\eta)} \psi^t(S^\ell)$$

subject to:

$$(X^n, Y^n, S^n, W^n, L^n) \in \mathcal{X}^n \qquad \forall n \in \mathcal{W}^\eta$$

$$\sigma_p^\eta \geq 0 \qquad \forall p \in \mathcal{I}$$

$$S_p^k \geq \sum_{n \in U} [d^n(1 - \sum_{\nu \in \mathcal{P}(k,n)} Y_p^\nu - L^n)] \qquad \forall U \in \mathcal{P}(k,l), k, l \in \mathcal{W}^\eta$$

$$\psi^\ell(S^\ell) := \min\{\theta^\ell : \theta^\ell \geq \sum_{m \in \mathcal{C}(n)} \rho^{nm}(v_l^m + (\pi_l^m)^\intercal \sigma^\ell)\} \qquad \forall l = 1, ..., i$$

$$X^n, S^n, W^n, L^n \geq 0; Y^n \in \{0, 1\} \qquad \forall p \in \mathcal{I}, n \in \mathcal{W}^\eta$$

Leads to a better approximation of the expected cost-to-go functions in practice.

Illustrative example

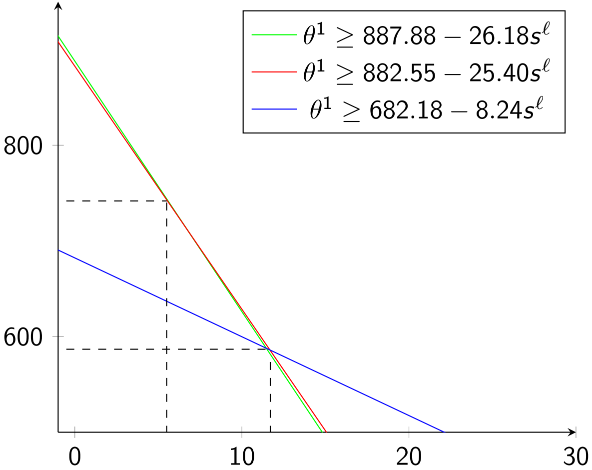


Figure: Illustration of different SB cuts.

Illustration of extSDDiP

Illustration of extSDDiP

At each iteration i:

- Sampling step

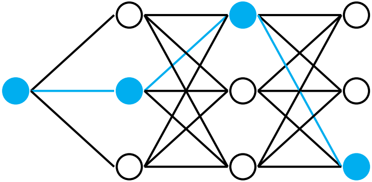


Illustration of extSDDiP

At each iteration i:

- Sampling step
- Forward step

$$\underline{Q}_i^{\eta}(S_i^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathcal{L}(\eta)} \psi_i^t(S^{\ell})$$

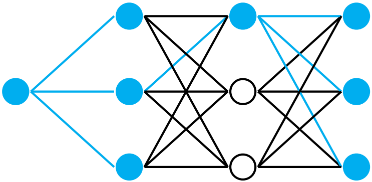
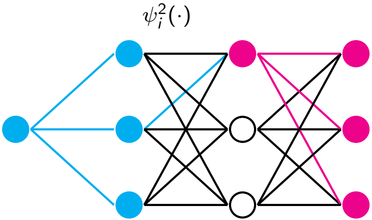


Illustration of extSDDiP

At each iteration i:

- Sampling step
- Forward step

$$\underline{Q}_i^\eta(S_i^{a^\eta}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathcal{L}(\eta)} \psi_i^t(S^\ell)$$



- Backward step

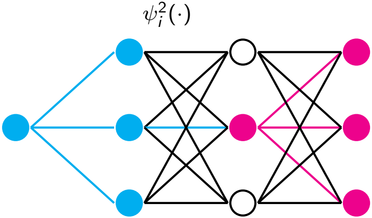
$\rightarrow$ Cutting-plane generation phase.

Illustration of extSDDiP

At each iteration i:

- Sampling step
- Forward step

$$\underline{Q}_i^\eta(S_i^{a^\eta}) := \min \sum_{n \in \mathcal{W}^\eta} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathcal{L}(\eta)} \psi_i^t(S^\ell)$$



- Backward step

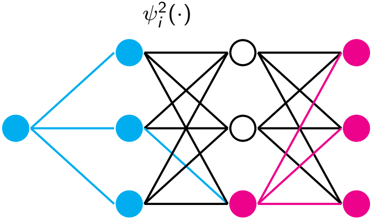
$\rightarrow$ Cutting-plane generation phase.

Illustration of extSDDiP

At each iteration i:

- Sampling step
- Forward step

$$\underline{Q}_i^{\eta}(S_i^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathcal{L}(\eta)} \psi_i^t(S^\ell)$$



- Backward step

$\rightarrow$ Cutting-plane generation phase.

$$\psi_i^t(S^\ell) := \min\{\theta^\ell : \theta^\ell \geq \sum_{m \in \mathcal{C}(\ell)} \rho^{\ell m}(v_l^m + (\pi_l^m)^\intercal S^\ell)\}$$

Illustration of extSDDiP

At each iteration i:

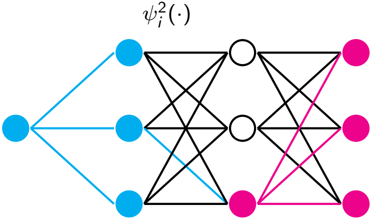
- Sampling step
- Forward step

$$\underline{Q}_i^{\eta}(S_i^{a^{\eta}}) := \min \sum_{n \in \mathcal{W}^{\eta}} \rho^n F^n(X^n, Y^n, S^n, W^n, L^n) + \sum_{\ell \in \mathcal{L}(\eta)} \psi_i^t(S^{\ell})$$

- Backward step

→ Cutting-plane generation phase.

$$\psi_i^t(S^{\ell}) := \min\{\theta^{\ell} : \theta^{\ell} \geq \sum_{m \in \mathcal{C}(\ell)} \rho^{\ell m}(v_l^m + (\pi_l^m)^{\intercal} S^{\ell})\}$$



Output: Lower and Upper bound

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 6 different structures of scenario tree leading to scenario trees involving between 1000 and 3.2 million scenarios.
- A total of 480 instances.
- Time limit of 7200 seconds.

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 6 different structures of scenario tree leading to scenario trees involving between 1000 and 3.2 million scenarios.
- A total of 480 instances.
- Time limit of 7200 seconds.

SDDiP algorithms:

- SDDiP: SDDiP algorithm proposed by Zou et al. [2019].

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 6 different structures of scenario tree leading to scenario trees involving between 1000 and 3.2 million scenarios.
- A total of 480 instances.
- Time limit of 7200 seconds.

SDDiP algorithms:

- SDDiP: SDDiP algorithm proposed by Zou et al. [2019].
- AppSDDiP: SDDiP algorithm using a full decomposition with continuous state variables.

Instances Generation and SDDiP algorithms

Random generation based on the numerical values used by Guan et al. [2009]

- All the parameters are defined by discrete uniform distribution $DU(L_j, U_j)$
- 6 different structures of scenario tree leading to scenario trees involving between 1000 and 3.2 million scenarios.
- A total of 480 instances.
- Time limit of 7200 seconds.

SDDiP algorithms:

- SDDiP: SDDiP algorithm proposed by Zou et al. [2019].
- AppSDDiP: SDDiP algorithm using a full decomposition with continuous state variables.
- ExtSDDiP: The proposed extended SDDiP algorithm using a partial dynamic decomposition.

Numerical results

Instance				CPLEX	SDDiP	AppSDDiP	ExtSDDiP
Σ	R	b	#*Scen*	Gap (Time)	Gap (Time)	Gap (Time)	Gap (Time)
4	10	1	1,000				
	20	1	8,000				
6	10	1	100,000				
	20	1	3,200,000				
8	5	2	78,125				
	5	5	78,125				
12	3	1	177,000				
	3	3	177,000				
Average							

- Total of 480 randomly generated instances
- Average values for 60 randomly generated instances
- Resolution with CPLEX 12.8
- Time limit : 7200s

- Results published in *International Journal of Production Research*. [Quezada et al., 2023]

Numerical results

Instance				CPLEX	SDDiP	AppSDDiP	ExtSDDiP
Σ	R	b	#*Scen*	Gap (Time)	Gap (Time)	Gap (Time)	Gap (Time)
4	10	1	1,000	0.24 (6,513)			
	20	1	8,000	1.57 (7,201)			
6	10	1	100,000	68.27 (7,217)			
	20	1	3,200,000	- (-)			
8	5	2	78,125	93,48 (7,236)			
	5	5	78,125	- (-)			
12	3	1	177,000	91,33 (7,243)			
	3	3	177,000	- (-)			
Average				50.98 (5,638)			

- Total of 480 randomly generated instances
- Average values for 60 randomly generated instances
- Resolution with CPLEX 12.8
- Time limit : 7200s

- Results published in *International Journal of Production Research*. [Quezada et al., 2023]

Numerical results

Instance				CPLEX	SDDiP	AppSDDiP	ExtSDDiP
Σ	R	b	#*Scen*	Gap (Time)	Gap (Time)	Gap (Time)	Gap (Time)
4	10	1	1,000	0.24 (6,513)	23.55 (4,306)		
	20	1	8,000	1.57 (7,201)	22.88 (4,202)		
6	10	1	100,000	68.27 (7,217)	30.37 (5,641)		
	20	1	3,200,000	- (-)	35.67 (5,608)		
8	5	2	78,125	93,48 (7,236)	47.76 (7,151)		
	5	5	78,125	- (-)	41.74 (7,222)		
12	3	1	177,000	91,33 (7,243)	51.04 (7,207)		
	3	3	177,000	- (-)	55.60 (7,230)		
Average				50.98 (5,638)	38.57 (6,071)		

- Total of 480 randomly generated instances
- Average values for 60 randomly generated instances
- Resolution with CPLEX 12.8
- Time limit : 7200s

- Results published in *International Journal of Production Research*. [Quezada et al., 2023]

Numerical results

Instance				CPLEX	SDDiP	AppSDDiP	ExtSDDiP
Σ	R	b	#*Scen*	Gap (Time)	Gap (Time)	Gap (Time)	Gap (Time)
4	10	1	1,000	0.24 (6,513)	23.55 (4,306)	7.46 (1,358)	
	20	1	8,000	1.57 (7,201)	22.88 (4,202)	6.98 (1,974)	
6	10	1	100,000	68.27 (7,217)	30.37 (5,641)	8.81 (2,513)	
	20	1	3,200,000	- (-)	35.67 (5,608)	9.83 (3,664)	
8	5	2	78,125	93,48 (7,236)	47.76 (7,151)	9.11 (3,929)	
	5	5	78,125	- (-)	41.74 (7,222)	6.68 (3,815)	
12	3	1	177,000	91,33 (7,243)	51.04 (7,207)	11.00 (3,313)	
	3	3	177,000	- (-)	55.60 (7,230)	9.72 (3,607)	
Average				50.98 (5,638)	38.57 (6,071)	8.70 (3021)	

- Total of 480 randomly generated instances
- Average values for 60 randomly generated instances
- Resolution with CPLEX 12.8
- Time limit : 7200s

- Results published in *International Journal of Production Research*. [Quezada et al., 2023]

Numerical results

Instance				CPLEX	SDDiP	AppSDDiP	ExtSDDiP
Σ	R	b	#*Scen*	Gap (Time)	Gap (Time)	Gap (Time)	Gap (Time)
4	10	1	1,000	0.24 (6,513)	23.55 (4,306)	7.46 (1,358)	1.18 (1,956)
	20	1	8,000	1.57 (7,201)	22.88 (4,202)	6.98 (1,974)	4.70 (3,463)
6	10	1	100,000	68.27 (7,217)	30.37 (5,641)	8.81 (2,513)	5.61 (4,579)
	20	1	3,200,000	- (-)	35.67 (5,608)	9.83 (3,664)	7.59 (4,814)
8	5	2	78,125	93,48 (7,236)	47.76 (7,151)	9.11 (3,929)	5.64 (4,579)
	5	5	78,125	- (-)	41.74 (7,222)	6.68 (3,815)	4.11 (5,884)
12	3	1	177,000	91,33 (7,243)	51.04 (7,207)	11.00 (3,313)	6.29 (3,595)
	3	3	177,000	- (-)	55.60 (7,230)	9.72 (3,607)	5.65 (5.987)
Average				50.98 (5,638)	38.57 (6,071)	8.70 (3021)	5.10 (4357)

- Total of 480 randomly generated instances
- Average values for 60 randomly generated instances
- Resolution with CPLEX 12.8
- Time limit : 7200s

- Results published in *International Journal of Production Research*. [Quezada et al., 2023]

Two approaches:

Two approaches:

- Branch-and-cut algorithm
  $\rightarrow$ New sets of effective valid inequalities

Two approaches:

- Branch-and-cut algorithm
  $\rightarrow$ New sets of effective valid inequalities

  Advantages:

  $\rightarrow$ Exact solution for small-size scenarios trees.
  $\rightarrow$ No assumptions on the stochastic parameters.

Two approaches:

- Branch-and-cut algorithm
  $\rightarrow$ New sets of effective valid inequalities

  Advantages:

  $\rightarrow$ Exact solution for small-size scenarios trees.
  $\rightarrow$ No assumptions on the stochastic parameters.

  Disadvantages:

  $\rightarrow$ Poor performance (or impractical) for large-size scenario trees.

Two approaches:

- Branch-and-cut algorithm
  $\rightarrow$ New sets of effective valid inequalities

  Advantages:

  $\rightarrow$ Exact solution for small-size scenarios trees.
  $\rightarrow$ No assumptions on the stochastic parameters.

  Disadvantages:

  $\rightarrow$ Poor performance (or impractical) for large-size scenario trees.

- Stochastic dual dynamic programming algorithm.
  $\rightarrow$ New dynamic programming decomposition
  $\rightarrow$ New cuts strategy to approximate expected cost-to-go functions.

Two approaches:

- Branch-and-cut algorithm
  $\rightarrow$ New sets of effective valid inequalities

  Advantages:
  - $\rightarrow$ Exact solution for small-size scenarios trees.
  - $\rightarrow$ No assumptions on the stochastic parameters.

  Disadvantages:
  - $\rightarrow$ Poor performance (or impractical) for large-size scenario trees.

- Stochastic dual dynamic programming algorithm.
  $\rightarrow$ New dynamic programming decomposition
  $\rightarrow$ New cuts strategy to approximate expected cost-to-go functions.

  Advantages:
  - $\rightarrow$ Near-optimal solutions for large-size scenario trees.
  - $\rightarrow$ Better approximation of expected cots-to-go functions through additional strengthened Benders' cuts.

Two approaches:

- Branch-and-cut algorithm
  → New sets of effective valid inequalities

  Advantages:

  → Exact solution for small-size scenarios trees.
  → No assumptions on the stochastic parameters.

  Disadvantages:

  → Poor performance (or impractical) for large-size scenario trees.

- Stochastic dual dynamic programming algorithm.
  → New dynamic programming decomposition
  → New cuts strategy to approximate expected cost-to-go functions.

  Advantages:

  → Near-optimal solutions for large-size scenario trees.
  → Better approximation of expected cots-to-go functions through additional strengthened Benders' cuts.

  Disadvantages:

  → Stage-wise independence assumption.

Acknowledgment

We would like to thank the Programa de Cooperación Científica ECOS-ANID PC23E10-ECOS230013 for their support and funding, which made this research and presentation possible.


soolong
matondo
meharbani
merkzi
kinisou
thank you
wado
maururu
rahmat
murakoze
paldies
dėkoju
salamat
barkal
welalin
manana
sadǫl
tenki
danki
ngiyabonga
blagodaria
gracias
cheers
akiba
stuutiyi
dekuji
grassie
dankie
sobodi
tänan
tanmirt
waybale
waita
manjuthe
omol
efharisto
yekeniele
zikomo
ahsante
multumesc
dankewol
nuhun
dziekuje
obrigado
dziakuju
arigato
blagodaram
dakujem
obrigada
madlobt
saha
tashakor
merci
shukriyaa
sulpáy
spasiba
grazie
dankegon
miigwech
matondi
misaotra
gràcies
gratzias
modupe
danke
shakkran
supas
shukria
mahalo
hvala
marahaba
aitäh
bayarlalaa
köszönöm
takk
vinaka
talofa
spas
tack
chokrane
tanemirt
nkosi
trugéré
buznyg
shukran
nandri
bedankt
menlau
mèsi
aabar
kiitos
skee
taiku

References I

Hyung-Dae Ahn, Dong-Ho Lee, and Hwa-Joong Kim. Solution algorithms for dynamic lot-sizing in remanufacturing systems. *International Journal of Production Research*, 49(22):6729–6748, 2011.

C Franke, B Basdere, M Ciupek, and S Seliger. Remanufacturing of mobile phones—capacity, program and facility adaptation planning. *Omega*, 34(6):562–570, 2006.

Yongpei Guan, Shabbir Ahmed, and George L Nemhauser. Cutting planes for multistage stochastic integer programs. *Operations research*, 57(2):287–298, 2009.

Shui Hua Han, MY Dong, Shui Xiu Lu, Stephen CH Leung, and Ming Kim Lim. Production planning for hybrid remanufacturing and manufacturing system with component recovery. *Journal of the Operational Research Society*, 64(10):1447–1460, 2013.

Mehmet Ali Ilgin and Surendra M Gupta. Environmentally conscious manufacturing and product recovery (ecmpro): a review of the state of the art. *Journal of environmental management*, 91(3):563–591, 2010.

Vaidyanathan Jayaraman. Production planning for closed-loop supply chains with product recovery and reuse: an analytical approach. *International Journal of Production Research*, 44(5):981–998, 2006.

References II

Muris Lage Junior and Moacir Godinho Filho. Production planning and control for remanufacturing: literature review and analysis. *Production Planning & Control*, 23(6): 419–435, 2012.

Muris Lage Junior and Moacir Godinho Filho. Production planning and control for remanufacturing: exploring characteristics and difficulties with case studies. *Production Planning & Control*, 27(3):212–225, 2016.

Marko Loparic, Yves Pochet, and Laurence A Wolsey. The uncapacitated lot-sizing problem with sales and safety stocks. *Mathematical Programming*, 89(3):487–504, 2001.

Mario VF Pereira and Leontina MVG Pinto. Multi-stage stochastic optimization applied to energy planning. *Mathematical programming*, 52(1-3):359–375, 1991.

Yves Pochet and Laurence A Wolsey. *Production planning by mixed integer programming*. Springer Science & Business Media, 2006.

Franco Quezada, Céline Gicquel, Safia Kedad-Sidhoum, and Dong Quan Vu. A multi-stage stochastic integer programming approach for a multi-echelon lot-sizing problem with returns and lost sales. *Computers & Operations Research*, 116:104865, 2020.

References III

Franco Quezada, Céline Gicquel, and Safia Kedad-Sidhoum. A stochastic dual dynamic integer programming based approach for remanufacturing planning under uncertainty. *International Journal of Production Research*, 61(17):5992–6012, 2023. doi: 10.1080/00207543.2022.2120924. URL https://doi.org/10.1080/00207543.2022.2120924.

Jikai Zou, Shabbir Ahmed, and Xu Andy Sun. Stochastic dual dynamic integer programming. *Mathematical Programming*, 175(1):461–502, 2019.