

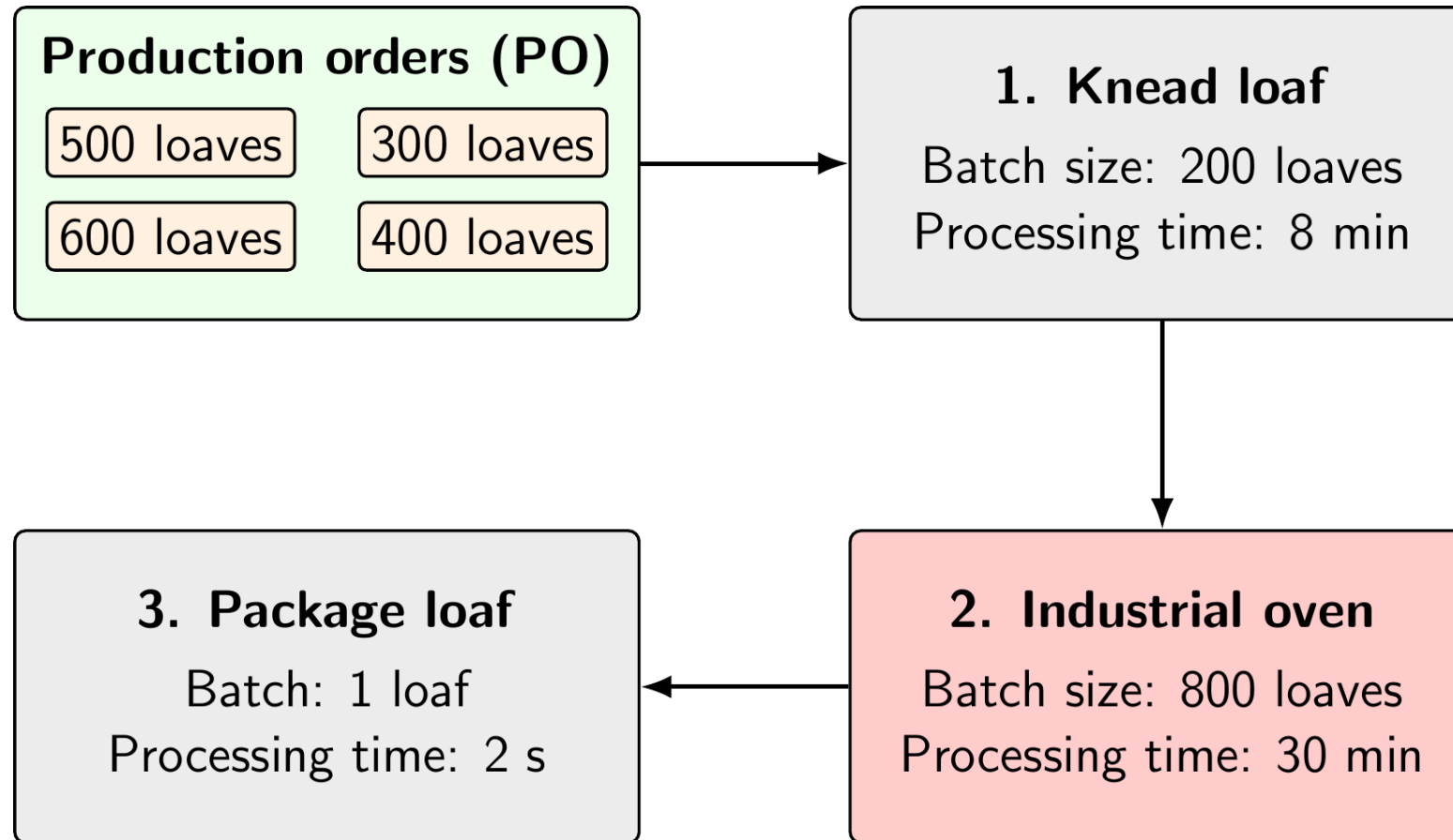
Stochastic batch scheduling in an industrial context

Florent Hervers^{1,2}, Guillaume Derval¹, Quentin Louveaux¹, Xavier Renard²

¹ University of Liège, Montefiore Institute, Belgium

² SmartYou

Context



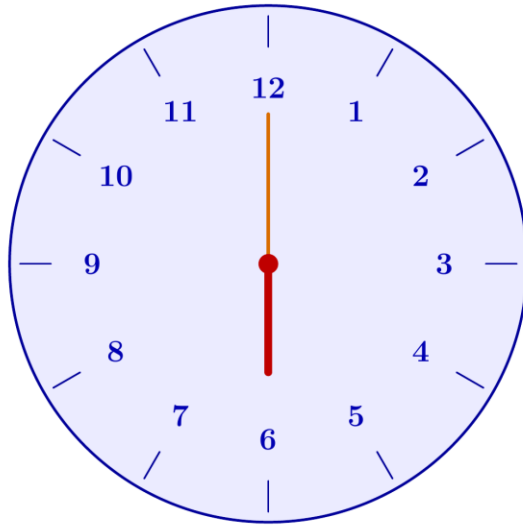
Problem statement

- Input data:
 - Oven capacity
 - Oven baking duration
 - Loaves waiting at the oven
 - Sizes of next production orders
 - Estimated arrival time at the oven
- Decisions:
 - **Start** the oven
 - **Wait** for the next production order

Example

Production order	Arrival time at oven	Size
1	0	500 loaves
2	10	300 loaves
3	40	600 loaves
4	70	400 loaves

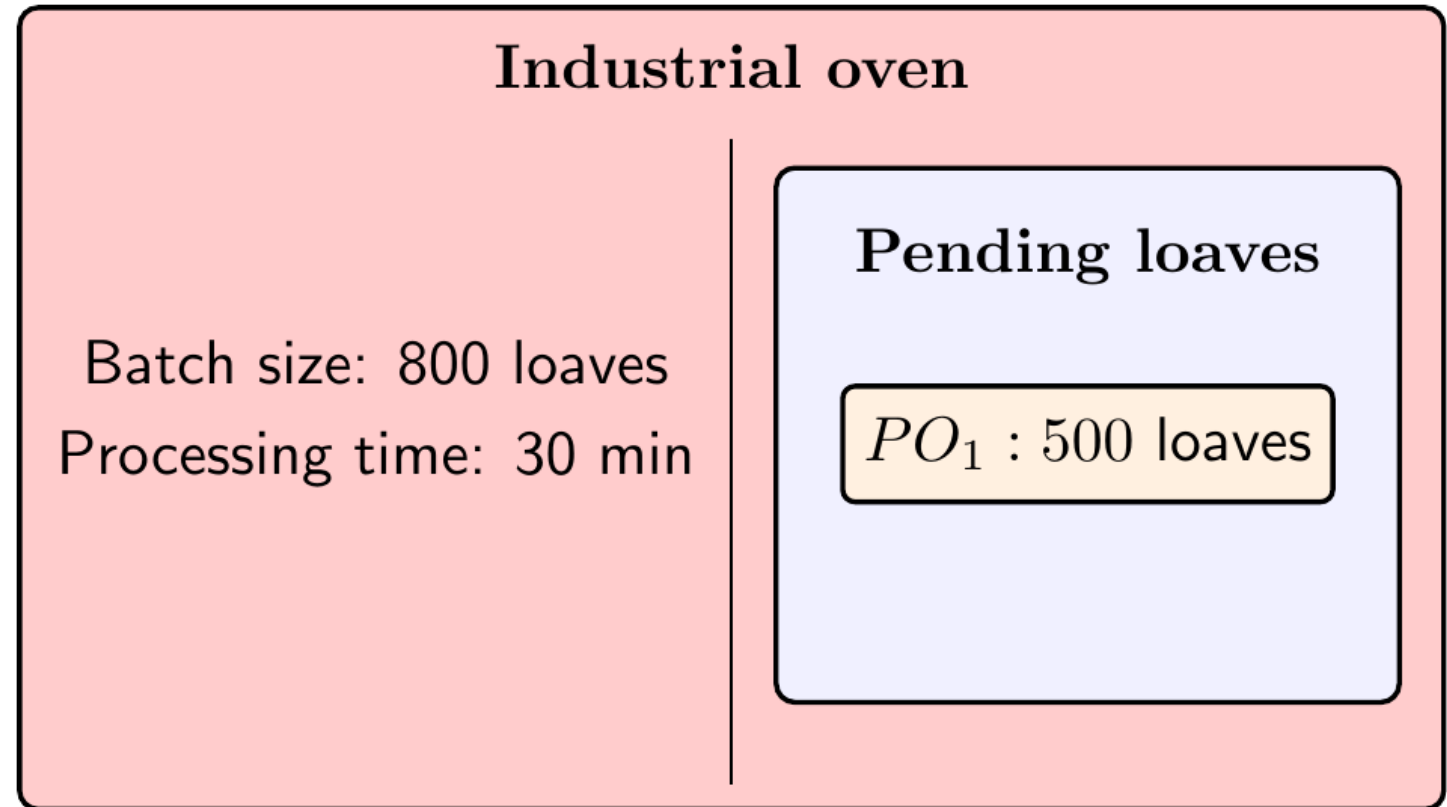
Initial configuration



Events:

- PO 1 just arrived

Decision = wait



First scenario



Events:

- PO 2 just arrived

Decision = start

Industrial oven

Batch size: 800 loaves
Processing time: 30 min

$$O_{10}(S_1) = 10 \text{ min}$$

Pending loaves

PO_1 : 500 loaves

PO_2 : 300 loaves

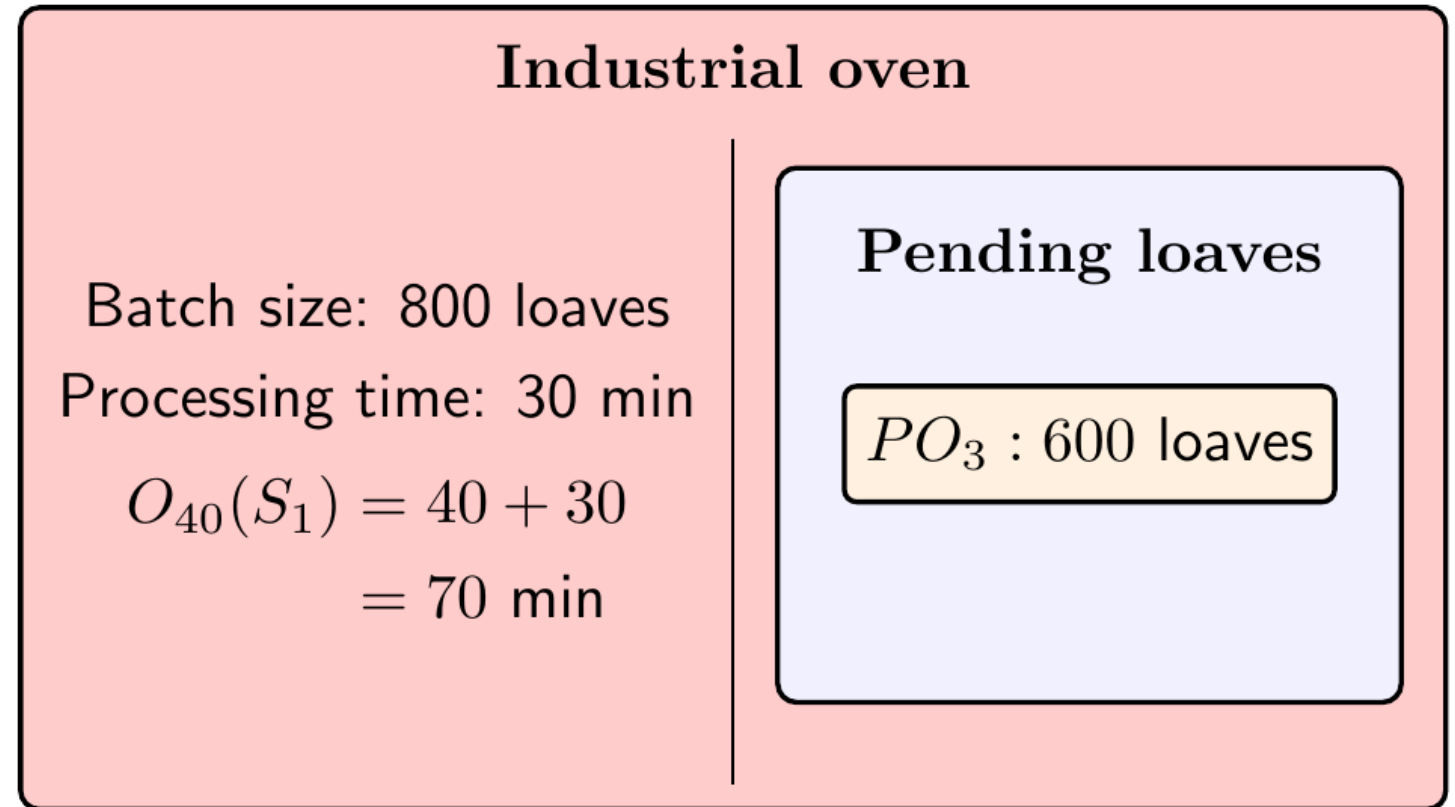
First scenario



Events:

- PO 3 just arrived
- The oven became available

Decision = start



First scenario



Events:

- PO 4 just arrived
- The oven became available

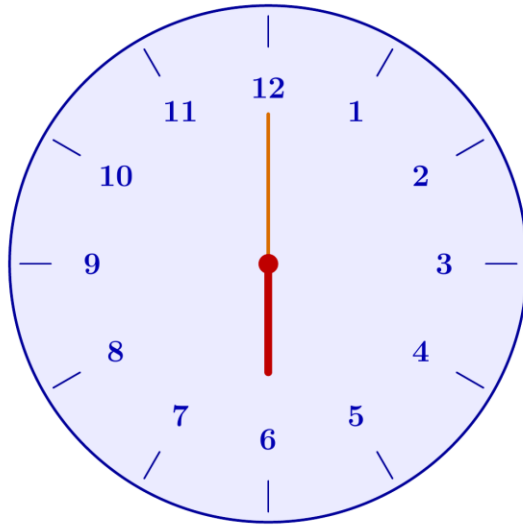
Industrial oven

Batch size: 800 loaves
Processing time: 30 min
 $O_{70}(S_1) = 40 + 30 + 30$
 $= 100 \text{ min}$

Pending loaves

$PO_4 : 400 \text{ loaves}$

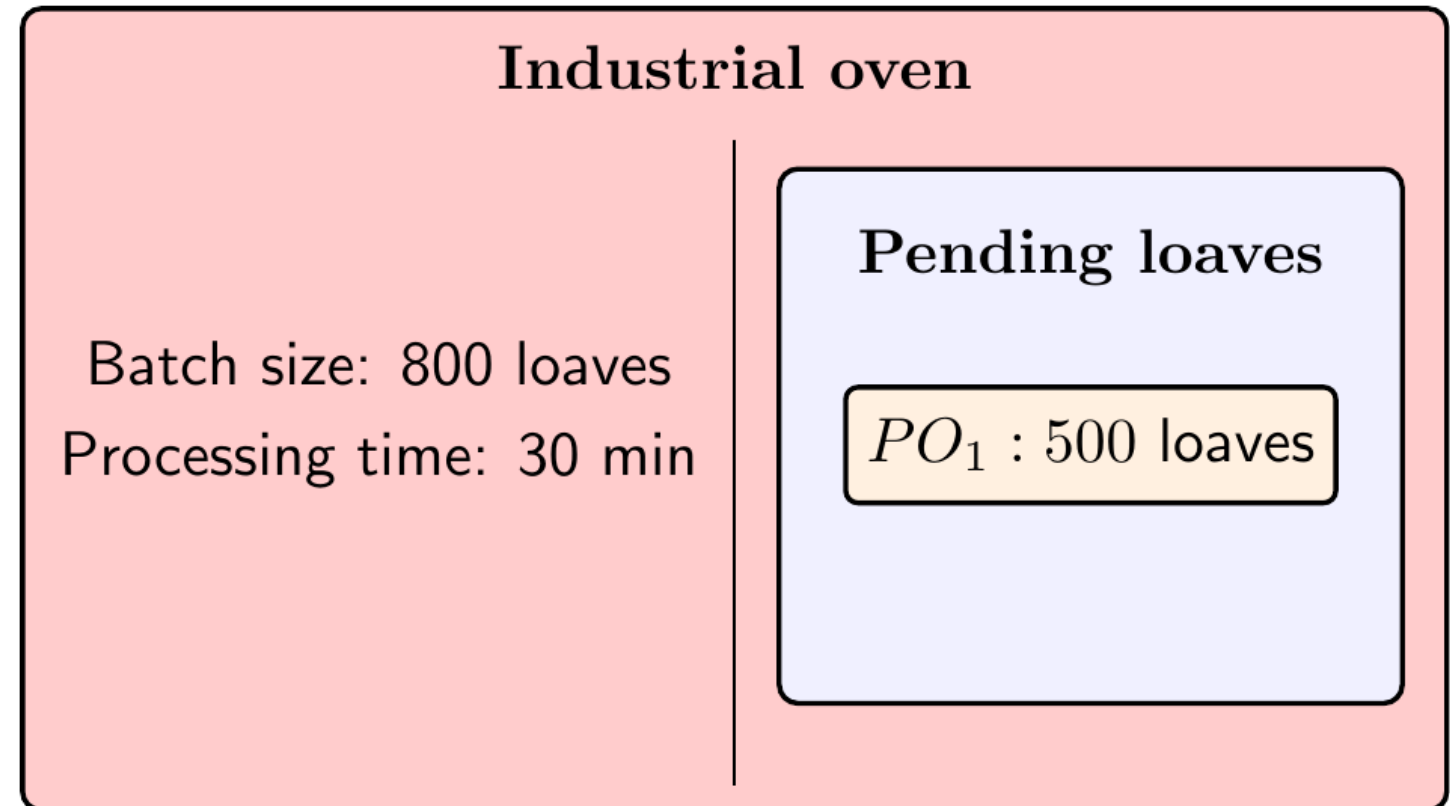
Initial configuration



Events:

- PO 1 just arrived

Decision = start



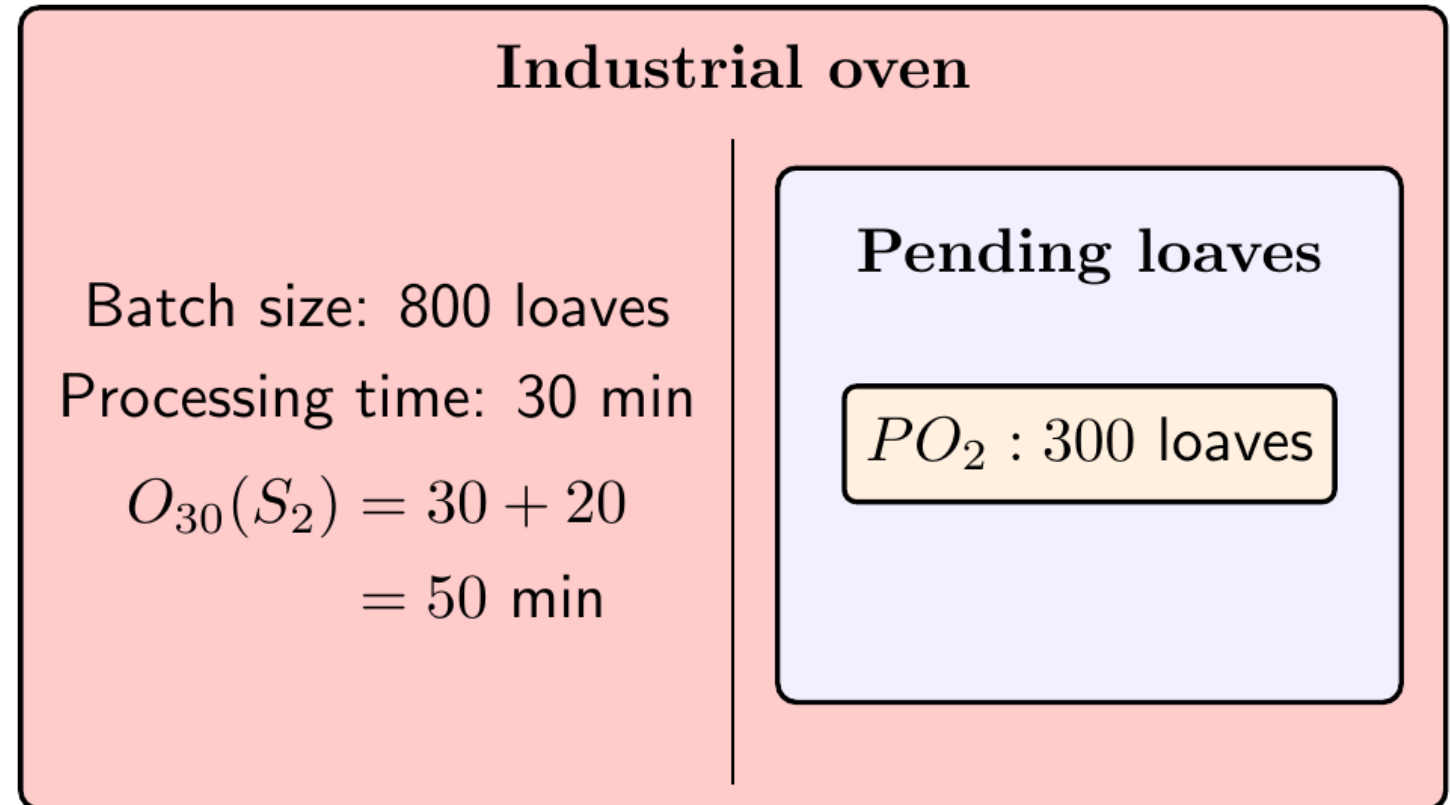
Second scenario



Events:

- PO 2 arrived 20 minutes ago
- The oven became available

Decision = wait



Second scenario



Events:

- PO 3 just arrived

Decision = start

Industrial oven

Batch size: 800 loaves
Processing time: 30 min

$$O_{40}(S_2) = 30 + 30 \\ = 60 \text{ min}$$

Pending loaves

PO_2 : 300 loaves

PO_3 : 600 loaves

Second scenario



Events:

- PO 4 just arrived
- The oven became available

Industrial oven

Batch size: 800 loaves
Processing time: 30 min
 $O_{70}(S_2) = 30 + 60 + 30$
 $= 120 \text{ min}$

Pending loaves

PO_3 : 100 loaves

PO_4 : 400 loaves

Summary

	Scenario 1 (S_1)	Scenario 2 (S_2)
Decisions	Wait, Start, Start	Start, Wait, Start
Pending loaves at $t = 70$	400	500
O_{70}	100	120

Dynamic programming (DP) algorithm

- Objective:

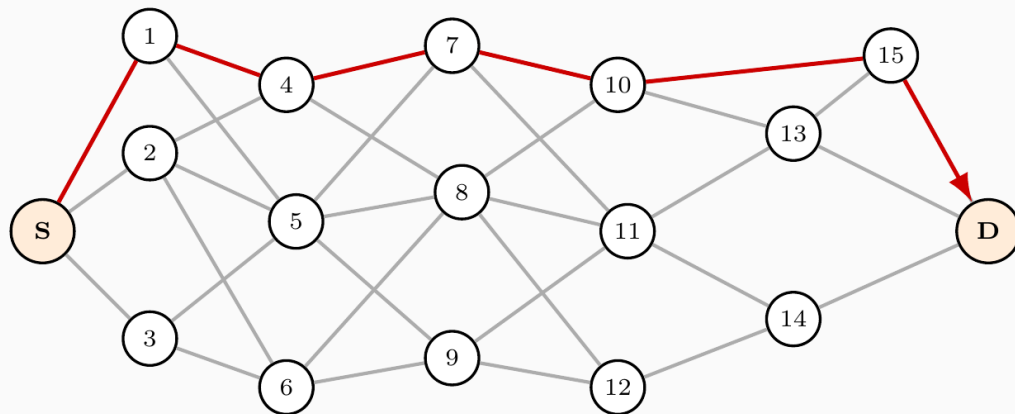
$$O(S) = \min \sum_{i=1}^N completion_i - arrival_i$$

- Evaluate the different scenarios in parallel
- Keep only the best scenarios

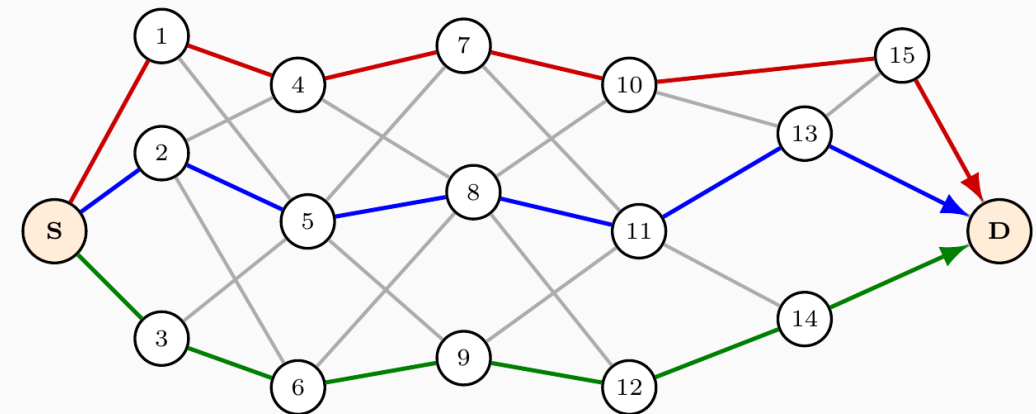
Limitations

- No randomness included
- Offline algorithm
 - Not suited for industrial use
 - Requires high prediction accuracy
- Difficult to explain the decision

Follow up



Follow Route B: $S \rightarrow 1 \rightarrow 4 \rightarrow 7 \rightarrow 10 \rightarrow 15 \rightarrow D$



Route A: 24 min
Low congestion expected.

Route B: 21 min
Fastest with one toll.

Route C: 27 min
Safer with no toll

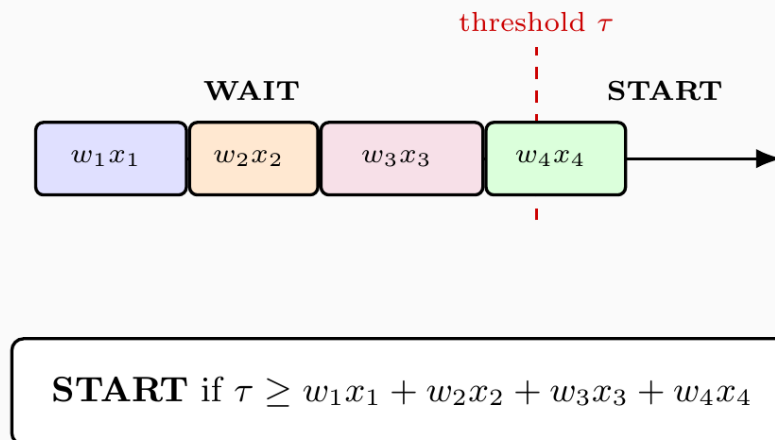
Online algorithms

- Focus on the decision to take now
- Provide the best decision on average
- Input features:
 - Pending units
 - Estimated arrival time and size of the next PO

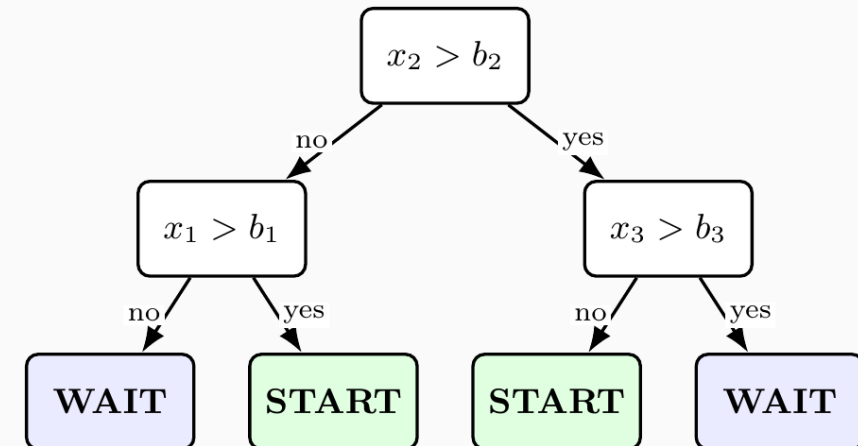
Online algorithms

- Limited to explainable algorithms

Linear models



Decision trees



Online rules: decision trees

- Classification and regression trees (CART) = heuristic.
- What about an MIP formulation?
 - Bertsimas and Dunn (2017): Optimal classification trees (OCT)

Bertsimas, D., & Dunn, J. (2017). Optimal classification trees. *Machine Learning*, 106(7), 1039–1082. <https://doi.org/10.1007/s10994-017-5633-9>

OCT iterative training

- Can we select training samples effectively?
 1. Start from an OCT built on a small dataset.
 2. Sample new points from the dataset
 3. Expand the model with misclassified points and points close to the decision boundary.
 4. Optimize the expanded model
 5. Start from step 2

Synthetic dataset creation

1. Generate several instances

Production order	Arrival time at oven	Size
1	0	500 loaves
2	10	300 loaves
3	40	600 loaves
4	70	400 loaves

2. Solve with the DP algorithm

3. Create the couple features/decision based on the solution

$$S_1 \text{ decisions: Wait, Start, Start} \Rightarrow \left\{ \begin{array}{l} (500, 10, 300), \text{ Wait} \\ (800, 30, 600), \text{ Start} \\ (600, 30, 400), \text{ Start} \end{array} \right.$$

Results

	Wait accuracy	Start accuracy	Average regret
Wait until full	/	/	4 384 746
Always Start	/	/	92 744
SVM	85.518 %	97.783%	20 666
Ridge regression	64.217 %	97.256 %	70 079
CART	89.986 %	97.565 %	38 608
OCT	82.147 %	98.107 %	34 013
OCT (iterative)	70.879 %	96.631 %	28 027

Thank you for your attention!

Questions?